Imperial College London

Micromechanical Modelling of Damage Healing in Free Cutting Steel

A thesis submitted for the Degree of Doctor of Philosophy

by

Shireen Afshan

September 2013

Department of Mechanical Engineering
Imperial College London

Declaration of originality

I confirm that this thesis is my own work and that any material from published or unpublished work from others is appropriately referenced.

Signature:

Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Acknowledgement

Summary

Continuous casting is used to solidify most of the steel produced in the world every year. The process reduces the number of required milling stages and results in qualitative semi-finished products such as billets, blooms and slabs which will later be rolled into more specific shapes. Extending the range of finished product sizes produced from a given concast bloom or billet section is often limited by the minimum area reduction required to ensure effective consolidation and final mechanical properties. Predicting effective consolidation or level of remnant porosity has always been an important issue for steel producers as it will affect the mechanical properties of final products (strength, ductility, etc.). It is known that partial or complete recovery of strength in such porous materials can be obtained by pore closure and diffusive healing processes at elevated temperature. Devising an appropriate healing process which does not cause discontinuity in the microstructure and mechanical properties at the healing sites and prevents distortion of the component during bonding requires an accurate choice of thermo-mechanical processing parameters. Although there has been considerable work on materials such as titanium alloys, aluminium alloys and copper, damage healing in free cutting steel has not received much attention. The main aim of this research is to develop a realistic damage healing computational approach that can predict damage healing or recovery during soaking under different compressive stress levels, and be used for hot rolling applications.

This study investigates the void elimination process through two stages of void closure and healing. An Abaqus/UMAT subroutine has been developed for the analysis of the material porosity elimination process including two stages of void closure and healing. This study uses the Gurson-Tvergaard model under hydrostatic compression to predict the void closure. A novel approach has been developed in the present work to identify the Gurson-Tvergaard model parameters using a non-

gradient based optimisation search method (Pattern Search Method). The healing process is modelled based on a combination of diffusion bonding, creep and plasticity following the Pilling model and can be adapted to any other healing/diffusion bonding model. The material model has been calibrated for free cutting steel and a stress state representative of the rolling process, and used to predict the closure and healing processes under rolling. The effect of parameters such as Roll Gap shape Factor (RGF), initial amount and distribution of void volume fraction on porosity elimination during rolling has also been investigated.

An experimental technique has been developed to identify the conditions (temperature, pressure, time) required for void elimination in Free Cutting Steel (FCS). Different combinations of load and time were tested and optimum conditions have been obtained. Tensile tests on the bonded specimens have been carried out to measure the strength of the bonded region. The position of fracture on the specimen and also the cross section of the fracture surface have been inspected.

The experimental results have been used to calibrate the developed void elimination model. Using the developed model, predictions of densification and healing can be made for optimisation of the rolling schedule.

Table of Contents

List of Figures

xvi

xviii

List of Tables

Nomenclature

A	Dimensionless constant in Beere and Speight model
A_ε	Strain dependent void nucleation intensity
A_c	Power law creep pre-exponential constant
b	Lattice size
b_1	Void to void distance in Wang and Li model
b_v	Burger's vector
B_σ	Stress dependent void nucleation intensity
c	Grain boundary radius
d	Grain size
D	Density of material relative to water
D	Diffusion coefficient
D_o	Pre-exponential factor
D_{gb}	Grain boundary diffusion coefficient
D_p	Pipe diffusion coefficient associated with dislocations
D_s	Surface diffusion coefficient
D_v	Lattice diffusion coefficient
E	Young's modulus
E_A	Activation energy
E_t	Tangent modulus of the matrix material
f	Void volume fraction in Gurson yield function
f_0	Initial void volume fraction
f_1^b, f_1	Initial void volume fraction inside and outside the band (Ohno and Hutchinson approach)
f_c	The critical void volume fraction
f_F	Void volume fraction at final fracture
f_h	Fractional area of non-bonded surface
f_N	Volume fraction of void nucleating particles
f^*	Accelerated void volume fraction
f_u^*	$= {}^1\!/_{q_1}$
$\dot{f}$	Total rate of change in void volume fraction

$\dot{f}_g$	The rate of growth of existing voids
$\dot{f}_n$	Void nucleation rate
F	Frequency factor
g	Arrhenius' process rate
H	Height of the cylindrical cell model
H_1, H_2	Work-piece thickness before and after passing through the rollers respectively.
H_m	Mean thickness of the work-piece in rolling
h	Instantaneous height of the cylinder bore (Pilling model)
h_o	Initial height of the cylinder bore (Pilling model)
i	Element number
I	Array storing element information
I_{ij}	Second-order unity tensor
j	Integration point
J	Diffusion flux
k_n	Increment number
k	Iteration number
K	Boltzmann's constant
k_{apex}	Curvature of the void apex
l_I	Diffusion distance
m	Void shape factor
m_e	Equilibrium state void shape parameter
M	Number of data points
n	Exponent of power law creep
N	Number of intervals
N_i	Number of interfaces intersecting the void surface
P	Applied pressure
P_h	Hydrostatic pressure
P_x	Lateral pressure
P_y	Axial pressure
$P*$	Threshold pressure
q_1, q_2 and q_3	Gurson's model fitting parameters
Q	Activation enthalpy

r	Bond neck radius (Derby and Wallach model)
$r_{c_{tip}}$	Radius of curvature at the void tip
$r_{c_{top}}$	Radius of curvature at the top of the void
r_h and r_o	The inner and outer radii of the cylinder respectively in Pilling model
r_i	Instantaneous inner radius in Pilling model
r_t	Radius of the cylindrical cell
R	The gas constant
R_r	Roller radius
$\bar{R}$	Average void size
s	Cluster size in Ohno and Hutchinson's approach
s_m	Mesh size
s_N	Standard deviation for nucleation strain
S_e	Element status
S	Sign of the bonding pressure, negative if compressive (Derby and Wallach model)
t	Time
t_c	Closure time
t_e	Void elimination time
t_{es}	Scaled void elimination time
t_h	Healing time
t_r	Rupture time
t_s	Simulation time
T	Stress triaxiality
T	Absolute temperature
T_m	Melting temperature
U	Degree of porosity
V	Volume transferred in bonding
V_m	Volume of the matrix
V_{mo}	Volume of the matrix at the start of plastic deformation
V_v	Volume of the void
$\dot{V}$	Rate of change of V with respect to time
vvf	Void volume fraction
vvf_c	Void volume fraction at closure state

W	Radius of the cylindrical cell model
x_1	Half void spacing (in Derby and Wallach model)
x_2	Unit of bonded length (in Derby and Wallach model)
y_l	Work-piece diagonal
α	Volume parameter ($\alpha = \rho/\rho_0$)
$\propto$	Scale parameter
β	The angle between a surface ridge and the horizontal (Derby and Wallach model)
γ	Surface energy
δ_{gb}	Grain boundary width (thickness)
δ_s	Thickness of surface diffusion layer
ε_M^p	Effective plastic strain of the matrix material
ε_N	Mean strain for nucleation
$\dot{\varepsilon}$	Strain rate
$\dot{\varepsilon}_c$	Creep strain rate
$\dot{\varepsilon}_{ij}^p$	Plastic strain rate
$\dot{\varepsilon}_r$	Effective strain rate in the radial direction
η	The number of atoms removed from grain boundary per unit time and volume
η_s	The scale factor
λ	Band thickness in Ohno and Hutchinson approach
Λ	Dimensionless parameter in Beere and Speight model
μ	Chemical potential
ξ	Radius of the spheroid at the void apex
ρ	Void radius
ρ_x	Radius of the spheroid in x and y direction in Wang and Li model
ρ_x	Radius of the spheroid in z direction in Wang and Li model
ρ_0	Initial void radius
σ	Stress
σ_{eq}	Von Mises or equivalent stress
σ_h	Hydrostatic stress
σ_n	Normal stress
σ_N	Mean stress for nucleation

σ_Y	Yield stress
$\bar{\sigma}$	Flow stress
Φ	Gurson yield function
φ	Void concentration
ψ	Dimensionless loading parameter
ψ_c	Critical value of ψ
ω	Stress ratio
Ω	Atomic volume

Chapter 1

Introduction

In this chapter the aim of the research and the motivation for the present project are discussed. The application of the project in the global steel industry is also outlined.

1.1 The research motivation

Defects undermine the mechanical properties of materials and reduce the structural integrity and service life. Voids and micro damage will grow under continuous loading. With increased concentration of micro damage in a material, micro cracks will be initiated, the spread and coalescence of which will cause the final macroscopic fracture of the material as critical conditions are reached.

The prediction and characterisation of micro crack propagation and the final failure of the material is of such importance that it has become a special field in materials science. Various fracture models have therefore been proposed to quantify the damage associated with material deformation and are used to predict fracture (Gurson, 1977, McClintock, 1968, Rice and Tracey, 1969, Rousselier, 1987).

Although high quality design and manufacturing processes can result in robust, strong products, the creation of damage cannot be avoided in some cases. However, this will not be an issue if processes such as damage elimination or healing are included in the application.

Healing is a process by which material defects such as voids and cracks reduce, resulting in a stiffer and stronger material with enhanced mechanical properties. It has been an attractive concept in different fields. Use of healing can be seen in many applications involving polymers and composites (Blaiszik et al., 2010, Wool and Oconnor, 1981), biomaterials (Fratzl et al., 2008, Vermolen et al., 2008) and also the recovery of concrete (Li et al., 2008, Reinhardt and Jooss, 2003). Metal healing has been mostly studied in terms of sinter powder metallurgy where mass transfer of metal occurs at high temperatures such as 0.7-0.9 of the melting temperature, producing either solid state or liquid phase bonding across powder interfaces (Lumley and Polmear, 2007).

Although healing and diffusion bonding have been widely studied for different alloys (Huang et al., 1999, Salehi et al., 1992, Somekawaa et al., 2003), the conditions required to create self-bonding of cast steel have never been investigated. Devising an appropriate healing process which does not cause discontinuity in the microstructure and in properties at the bonds and prevents distortion of the component during bonding requires an appropriate combination of temperature, pressure and time combination. This motivated the author to pursue this research with the objectives explained in section 1.3.

1.2 The research applications

Continuous casting is used to solidify most of the 750 million tons of steel produced in the world every year (Thomas, 2001). The process reduces the number of required milling stages and results in qualitative semi-finished products such as billets, blooms and slabs which will later be rolled into more specific shapes. Extending the range of finished product sizes produced from a given concast bloom or billet section is often limited by the minimum area reduction required to ensure effective central consolidation and final mechanical properties. Predicting effective consolidation or level of remnant porosity for a range of steel grade (function of solidification regime), billet size, pass schedule/roll design and thermo-mechanical conditions has always been an important issue for steel producers as it will affect the mechanical properties of final products (strength, ductility, etc.) (Watts et al., 1999). It is known that partial or complete recovery of strength in such porous materials can be obtained by pore closure and diffusive healing processes at elevated temperature.

This research will therefore focus on developing a realistic damage healing model that can predict damage healing or recovery during soaking under different compressive stress levels, and be used for hot rolling applications.

1.3 The research objectives

Prediction of the extent of porosity eliminated under particular temperature, time and load conditions requires a combination of high quality experimental data and observations as well as robust physically based models. Therefore the following objectives had to be achieved in this project:

- Understanding the damage recovery and diffusion bonding of steels under soaking and compressive stress conditions.
- Developing a method which enables damage healing under compressive stress conditions to be modelled.
- Establishing test procedures to enable damage healing behaviour under hot deformation conditions to be characterised.
- Implementing the determined model for Tata Steel applications.

1.4 Thesis outline

Chapter two is a literature review of the previously mentioned topics in this chapter, i.e. the damage concept and cast steel damage characteristics, the Gurson model, void closure, void healing and the concept of diffusion bonding.

Chapter three reviews the existing models for void elimination including models originating from sintering concepts and also those derived from growth models. The similarities in these models are discussed and the parameters causing discrepancies are briefly explained. The applicability of these models for the purpose of this study has also been discussed.

Chapter four explores the applicability of the Gurson model to predict void closure. A novel approach is developed in this chapter to identify the Gurson-Tvergaard (GT) model fitting parameters using a non-gradient based optimisation search method (Pattern Search Method). This method of calibration is advantageous over trial-and-

error fitting of FE simulations to experimental data which is expensive and time consuming.

In Chapter five an ABAQUS/UMAT subroutine is developed for the analysis of the material porosity elimination process including two stages of void closure and healing. The approach uses the GT model under compressive loading to predict void closure, and the so-called Pilling model to predict the healing time.

In Chapter six, the developed UMAT is applied to model the void elimination of a steel work-piece during rolling. The effect of parameters such as the Roll Gap shape Factor (RGF) and initial porosity distribution on void elimination processes are investigated.

An experimental technique is developed in Chapter seven to estimate the load, temperature and time required for healing of Free Cutting Steel (FCS). Different analysis techniques are used to examine the extent of healing and the optimum bonding parameters (load, temperature and time combination leading to complete bonding) are obtained. Experimental results are used to calibrate and validate the healing model previously introduced in previous chapters.

Finally, in Chapter eight conclusions are made based on the results gained in previous chapters and some guidelines are presented for future work.

1.5 References

Blaiszik, B. J., Kramer, S. L. B., Olugebefola, S. C., Moore, J. S., Sottos, N. R. & White, S. R. 2010. Self-healing polymers and composites. *Annual Review Materials*, 40**,** 179–211.

Fratzl, P., Weinkamer, R. & Zwaag, S. 2008. Hierarchical structure and repair of bone: deformation, remodelling, healing self healing materials. *Self Healing Materials*. Springer Netherlands.

Gurson, A. L. 1977. Continuum theory of ductile rupture by void nucleation and growth: Part I-Yield criteria and flow rules for porous ductile media. *Journal of Engineering Materials and Technology:Transactions of the ASME*, 99**,** 2-15.

Huang, Y., Ridley, N. & Humphreys, F. J. 1999. Diffusion bonding of superplastic 7075 aluminium alloy. *Materials Science and Engineering A*, 266**,** 295-302.

Li, V., Yang, E. & Zwaag, S. 2008. Self healing in concrete materials. *Self Healing Materials*. Springer Netherlands.

Lumley, R. N. & Polmear, I. J. 2007. Advances in self healing of metals. *Proceedings of the First International Conference on Self Healing Materials*. The Netherlands: Noordwijk aan Zee.

McClintock, F. A. 1968. A criterion for ductile fracture by the growth of holes. *Journal of Applied Mechanics*, 35**,** 363-371.

Reinhardt, H. W. & Jooss, M. 2003. Permeability and self-healing of cracked concrete as a function of temperature and crack width. *Cement and Concrete Research*, 33**,** 981-985.

Rice, J. R. & Tracey, D. M. 1969. On the ductile enlargement of voids in triaxial stress fields. *Journal of the Mechanics and Physics of Solids*, 17**,** 201-217.

Rousselier, G. 1987. Ductile fracture models and their potential in local approach of fracture. *Nuclear Engineering and Design*, 105**,** 97-111.

Salehi, M. T., Pilling, J., Rideley, N. & Hamilton, D. L. 1992. Isostatic diffusion bonding of superplastic Ti-6A1-4V. *Materials Science and Engineering: A,* 150**,** 1-6.

Somekawaa, H., Hosokawab, H., Watanabec, H. & Higashia, K. 2003. Diffusion bonding in superplastic magnesium alloys. *Materials Science and Engineering: A*, 339**,** 328-333.

Thomas, B. 2001. Continuous casting of steel. *In:* YU, O. (ed.) *Modeling for Casting and Solidification Processing*. New York: Marcel Dekker.

Vermolen, F., Rossum, M., Perez, E., Adam, J. & Zwaag, S. 2008. Modeling of self healing of skin tissue. *Self Healing Materials*. Springer Netherlands.

Watts, G. J., Farrugia, D. & Ingham, P. M. 1999. Comparison of methods for predicting the evolution of damage during metal forming. *Proceedings of the*

First International Conference on Component Optimisation. University of Wales, Swansea.

Wool, R. P. & Oconnor, K. M. 1981. A theory crack healing in polymers. *Journal of Applied Physics*, 52**,** 5953-5963.

Chapter 2

The concept of porosity and its elimination in cast steel

Failure of material is due to heterogeneous micro processes that deteriorate the mechanical properties of the material. A solid is regarded as damaged if some of its microstructural bonds are lost. Materials containing micro damage such as distributed microscopic voids, cavities, or micro cracks which might be process-induced or appear during loading are more prone to failure.

Damage recovery can be viewed as the reverse of the damage evolution process. Therefore it is essential to briefly review the damage concept and the developed ductile fracture models for porous materials. Amongst all ductile fracture models the well-known Gurson-Tvergaard (GT) (1981) approach will be of particular interest here since this study investigates the applicability of the GT model under reverse loading to predict void closure.

2.1 Porosity

Porosity is the amount of voids or empty spaces in a material and is usually defined by the parameter void volume fraction:

$$void\ volume\ fraction = \frac{volume\ of\ the\ voids}{volume\ of\ the\ bulk\ material}$$

2.1

In most cases porosity is undesirable and causes deterioration of mechanical properties of materials. In the case of cast steel, producing a contiguous, porosity-free casting is very important. However, porosity may still be found in some parts which cannot be removed without unreasonable cost.

Detection and quantification of porosity using non-destructive methods is difficult. Microporosity maybe concealed within large sections and remain undetectable. Non-destructive techniques such as 2D and 3D metallographic methods (optical microscopy, scanning electron microscopy and X-ray tomography) are not feasible for large parts.

The methods used by designers to account for porosity in castings are diverse and uncertain. Despite performing a wide range of expensive quality control tests and using large safety factors, perfect performance cannot be guaranteed.

The porosity, however, can be minimised or eliminated by hot work. Cast steel with pre-existing porosity is compressed at high temperature during the rolling process where the cross section of the metal stock is reduced and products of desired shapes (sheets, bars, rods, etc.) are obtained.

2.1.1 Categorising porosity

Porous materials can be categorised into three groups based on porosity level (Zhang and Wang, 2005). The first group involves materials with porosity less than 1%;

9

these materials are usually described by models which ignore void interaction and assume isolated pores or a uniform distribution of pores. The well-known Gurson model and its modifications provide a good description for this group of porous materials.

The second category involves porous materials with low to medium porosity levels (10-70%) and includes recently developed heavy metal foams and sintered porous metals. Due to the lack of natural materials in this mid porosity category, this group is usually less investigated than the other categories.

The third group involves materials with porosity greater than 70%. A typical example of materials with this porosity range is cellular or foam materials, particularly open-cell foams. These materials are considered as structures where the solid medium acts as structural components, such as walls and edges, and pores are considered as empty spaces which have no effect on the overall strength of the materials. The modelling of these materials involves using representative volumes with strut-like edges of solid material for open-cell, sponge materials, or membrane-walled cells for closed cell, honeycomb materials.

The type of porosity found in as-cast steel is usually less than 1% and is better described by the first group. Therefore, the focus of this study will only be on the first category.

2.1.2 Effect of porosity on material properties

Pre-existing micro damage causes mechanical property degradation in materials. A study on cast steel showed that the elastic modulus of these materials decreases nonlinearly with porosity and that the steel exhibits a critical porosity level above which it loses all its stiffness (Hardin and Becketmann, 2007). The effect of micro porosity on the tensile properties of some aluminium alloys has also been investigated by some researchers (Surappa et al., 1986). Tensile loading of pore

containing materials causes strain concentrations in the vicinity of the voids and a corresponding flow stress reduction. In porous materials, a non-uniform porosity distribution leads to flow localisation and fracture initiation in high pore content paths. This explains the significant decrease in strength and ductility of damage-containing materials under tension. Figure 2.1 shows the porosity influence on yield stress, the elongation to failure and the percentage reduction in area for commercially pure titanium (C.P.Ti) and Ti-6Al-4V (Bourcier et al., 1986).

The adverse influence of pre-existing damage, particularly porosity, on mechanical properties of materials has created an urgent need for the development of methods of damage healing and porosity elimination.

2.2 Damage Concept

2.2.1 Damage in cast steel

Continuous casting involves flow of molten steel from a ladle, through a tundish into the mould. The liquid steel must be prevented from exposure to the oxygen and nitrogen in the atmosphere. In the mould, the hot liquid steel freezes alongside the water-cooled copper mould walls to make a solid shell which solidifies before the middle section. Towards the end of the mould, more of the liquid core solidifies, increasing the shell thickness. Solid steel is then withdrawn in a steady state from the mould base via rolls lower in the machine (Thomas, 2001). A schematic of continuous casting of steel is shown in Figure 2.2.

Porosity in casts could be caused by factors such as entrained air during filling, solidification shrinkage and dissolved gases (Monroe, 2005).

(a)

(b)

(c)
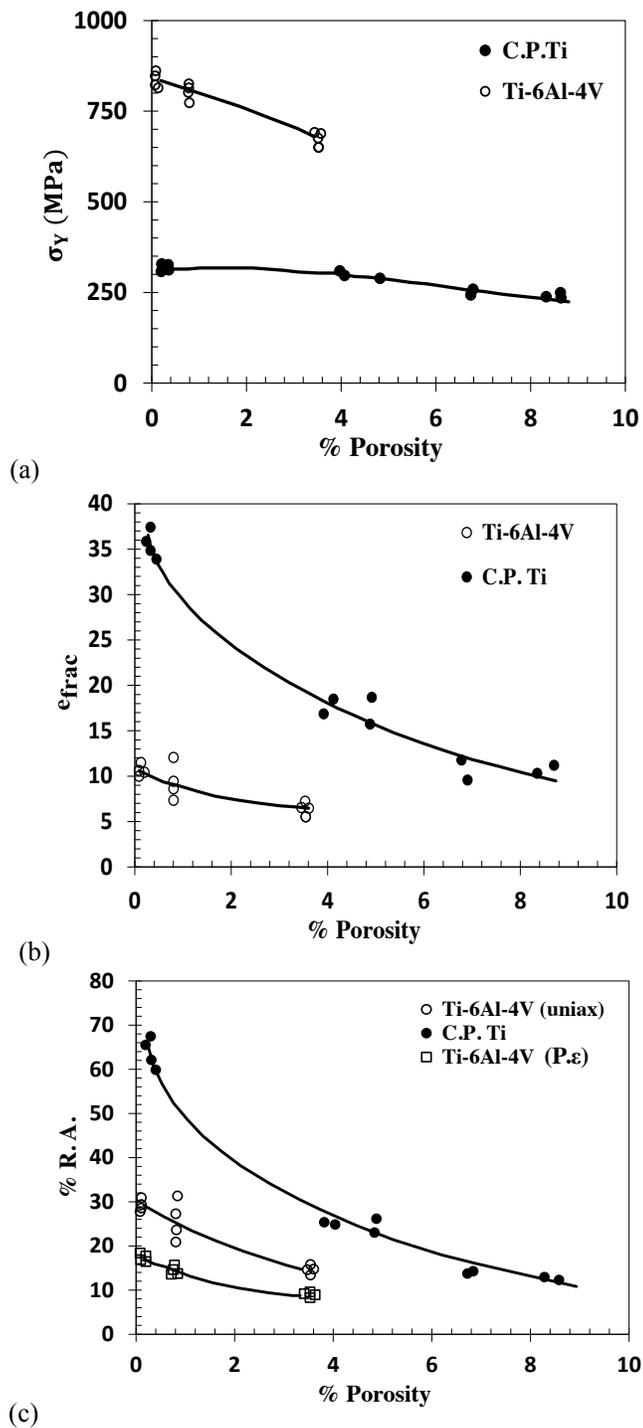
Figure 2.1*- The influence of porosity on (a) the yield stress (b) the elongation to failure, and (c) the percentage reduction in area %R.A. for C.P.Ti and Ti-6Al-4V (Bourcier et al., 1986). All data is from tests in uniaxial tension (uniax) except for the plane strain (P.ε) tests of Ti-6Al-4V in (c).*
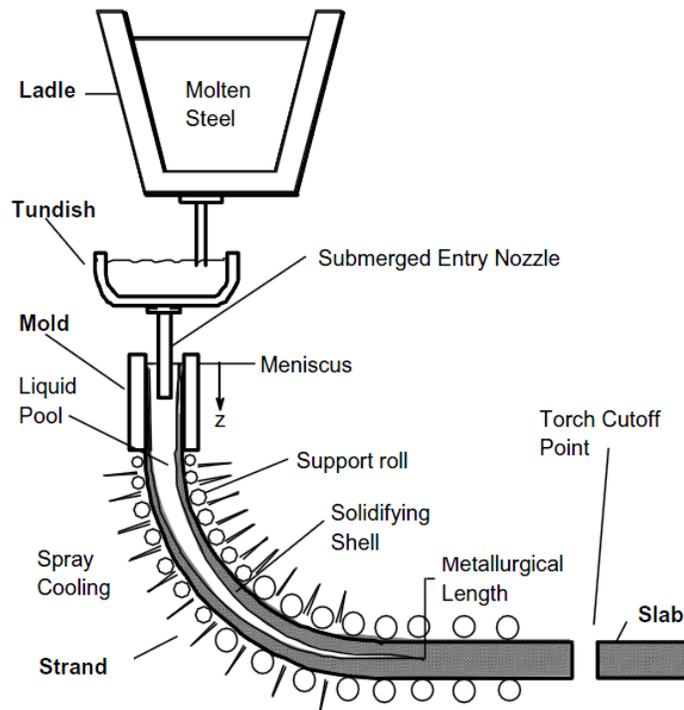
Figure 2.2*- Schematic of continuous casting of steel (Thomas, 2001).*

Solidification shrinkage occurs as a result of a change in density or specific volume when liquid metal solidifies. As solidification occurs the temperature and pressure near the solidifying interface drops, reducing the gas solubility in the liquid. Gases will therefore be rejected at the solid-liquid interface. Porosity formed under this condition is termed shrinkage porosity. Air bubbles are another type of cast defects and occur due to incomplete air elimination in the process of mould filling where low pressure in the liquid metal can cause air entrainment. In addition to air bubbles, gas bubbles can also be blown into the liquid from the core.

The size of individual pores may range in length from a few microns to a few millimetres. Large voids occur during solidification. During solidification of liquid metal, the volume decreases due to a difference in density between the liquid and solid phase. The liquid metal flows from neighbouring areas via inter-dendritic small channels to these parts (known as feeding process). As solidification progresses, the ability to feed becomes difficult due to the decreasing size of the channels. Therefore, some areas get isolated from the metal supply, leading to formation of

13

large pores. Figure 2.3 shows some SEM images of porosity in the centre of a leaded free cutting steel (LFCS) bloom.
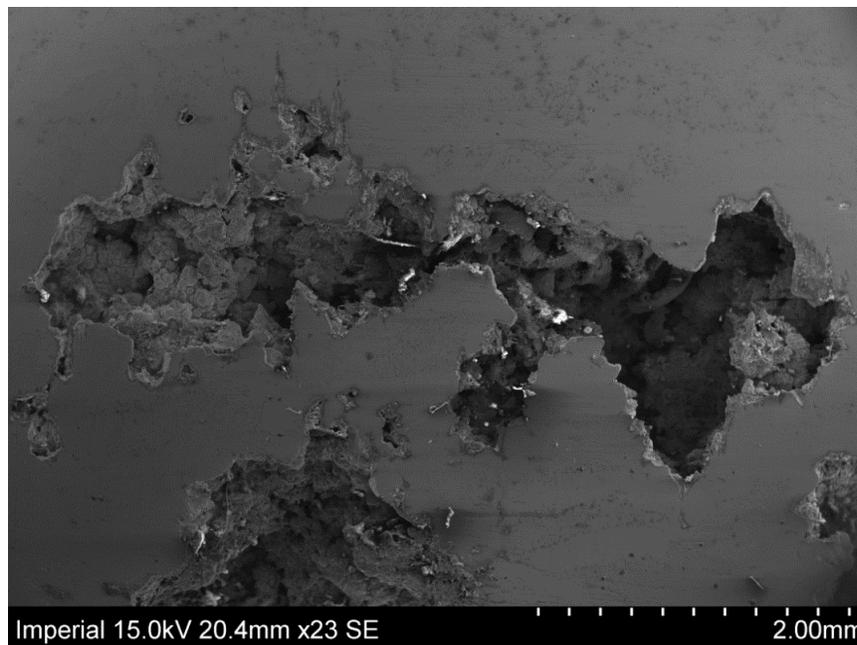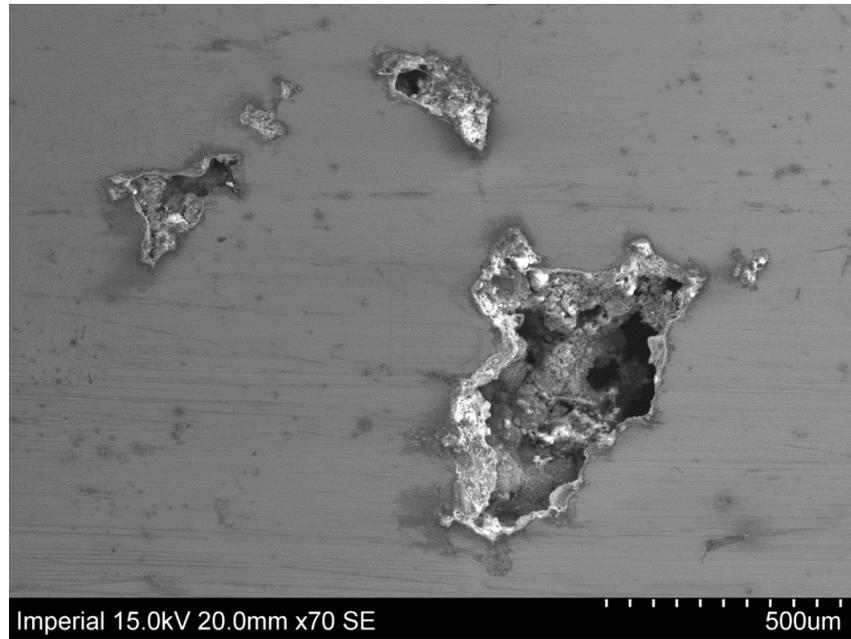
Figure 2.3- *SEM images of porosity in the centre of a leaded free cutting steel (LFCS) bloom (continued).*
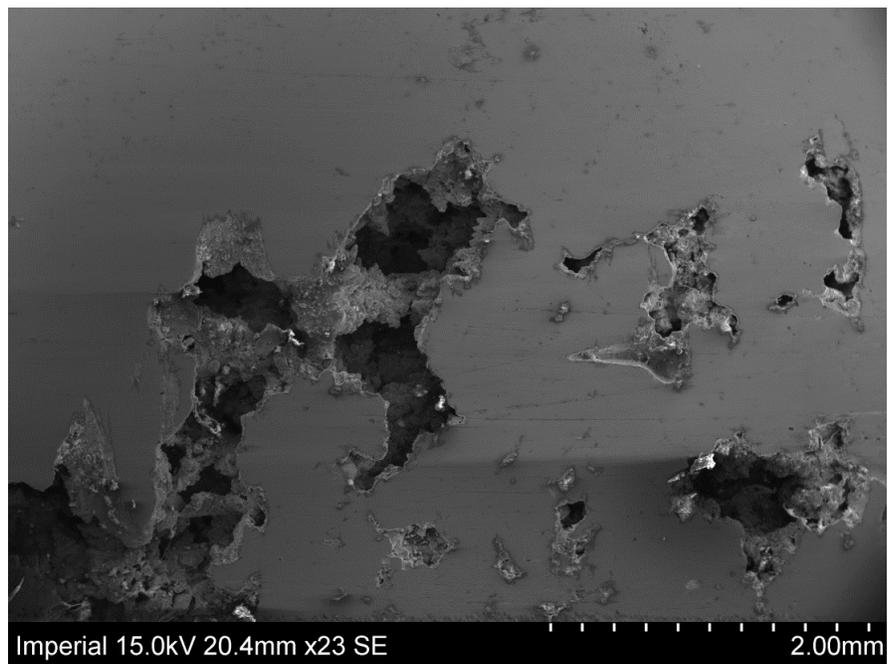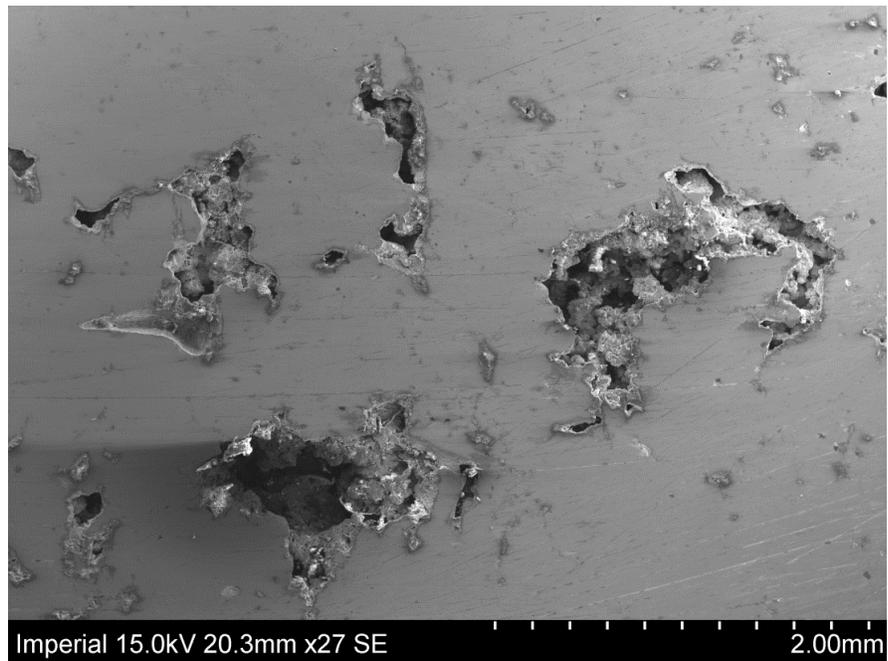
14

Imperial 15.0kV 20.3mm x27 SE   2.00mm

Imperial 15.0kV 20.4mm x23 SE   2.00mm

Figure 2.3- SEM images of porosity in the centre of a leaded free cutting steel (LFCS)

bloom.

15

2.2.2 Damage mechanisms

Depending on the loading conditions and temperatures involved, damage could be caused by different mechanisms; these include cleavage/brittle fracture, ductile fracture, creep and fatigue (Lemaitre, 1996). Due to hot temperature deformation of cast steel in rolling, the process of ductile fracture will be more of an interest here.

Ductile damage develops in a multistep failure process which involves several concurrent and interactive mechanisms – void nucleation, growth, and coalescence progressing simultaneously at different rates (Figure 2.4).

Nucleation: Impurities such as inclusions and second-phase particles exist in structural alloys. Fracture of these particles or decohesion of the particle-matrix interface by deformation causes cavities to nucleate (Goods and Brown, 1979). Void nucleation is material-dependent and is highly influenced by particle size and particle volume fraction (Gurland, 1972). Larger particle sizes and volume fractions favour the nucleation process. This can be explained in terms of an increased number of dislocations surrounding large particles making the plastic relaxation easier and also the increase of interactions between the secondary plastic zones of particles at high particle volume fractions (Goods and Brown, 1979). The nucleation mechanism can be stress controlled, where the relieved shear stresses at the grain boundary give rise to a normal stress, causing the formation of a cavity (Argon et al., 1975). However, some other researchers believe that it is the local state of deformation that determines the cavity initiation (Lindley et al., 1970).

Growth: The nucleated cavities will expand simultaneously under increased plastic deformation. Models have been developed to study the growth of an isolated cylindrical (McClintock, 1968) or spherical void (Rice and Tracey, 1969, Budiansky et al., 1982) in a remotely stress strain rate field. All these models indicate an exponential amplification of the growth rate by the stress triaxiality. Later,

constitutive equations for porous ductile materials have been obtained based on homogenization theory (Gurson, 1977, Rousselier, 1987).
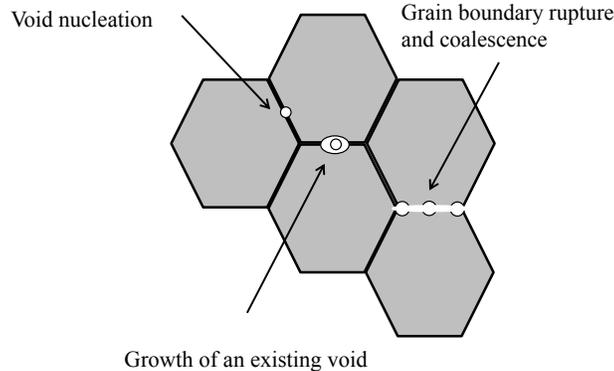
Figure 2.4- Damage mechanisms (Michel, 2004).

Coalescence: Void growth is followed by micro-void coalescence which occurs when the matrix between neighbouring voids is sufficiently thinned down and a certain critical condition for local plastic instability is reached; this can usually be seen during necking. Unlike nucleation and growth, when the plastic deformation is dilatational and macroscopically homogenous, coalescence involves non homogeneous, localised deformation. Different approaches have been developed for the prediction of coalescence onset and process. In some studies coalescence has been considered as accelerated growth and a critical value of the void volume fraction, f_c, which is a material-dependent constant, has been suggested as a criterion for coalescence onset. The coalescence process is then modelled by a simple heuristic method to decrease the stress to zero or to accelerate the damage evolution up to complete loss of the stress carrying capacity (Ruggieri et al., 1996, Sun et al., 1992, Xia et al., 1995, Tvergaard and Needleman, 1984). However, in some other studies it is argued that f_c cannot be a material parameter and varies with stress state and can be calculated from micromechanically based models (Thomason, 1990). Good reviews on prediction of coalescence onset and process are done by Benzerga and Leblond (2010) and Pardeon (2010).

17

2.3 Gurson model

Micromechanical models and computational tools have been developed widely to analyse and predict ductile fracture. The first quantitative analysis of the growth mechanics of an isolated void in a nonlinear material under triaxial loading conditions was given by McClintock for cylindrical voids (1968). The initiation of fracture in McClintock's model is based on the condition of voids touching the adjacent voids and despite its simplifications exhibits some fundamental features of ductile fracture. A later famous approach is the Rice and Tracy growth model (1969) which is based on dilatational growth analysis of a single spherical void in a material under a uniform stress state. Needleman (1972) applied Rice and Tracy's approach to a doubly periodic square array of circular cylindrical voids under plane strain conditions. Needleman's work inspired Gurson (1977) who later proposed an approach for obtaining an approximate yield surface for a material containing voids of long cylindrical and also spherical shapes.

The derivation of the Gurson yield function for spherical voids is based on the mechanics of growth of voids under an axisymmetric stress state. The yield function of the Gurson model depends on the von Mises, or equivalent stress, σ_{eq}; the hydrostatic stress, σ_h; the flow stress of the matrix material, $\bar{\sigma}$, and void volume fraction, f:

$$\Phi(\sigma_{eq}, \bar{\sigma}, f, \sigma_m) = \frac{\sigma_{eq}^2}{\bar{\sigma}^2} + 2f\cosh\left(\frac{3\sigma_h}{2\bar{\sigma}}\right) - (1 + f^2) = 0 \qquad 2.2$$

Figure 2.5 illustrates the Gurson model's yield surfaces for different values of void volume fraction f. The yield function is reduced to the classical von Mises yield criterion if $f = 0$.

The increase in void volume fraction could be due to the growth of existing voids and also due to the nucleation of new voids. The Gurson model treats the void

volume change due to both cases as that of a single void using homogenisation. Figure 2.6 shows the homogenisation process in the Gurson model.
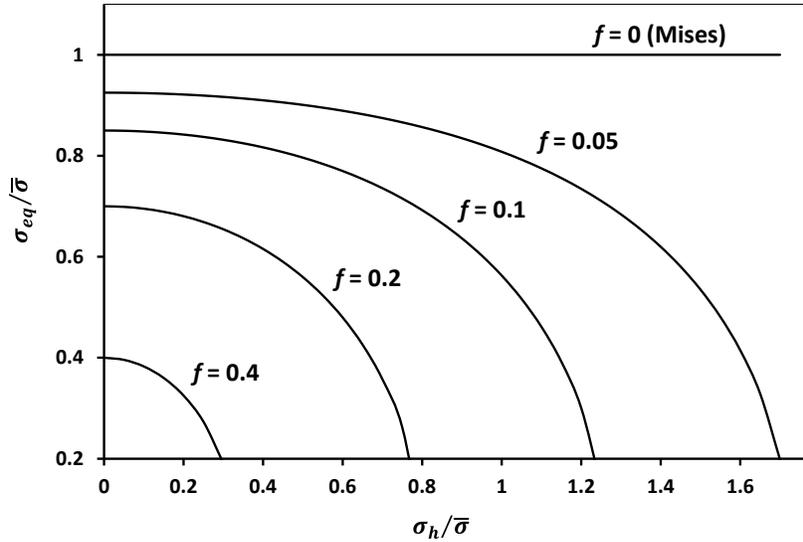
Figure 2.5 *- Schematic of the yield surface in the $\sigma_h/\bar{\sigma}$ - $\sigma_{eq}/\bar{\sigma}$ plane.*
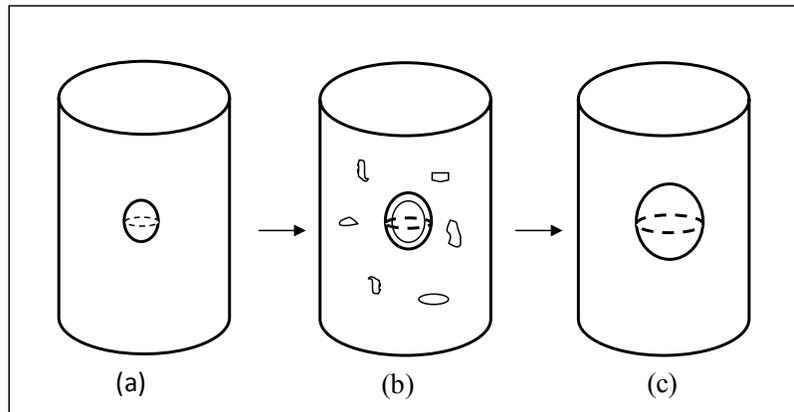
Figure 2.6 *- Homogenization of void nucleation and growth in the Gurson model.*

The rate of change of void volume fraction, $\dot{f}$, was defined as:

$$\dot{f} = \dot{f}_g + \dot{f}_n \tag{2.3}$$

where $\dot{f}_g$ is the rate of growth of existing voids and is defined as:

$$\dot{f}_g = (1 - f)\dot{\varepsilon}_{ij}^p I_{ij} \tag{2.4}$$

where $\dot{\varepsilon}_{ij}^p$ is a plastic strain rate and I_{ij} is the second-order unity tensor. The void nucleation rate $\dot{f}_n$ is defined as:

$$\dot{f}_n = A_\varepsilon \left(\frac{EE_t}{E - E_t}\right)\dot{\varepsilon}_{eq}^p + \frac{1}{3}B_\sigma(\sigma_h) \tag{2.5}$$

where A_ε and B_σ are the void nucleation intensities depending on the plastic strain increment and increase of hydrostatic stress (σ_h), respectively; E is Young's modulus and E_t is the current tangent modulus of the matrix material.

The parameters A_ε and B_σ are chosen so that void nucleation follows a normal distribution (Chu and Needleman, 1980). Thus, for nucleation controlled by the plastic strain, the parameters are specified by:

$$A_\varepsilon = \left(\frac{1}{E_t} - \frac{1}{E}\right)\frac{f_N}{s_N\sqrt{2\pi}}\exp\left\{-\frac{1}{2}\left(\frac{\varepsilon_M^p - \varepsilon_N}{s_N}\right)^2\right\}, \qquad B_\sigma = 0 \tag{2.6}$$

while for the stress controlled nucleation:

$$A_\varepsilon = B_\sigma = \left(\frac{1}{E_t} - \frac{1}{E}\right)\frac{f_N}{\sigma_Y s_N\sqrt{2\pi}}\exp\left\{-\frac{1}{2}\left(\frac{\bar{\sigma} + 1/3\,\sigma_h - \sigma_N}{s_N\sigma_Y}\right)^2\right\} \tag{2.7}$$

where σ_Y is the yield stress, ε_M^p is the effective plastic strain of the matrix material, f_N is the volume fraction of void nucleating particles, ε_N and σ_N are the mean strain and stress for nucleation, respectively, and s_N is the corresponding standard deviation.

Tvergaard (1981, 1982) studied the bifurcation prediction in the original Gurson model with his numerical analysis and found that the original Gurson model resulted in bifurcation at loads that were too small and strains that were twice the values he found numerically. Tvergaard therefore modified the Gurson model by introducing two fitting parameters, q_1 and q_2, in Gurson's yield function.

$$\Phi\left(\sigma_{eq}, \bar{\sigma}, f^*, \sigma_m\right) = \frac{\sigma_{eq}^2}{\bar{\sigma}^2} + 2\,q_1 f^* \cosh\left(\frac{3q_2\sigma_h}{2\bar{\sigma}}\right) - \left(1 + q_3 f^{*2}\right) = 0 \qquad 2.8$$

He found that $q_1 = 1.5$, $q_2 = 1.0$ and $q_3 = q_1{}^2$ resulted in better accuracy. Although the parameters q_1, q_2 and q_3 have been assumed as material constants by some researchers, their dependency on other factors such as triaxiality and initial void volume fraction has also been discussed. This matter will be investigated in the following chapters.

Another modification to the original Gurson model was related to modelling the complete loss of stress carrying capacity. The Gurson model predicts the total loss of load carrying capacity at $f = 100\%$. This critical void volume fraction, f_c, is unrealistically high and means the total disapearance of the material. The original Gurson model therefore can only model the homogeneous deformation phase (nucleation and growth) and is unable to predict localization and ductile fracture.

The function $f^*(f)$ was introduced by Tvergaard and Needleman (1984) to model the rapid loss of stress-carrying capacity and therefore to account for the coalescence.

$$f^* = \begin{cases} f & \text{if } f \leq f_c \\ f_c + \dfrac{f_u^* - f_c}{f_F - f_c}(f - f_c) & \text{if } f > f_c \end{cases} \qquad 2.9$$

where f_F is void volume fraction at final fracture and $f_u^* = {}^1\!/_{q_1}$.

Based on experimental (Goods and Brown, 1979) and numerical results (Andersson, 1977) Tvergaard and Needleman (1984) have chosen the values $f_c = 0.15$ and $f_c = 0.25$. This model is called Gurson-Tvergaard-Needleman (GTN) model and is used frequently in engineering applications.

Gurson's model has been criticised by Thomason (1998) as being based on a large number of adjustable parameters that have no physical significance in terms of the mechanism of ductile fracture. Thomason stated that unlike nucleation and growth which involve homogeneous deformation, coalescence involves a localised deformation and therefore should be modelled in a different approach. Thomason has questioned the assumption of ductile fracture being initiated at a critical void volume fraction and believes this assumption ignores the influence of the intervoid-matrix geometry and the magnitude of the normal stress on the critical conditions for the onset of coalescence. Thomason (1990) has proposed a mathematical plastic-limit load approach to predict the initiation of void coalescence.

Looking at damage recovery as the reverse of damage evolution, the void closure stage would correspond to the reverse growth; the coalescence stage hence will be of no interest. The criticisms of Thomason regarding the coalescence stage therefore will not affect the use of the GT model to predict void closure.

2.4 Void elimination

Void elimination can be investigated through two stages. The first stage is the void closure when the void shrinks to a slit (crack) as a result of compression. By the end of this stage the surfaces of the slit however are not bonded properly and will be separated easily by a small tension. A strong bond which has the mechanical properties of the original material is obtained in the second stage when the surfaces of the void are bonded.

2.4.1 Void closure

Figure 2.7 illustrates the void evolution from spherical shape to crack. Parameters such as the temperature during deformation, the temperature gradient, the maximum compressive stress and the deformation rate, have an influence on the void-closing process.

The problem of void closure has been studied by many researchers and great progress has been made to identify the parameters that affect the behaviour of voids during hot rolling or forging. Some of these studies generally have dealt with the combination of parameters such as the effects of reduction (decrease in the thickness of the work-piece), die geometry (die's shape and size) and the rate of deformation (Dudra and Young-Taek, 1990, Park and Yang, 1997, Stahlberg, 1986, Stahlberg et al., 1980, Wallero, 1985). Stahlberg et al. (1986) showed that void closure occurs more rapidly for large reductions and that small void fractions are easier to remove than large void fractions. They also pointed out that void volume fraction may increase at the beginning of deformation due to large tensile stresses but will decrease after a particular reduction, leading to void closure.

The effect of some rolling geometry parameters on void closure was also inspected in some studies. It was shown that large values of the ratio of roll radius to work-piece thickness will favour the void closure process (Wallero, 1985). Research performed by Dudra and Young-Taek (1990) studied the effect of die geometry (flat, double V and FML dies) and ingot geometry (square and round billets) on void closure in forging. A superior consolidation of porosity was observed for the V-shaped dies. This was due to the highest amount of hydrostatic stress and effective strain associated with these dies. In terms of ingot geometry, the voids in the square billet closed at a lower reduction but at higher press loads than the round billet.
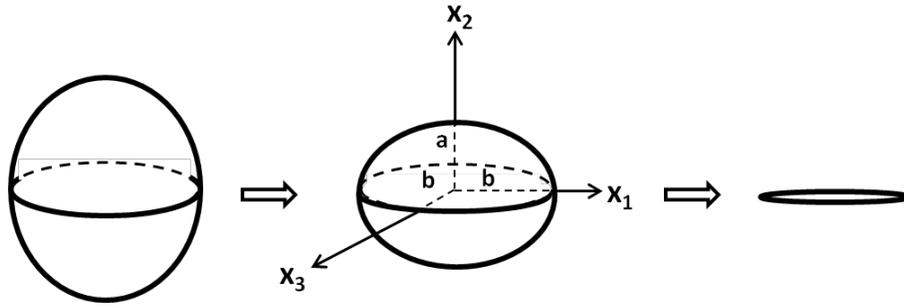
Figure 2.7 *- Void evolution from spherical shape to crack (Zhang et al., 2009).*

Other parameters such as pressure, temperature and time have also been investigated. An experimental investigation by Han et al (1997) on crack recovery of 20MnMo steel indicated that damage recovery could be achieved without plastic deformation and only with increased heating temperature and prolonged holding time. However, another experimental study on crack healing in 1045 steel showed that even after 120 min heat treatment at 1100°C, micro-voids were left in the crack healing area (Wei et al., 2006).

It is well known that void closure is more easily and quickly obtained if plastic deformation takes place under the influence of hydrostatic pressure (Keife and Stahlberg, 1980). Wang et al. (1996) found that pore closure occurs at a certain level of hydrostatic pressure and speeds up at high temperatures. It was also shown that the holding period of the pressure in the compressive state affects the degree of healing. Figure 2.8 shows the cell models used in FE simulations in some studies.

A recent study by Zhang et al. (2009) proposed a criterion for void closure in large ingots during hot forging. Zhang et al. investigated the effects of the Norton exponent, the stress triaxiality and the remote effective strain on void closure. It was found that the volumetric strain rate of the void had a direct relationship with the stress triaxiality and Norton exponent. At the early stage of void closure the rate of void closure was high and a smaller remote effective strain was required. This was due to the increased stress triaxiality level and Norton exponent. At the final stage of

24

void closure, however, the rate of void closure decreased. This was explained in terms of the instability of the void at this stage and the contact of void internal surfaces which may lead to some interactions such as friction and therefore can slow down the void closure process.

(a)

(b)

(c)
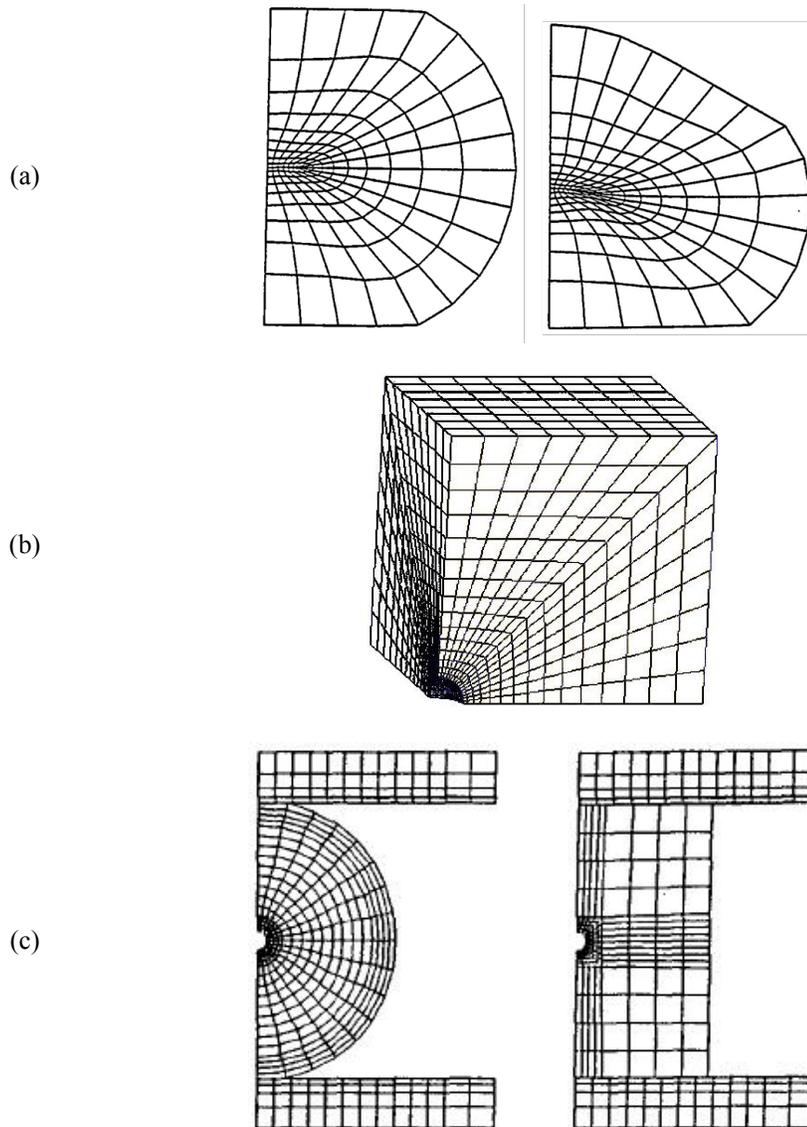
Figure 2.8- Cell models for different FE simulations: (a) A round billet compressed by flat dies (shown on the left) and Bottom-Flat-Top-V dies (shown on the right) (Park and Yang 1997), (b) Square ingot (Zhang et al. 2009), (c) Round ingot (shown on the left) and square ingot (shown on the right) compressed by flat dies (Dudra and Young-Taek, 1990).

Complicating matters, the porosity distribution in as-cast steel is generally highly nonuniform; a billet contains more, larger pores at the centre than the surface, therefore the size and distribution cannot be predicted precisely. The voids in different positions experience varying boundary conditions and will close under different loading conditions. To investigate the closure of a void at a specific position in a steel billet, the stress state at that point can be identified and triaxiality obtained by FE modelling of the rolling process. The conditions for plastic closure of a void in that position could therefore be predicted by determining a closure load versus triaxiality relationship obtained from an RVE model. The influences of temperature and strain rate on the void closure under different triaxialities are also important. These matters have been studied in the following chapters in more detail.

2.4.2 Void healing

The mechanism of healing substantially depends on parameters such as temperature, pressure and the duration of load-temperature application. The effect of these parameters on pore healing has been investigated by several researchers. A study on sintering of grain boundary cavitation in Nickel Chromium alloy investigated the effect of hydrostatic compressive stress on the rate of change of cavity radius for cavities with and without internal gas (Stevens and Flewitt, 1979). It was shown that increasing the hydrostatic compressive stress caused a substantial decrease in time to remove the cavity.

Temperature significantly speeds up the cavity removal process (Figure 2.9). Comparison of the change in cavity radius for cavities with internal gas and cavities with internal vacuum showed no difference until a critical size was reached for the cavities with internal gas, where no further radius reduction was observed. This was explained in terms of the internal pressure balancing the hydrostatic pressure and the surface energy.
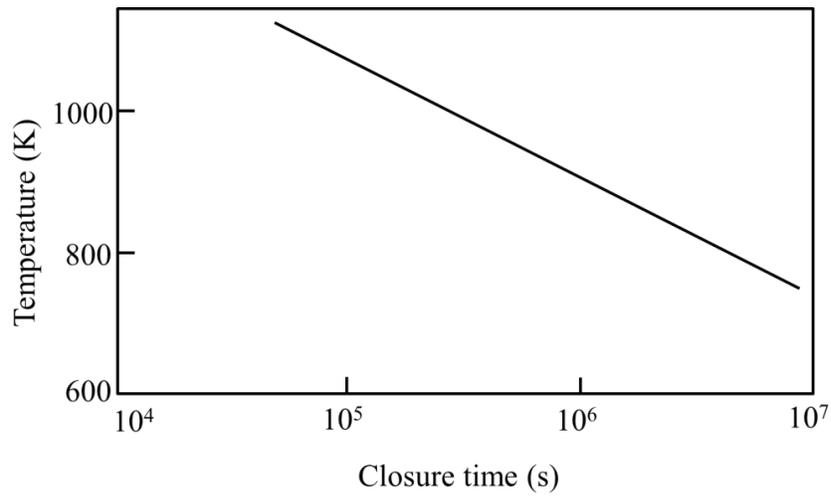
Figure 2.9- *Time to close a cavity for various temperatures (Stevens and Flewitt, 1979).*
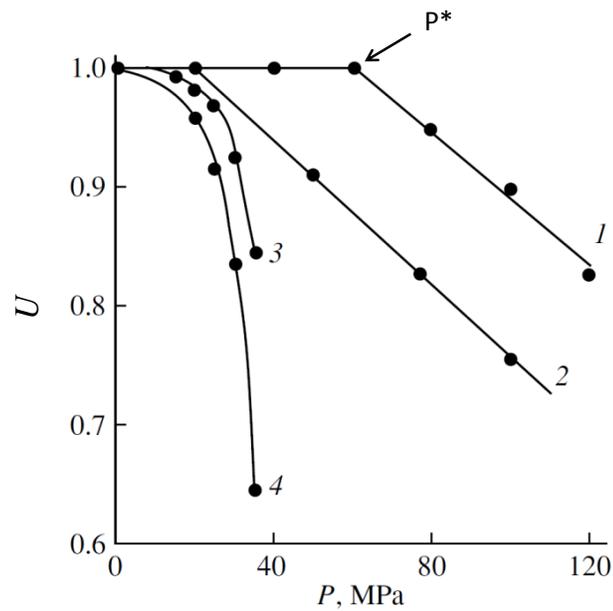
Figure 2.10- *Pressure dependence of zinc porosity at temperatures of (1) 18 ℃, (2) 93 ℃, (3) 210 ℃, and (4) 260 ℃ (Petrov and Razuvaeva, 2005). U is the degree of porosity and is defined as $U = \varphi \bar{R}^3$ (where φ is the void concentration and $\bar{R}$ is the average void size).*

Another study on porosity elimination in Zn showed that the decrease in porosity started from a threshold pressure, *P**, and then decreased with increasing pressure (Petrov and Razuvaeva, 2005). Figure 2.10 shows the degree of porosity of the samples, $U = \varphi \bar{R}^3$ (where φ is the void concentration and $\bar{R}$ is the average void size), as a function of pressure for Zn in the temperature range 18–260$^{\text{o}}$C. The threshold pressure *P** decreased noticeably with rising temperature. It was also observed that at room temperature the duration of applied pressure did not affect the porosity but for a fixed healing time, *t*, the increase in temperature caused a significant acceleration in the process of void healing.

The void healing concept has also been the subject of study for applications such as metal joining (Guo and Ridely, 1987, Derby and Wallach, 1984) and crack healing (Wang et al., 2008, Wei et al., 2004, Wang et al., 1996). In metal joining when two solid surfaces are brought in contact, the bond interface contains an array of voids with irregular shapes. Diffusion bonding processes therefore play an important role in bringing the surfaces of these voids to within an atomic spacing in order for a metallic bond to form over the whole bonding interface. Similar processes occur for crack healing, when after the tip blunting stage, one or more arrays of cylindrical pore channels form upon crack splitting. These pores will then evolve into spherical voids via a Rayleigh instability (Gupta, 1978).

2.4.3 Diffusion phenomenon

Diffusion bonding involves the joining of two solid surfaces under a low pressure and high temperature. The temperature range is usually between *50%* and *80%* of the absolute melting point of the material being bonded. To avoid large deformation of the joined surfaces the interfacial pressure must be sufficiently low, and therefore times of up to 100 min may be required to obtain satisfactory bonding (Derby and Wallach, 1982).

The application of diffusion bonding is advantageous over welding or adhesive bonding because of the fact that no melting or additional materials are involved, thus there is no discontinuity in the microstructure and in properties at the bonding sites. Additionally low bonding pressure will prevent any distortion of the components during bonding (Pilling and Ridely, 1987). Upon correct execution of the process, the bond produced will have indistinguishable properties from those of the bulk metal and the bond line will have continuous metallurgical structure.

Diffusion as the phenomenon of material transport by atomic motion is ideally represented by net flux, *J*. As given by Fick's first law, *J* is proportional to the concentration gradient, dc/dx, and diffusion coefficient, *D*:

$$J = -D(dc/dx) \qquad\qquad 2.10$$

The kinetics of many thermally activated processes follows the Arrhenius equation, expressing the relation between the process rate, *g*, and temperature (Pontikis, 1990):

$$g = F\exp(-E_A/RT) \qquad\qquad 2.11$$

where *F* is a frequency factor, *R* the gas constant, E_A is the activation energy and T the absolute temperature.

Similarly, diffusion obeys the Arrhenius equation. Therefore D can be expressed as:

$$D = D_o\exp(-Q/RT) \qquad\qquad 2.12$$

where *Q* is activation enthalpy and D_o is a pre-exponential factor.

Vacancies or other inhomogeneities within a metal are essential for atomic diffusion. Several different diffusion processes have been identified for solid state diffusion. The main diffusion processes are surface diffusion, grain boundary diffusion and lattice diffusion.

Boundary diffusion: When a pressure is uniformly applied perpendicular to a grain boundary, the stress in the boundary will redistribute around boundary cavities. A stress gradient results in a chemical potential gradient which causes diffusion.

Surface diffusion: The difference in curvature between void tip and void surface results in a chemical potential gradient described by the Gibbs-Thomson relationship. This causes a flow of matter along the void surface. It should be noted that diffusion around the surface area occurs not only through a thin layer of material (surface diffusion) but also through the volume adjacent to the surface (volume diffusion). This material is assumed to be taken uniformly from the void surface. This type of diffusion only alters the shape of the void; the volume remains constant.

Lattice diffusion: Lattice diffusion refers to atomic diffusion within the crystal lattice which occurs by substitutional mechanisms. In substitutional lattice diffusion the atom changes places with another atom. This relies on the existence of vacancies in the crystal lattice.

All sintering mechanisms and the associated coefficients have been summarised in Table 2.1, where D_s is the surface diffusion coefficient, D_v is the vacancy diffusion coefficient, D_{gb} is the grain boundary diffusion coefficient and D_p is the coefficient of pipe diffusion along dislocations.

Table 2.1*- Solid state diffusion mechanism (Lumley, 2008).*

Mechanism	Transport part	Associated diffusion coefficient	Source of matter transported to neck
1	Surface diffusion	D_s	Surface
2	Lattice Diffusion	D_v	Surface
3	Vapour Transport	None	Surface
4	Boundary Diffusion	D_{gb}	Grain Boundary
5	Lattice Diffusion	D_v	Grain Boundary
6	Lattice Diffusion	D_p	Dislocations

Since these processes have different activation energies, each will have a particular diffusion coefficient at different temperatures. Figure 2.11 shows the paths for matter transport to the neck region by diffusion mechanisms summarised in Table 2.1.
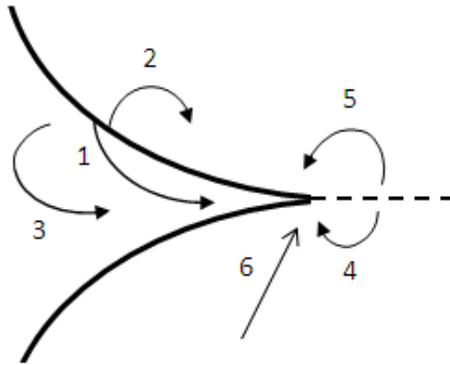
Figure 2.11- Alternative paths for matter transport to the neck region (Lumley, 2008).
Numbers refer to the mechanism numbers in Table 2.1.

2.5 Conclusions

The problem of porosity formation in the process of continuous casting of steel was explained. The characteristics of this type of porosity and the factors causing it were discussed. The adverse effects of porosity on mechanical properties of different materials investigated by other researchers were reviewed and the urgent need for methods of achieving damage healing and porosity elimination was highlighted. The void elimination process was investigated as two stages: void closure and healing. The parameters influencing the void closure process including the parameters that affect the behaviour of voids during hot rolling or forging (e.g. amount of reduction, die geometry and deformation rate) and also other parameters such as pressure, temperature and time, were discussed. It was shown that plastic deformation occurring under the influence of hydrostatic pressure at high temperature favours

void closure. The factors influencing the healing stage were also reviewed. Similar to the closure process the positive effects of increasing the hydrostatic compressive stress and temperature on the cavity removal process were outlined. It was suggested that porosity decrease may start at a threshold pressure which decreases noticeably with rising temperature.

The importance of diffusion phenomena on bonding the void surfaces was explained and the diffusion bonding mechanisms contributing to void healing were reviewed.

The next chapter will explain how diffusion bonding mechanisms have been combined in different models to predict void elimination.

2.6 References

Andersson, H. 1977. Analysis of a model for void growth and coalescence ahead of a moving crack tip. *Journal of the Mechanics and Physics of Solids,* 25**,** 217-233.

Argon, A. S., Im, J. A. & Safoglu, R. 1975. Cavity formation from inclusions in ductile fracture. . *Metallurgical and Materials Transactions A,* 6**,** 825-837.

Benzerga, A. A. & Leblond, J.-B. 2010. Ductile fracture by void growth to coalescence. *Advances in Applied Mechanics,* 44**,** 169-305.

Bourcier, R., Koss, D., Smelser, R. & Richmond, O. 1986. The influence of porosity on the deformation and fracture of alloys. *Acta Metallurgica,* 34**,** 2443-2453.

Budiansky, B., Hutchinson, J. W. & Slutsky, S. 1982. Void growth and collapse in viscous solids. *In:* HOPKINS, H. G. & SEWELL, M. J. (eds.) *Mechanics of solids*. Oxford: Pergamon Press.

Chu, C. C. & Needleman, A. 1980. Void nucleation effects in biaxially stretched sheets. *Journal of Engineering Materials and Technology,* 102**,** 249-257.

Derby, B. & Wallach, E. R. 1982. Theoretical model for diffusion bonding. *Metal Science,* 16**,** 49-56.

Derby, B. & Wallach, E. R. 1984. Diffusion bonds in copper. *Materials Science,* 19**,** 3140-3148.

Dudra, S. P. & Young-Taek, I. 1990. Analysis of void closure in open-die forging. *International Journal of Machine Tools and Manufacture,* 30**,** 65-75.

Goods, S. H. & Brown, L. M. 1979. The nucleation of cavities by plastic deformation. *Acta Metallurgica,* 27**,** 1-15.

Guo, Z. X. & Ridely, N. 1987. Modelling of diffusion bonding of metals. *Materials Science and Technology,* 3**,** 945-953.

Gupta, T. K. 1978. Instability of cylindrical voids in alumina. *Journal of the American Ceramic Society,* 61**,** 191-195.

Gurland, G. 1972. Observations on the fracture of cementite particles in a spherodised 1.05% C steel deformed at room temperature. *Acta Metallurgica,* 20**,** 735-741.

Gurson, A. L. 1977. Continuum theory of ductile rupture by void nucleation and growth: Part I-Yield criteria and flow rules for porous ductile media. *Journal of Engineering Materials and Technology:Transactions of the ASME,* 99**,** 2-15.

Han, J., Zhao, G. & Cao, Q. 1997. Internal crack recovery of 20MnMo steel. *Science in China Series E: Technological Sciences,* 40**,** 164-169.

Hardin, R. & Becketmann, C. 2007. Effect of porosity on the stiffness of cast steel. *Metallurgical and Materials Transactions A,* 38**,** 2992-3006.

Keife, H. & Stahlberg, U. 1980. Influence of pressure on the closure of voids during plastic deformation. *Journal of Mechanical Working Technology,* 4**,** 133-143.

Lemaitre, J., A 1996. *A course on damage mechanics*, Berlin

Lindley, T. C., Oates, G. & Richards, C. E. 1970. A critical of carbide cracking mechanisms in ferride/carbide aggregates. *Acta Metallurgica*, 18**,** 1127-1136.

Lumley, R. 2008. Self healing in aluminium alloys. *Self Healing Materials*, 100**,** 219-254.

McClintock, F. A. 1968. A Criterion for ductile fracture by the growth of holes. *Journal of Applied Mechanics*, 35**,** 363-371.

Michel, B. 2004. Formulation of a new intergranular creep damage model for austenitic stainless steels. *Nuclear Engineering and Design*, 227**,** 161-174.

Monroe, R. 2005. Porosity in castings. *AFS Transactions-American Foundary Society* 113**,** 519-546.

Needleman, A. 1972. Void growth in an elastic-plastic medium. *Journal of Applied Mechanics*, 39.

Pardoen, T., Scheyvarts F, Simar A, Tekoğlua, C. & Onckb, P. 2010. Multiscale modelling of ductile failure in metalic alloy. *Comptes Rendus Physique*, 11**,** 326–345.

Park, C. Y. & Yang, D. Y. 1997. Modelling of void crushing for large-ingot hot forging. *Journal of Materials Processing Technology*, 67**,** 195-200.

Petrov, A. I. & Razuvaeva, M. V. 2005. Initial stage of void and crack healing in polycrystalline metals under uniform compression. *Physics of the Solid State*, 47**,** 907–912.

Pilling, J. & Ridely, N. 1987. Solid state bonding of superplastic AA 7475. *Materials Science and Technology*, 3**,** 353-359.

Pontikis, V. 1990. *Thermally activated processes in solids*, Dordrecht, Boston and London, Kluwer Academic Publishers.

Rice, J. R. & Tracey, D. M. 1969. On the ductile enlargement of voids in triaxial stress fields. *Journal of the Mechanics and Physics of Solids*, 17**,** 201-217.

Rousselier, G. 1987. Ductile fracture models and their potential in local approach of fracture. *Nuclear Engineering and Design*, 105**,** 97-111.

Ruggieri, C., Panontin, T. L. & Dodds, R., H. 1996. Numerical modeling of ductile crack growth in 3-D using computational cell elements. *International Journal of Fracture*, 82**,** 67-95.

Stahlberg, U. 1986. Influence of spread and stress on the closure of a central longitudinal hole in the hot rolling of steel. *Journal of Mechanical Working Technology*, 13**,** 65-81.

Stahlberg, U., Keife, H. & Lundberg, M. 1980. A study of void closure during plastic deformation. *Journal of Mechanical Working Technology*, 4**,** 51-63.

Stevens, R. A. & Flewitt, P. E. J. 1979. The role of hydrostatic pressure on the sintering of creep cavities in a nickel-2% chromium alloy. *Acta Metallurgica*, 27**,** 67–77.

Sun, D. Z., Kienzler, R., Voss, B. & Schmitt, W. 1992. Application of micro-mechanical models to the prediction of ductile fracture. . *In:* AL, S. N. A. E. (ed.) *Fracture mechanics: 22nd Symposium (Volume II), ASTM STP 1131*. Philadelphia: American Society for Testing and Materials.

Surappa, M., Blank, E. & Jaquet, J. 1986. Effect of macro-porosity on the strength and ductility of cast Al-alloy 7If-0.3mg effect of macro-porosity on strength and ductility of cast alloy Al-Si-7 0 3 Mg. *Scripta Materialia,* 20**,** 1281-1286.

Thomas, B. G. 2001. Continuous casting. *Encyclopedia of Materials: Science and Technology,* 2**,** 1595-1599.

Thomason, P. F. 1990. *Ductile fracture of metals*, Oxford, Pergamon Press.

Thomason, P. F. 1998. A view on ductile fracture modelling. *Fatigue and Fracture of Engineering Materials and Structures,* 21**,** 1105-1122.

Tvergaard, V. 1981. Influence of voids on shear band instabilities under plane strain conditions. *International Journal of Fracture,* 17**,** 389-407.

Tvergaard, V. 1982. On localization in ductile materials containing spherical voids. *International Journal of Fracture,* 18**,** 237-252.

Tvergaard, V. & Needleman, A. 1984. Analysis of the cup-cone fracture in a round tensile bar. *Acta Metallurgica,* 32**,** 157-169.

Wallero, A. 1985. Closing of a central longitudinal pore in hot rolling. *Journal of Mechanical Working Technology,* 12**,** 233-242.

Wang, A., Thomson, P. F. & Hodgson, P. D. 1996. A study of pore closure and welding in hot rolling process. *Journal of Materials Processing Technology,* 60**,** 1996.

Wang, H., Huang, P. & Li, Z. 2008. Crack and Void Healing in Metals. 100**,** 255-277.

Wei, D., Han, J., Jiang, Z. Y., Lu, C. & Tieu, A. K. 2006. A study on crack healing in 1045 steel. *Journal of Materials Processing Technology,* 177**,** 233-237.

Wei, D., Han, J., Tieu, K. & Jiang, Z. 2004. Simulation of crack healing in BCC Fe. *Scripta Materialia,* 51**,** 583-587.

Xia, L., Shih, C. F. & Hutchinson, J., W 1995. A computational approach do ductile crack growth under large scale yielding conditions. *Journal of the Mechanics and Physics of Solids,* 43**,** 389-413.

Zhang, E. & Wang, B. 2005. On the compressive behaviour of sintered porous coppers with low to medium porosities-Part I: Experimental study. *International Journal of Mechanical Sciences,* 47**,** 744-756.

Zhang, X., Cui, Z., Chen, W. & Li, Y. 2009. A criterion for void closure in large ingots during hot forging. *Journal of Materials Processing Technology,* 209**,** 1950-1959.

Chapter 3

Existing diffusive void growth/shrinkage models

Despite similarities in assumptions and approaches taken in developing void shrinkage models, the existing models predict different void shrinkage rates. The discrepancies in these models are related to their origin. These models have been reviewed in detail here and their applicability for the purpose of this study has been discussed.

3.1 Factors influencing the void shrinkage models

In this section the parameters which cause discrepancies in existing void shrinkage models are briefly explained.

3.1.1 Bonding mechanisms involved

There are many possible bonding mechanisms, where each transports material to deform the void surface and therefore shrink it in a different way. The differences in existing void shrinkage models are due to the origins of the models. Some existing void shrinkage models (Guo and Ridely, 1987, Hancock, 1976) were obtained from void growth models, where void shrinkage was considered as negative void growth. Some other models (Derby and Wallach, 1982, Hill and Wallach, 1989), on the other hand, originated from powder sintering models. Modelling of powder sintering process involves two stages. The initial stage deals with low densification (D < 0.9) and involves the growth of initial contact between particles. Here D is the density of the material relative to water. The final stage, however, deals with higher densification (D > 0.9), where the compact is modelled as a homogeneous solid containing spherical voids. The final stage therefore is more interesting in void shrinkage models (Artz et al., 1983, Ashby, 1974).

The discrepancies in void shrinkage rates predicted by these models can be explained in terms of different mechanisms used in each model. Different models propose different predominant mechanisms. A mechanism which is considered as dominant in one model can be ignored in another one. A good example is the consideration of surface diffusion in the computation of void shrinkage rate. The contribution of surface diffusion to the shrinkage process has been ignored by some researchers stating that it mainly affects the void shape and does not reduce the volume of the

cavity (Guo and Ridely, 1987). Some other models however have taken into account surface diffusion as an important mechanism (Chuang and Rice, 1973, Needleman and Rice, 1980, Wang and Li, 2004).

Table 3.1*- Existing models of diffusion mechanisms (Takahashi and Inoue, 1992)*

Process	Models	Methods of combining diffusion mechanisms and characteristics of model.
Diffusion bonding	Derby and Wallach Hill and Wallach	All diffusional mechanisms are taken into consideration; parallel operation is assumed; void shape is elliptical.
	Guo and Ridley	Parallel operation is assumed; void shape is lenticular; diffusion distance is varied; rigid collapse is considered.
Powder sintering	Ashby	All diffusional mechanisms are independent; complete parallel operation is assumed.
	Arzt et al	Only interface source mechanisms (interface and volume diffusion) are considered; surface source mechanisms are ignored.
Void growth	Beere and Speight	Interface diffusion only; diffusion distance varied; effects of creep rigid collapse are considered.
	Hull and Rimmer	Series operation of surface and interface diffusion.
	Chen and Argon	Series operation of surface and interface diffusion.
	Needleman and Rice	Series operation of surface and interface diffusion; effect of plastic creep on diffusional flow; a numerical model.

The void shrinkage models originating from sintering models usually take all diffusion mechanisms into consideration (Derby and Wallach, 1982, Guo and Ridely, 1987, Hill and Wallach, 1989). Derby & Wallach (1982) have considered all possible mechanisms which are shown in Figure 3.1. The void shrinkage models derived from void growth models however mostly consider surface and interface mechanisms and ignore creep and plastic deformation.

The existing diffusion bonding models are shown in Table 3.1. The origin and characteristics of the models are also presented.

1) Surface diffusion from surface sources to a neck

2) Volume diffusion from surface sources to a neck

3) Diffusion along the bond interface from interfacial sources to a neck

4) Volume diffusion from interfacial sources to a neck

5) Power law creep deforming the ridge

6) Plastic yielding deforming the ridge
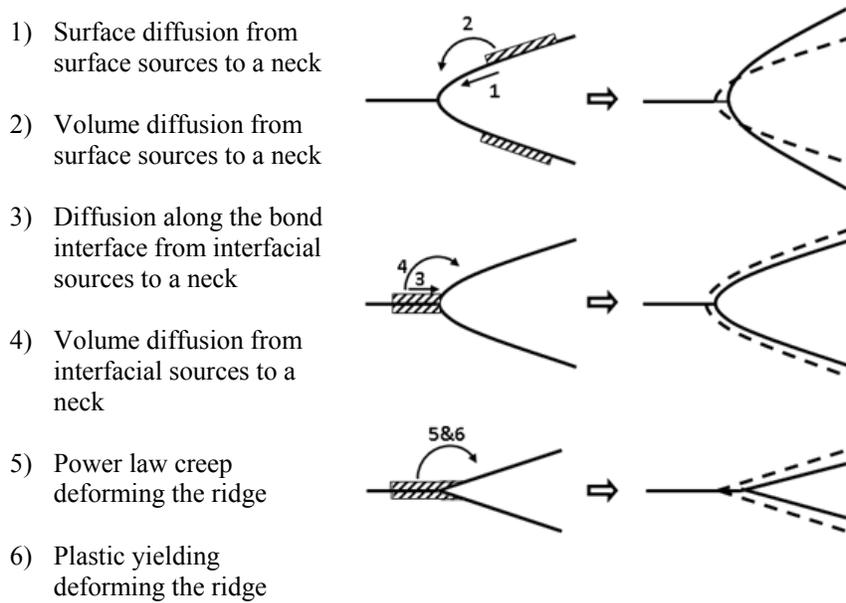
Figure 3.1 - *The effect of bonding mechanisms on void shape (Derby and Wallach, 1982).*

3.1.2 Void Geometry

Different void geometries have been considered for the modelling of void healing. A popular assumption is that surface diffusion is fast enough to keep the void shape spherical (Speight and Beere, 1975, Sun, 2002, Beere and Speight, 1978).

Some models have considered the contact surface to contain triangular ridges of certain height and wavelength, which then gradually shrink into cylindrical channels with radial symmetry (circular cross section) (Derby and Wallach, 1982). Wang and Li (2004) have suggested that as atoms migrate along the void surface and grain-boundary, the grain-boundary void changes its shape and volume and the equilibrium shape of a grain-boundary void is a spheroid.

A cylindrical shape has also been assumed for some models (Chen and Argon, 1981, Pilling and Ridely, 1987). A cylindrical pipe shape with either lenticular or rhombic cross section has been considered in some cases. Gue and Ridly (1987) assumed a lenticular initial void shape. Takahashi on the other hand applied a rhombic geometry as the initial void shape and a lenticular geometry during diffusion. Figure 3.2 shows the schematic illustration of different void shapes used in different models.

3.1.3 Void position

Healing models have been developed for voids within the grain boundary (Wang and Li, 2003, Sun, 2002) and also for voids on the grain boundary (Chen and Argon, 1981, Rimmer, 1959, Wang and Li, 2003, Beere and Speight, 1978). The final shape of a void within a grain is a result of the competition between the difference in the elastic energy stored in the solid and the surface energy. For a grain boundary void, however, this would depend on the variation in the elastic energy, the surface and grain boundary energies (Wang and Li, 2004). For voids within the grain boundary the primary mechanisms would be lattice diffusion and surface diffusion. For spherical voids, where surface diffusion is fast, lattice diffusion is the dominant mechanism and the surface diffusion is ignored (Wang and Li, 2003)
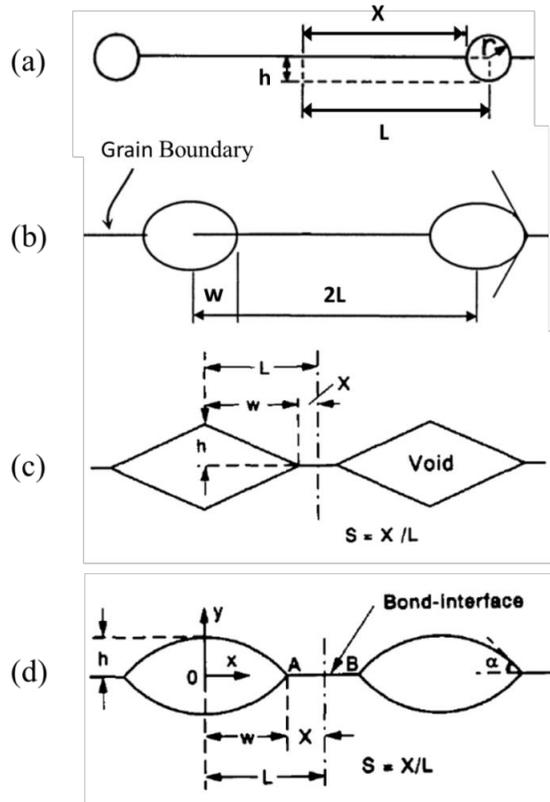
Figure 3.2- *The schematic illustration of void shapes.(a) Cylindrical voids (Derby and Wallach, 1982)(b) Spheroidal voids (Wang and Li, 2004)(c) Rhombic voids (Takahashi and Inoue, 1992)(d) Lenticular voids (Takahashi and Inoue, 1992).*

3.1.4 Series and parallel operations

Looking at models for void shrinkage on a bond interface, it can be seen that parallel operations have usually been considered. In parallel operations all possible bonding mechanisms act independently in such a way that their contribution to bonding can be added together to obtain the total amount of bonding. Figure 3.3(a) shows a parallel operation where surface source mechanisms operate independently of interface source mechanisms. As shown in Figure 3.3a diffusion of atoms toward the void tip reduces the radius of curvature at the tip ($r_{c_{tip}}$) compared to the radius of curvature at the top of the void ($r_{c_{top}}$).

In void growth models it is assumed that interface and surface diffusion occur in series. In series operations, depending on the values of surface diffusion and grain boundary diffusion coefficients (D_s and D_{gb}, respectively), the rate controlling mechanism can be identified. If $D_s \gg D_{gb}$, surface diffusion is much faster than grain boundary diffusion. This means that grain boundary diffusion is rate controlling and surface diffusion can be ignored in calculating the healing time. This suggests equal radius of curvature at the void top and void tip ($r_{c_{tip}} = r_{c_{top}}$) (see Figure 3.3b). However, if $D_{gb} \gg D_s$ (grain boundary diffusion is much faster than surface diffusion), surface diffusion is rate controlling, resulting in a convex surface at the void tip and the moving of the sink point towards the void surface (Figure 3.3c).
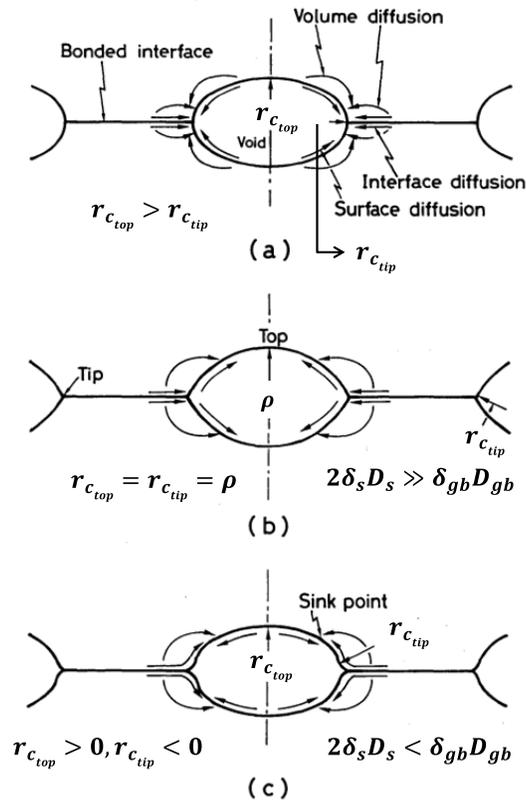
Figure 3.3*- Schematic illustrations of diffusion flux of atoms under compressive stress: parallel operation (model A); series operation (model B), when $D_{gb} \ll D_s$; series operation (model C), when $D_{gb} \gg D_s$ (Takahashi and Inoue, 1992).*

3.2 Models for diffusive void growth

Some void shrinkage models were developed from void growth models, where void shrinkage was considered as negative void growth. Below, some of these models are explained.

3.2.1 Hull and Rimmer model

The diffusive growth of grain boundary cavities has received continuous attention. Hull and Rimmer (1959) were one of the first to propose a diffusive mechanism for growth of an isolated cavity in a material under an applied external stress. Their model suggests that vacancies for void growth are formed by tension across the grain boundaries. This requires that the activation energy for failure is that of grain boundary diffusion. It is assumed that the vacancy chemical potential has a defined value at the cavity and the sum of the local stresses over the grain boundary equals the applied stress. This model also assumes that the vacancies are created uniformly over the entire boundary.

The rate of growth of a void is determined by the gradient of chemical potential, $\nabla\mu$, since the diffusion flux, J, is given by 3.1:

$$J = \frac{D_{gb}}{K\mathrm{T}\Omega}\nabla\mu \qquad\qquad 3.1$$

where K is Boltzmann's constant, Ω is the atomic volume, T is the absolute temperature and D_{gb} is the grain boundary diffusion coefficient. The potential in the grain boundary is:

$$\mu = -\sigma_n\Omega \qquad\qquad 3.2$$

where σ_n is the normal stress across the boundary and has a maximum value of $\sigma - P_h$ on the grain boundary (P_h is hydrostatic pressure). On the surface of a void $\mu = -2\gamma\Omega/\rho$, where γ is the surface tension of the metal and ρ is void radius (see page xxi).

For voids on the grain boundary with centre to centre separation distance, b (see Figure 3.4), a rough approximation would be:

$$\nabla\mu \sim -\frac{\Omega}{b}(\sigma - P_h - \frac{2\gamma}{\rho})$$

3.3

$$J \sim -\frac{D_{gb}}{KTb}(\sigma - P_h - \frac{2\gamma}{\rho})$$

3.4

The void surface area from which diffusion occurs is $2\pi\rho\delta_{gb}$, where δ_{gb} is the grain boundary width. Since diffusion flux, J = total flow rate of atoms/area, therefore, total flow rate of atoms = $2\pi\rho\delta_{gb} \cdot J$. The total flow rate of atoms is the number of moles of material transferred per second (mol/s). To obtain the void volume growth, the flow rate is multiplied by atomic volume, Ω. Therefore, neglecting the surface tension term γ, the rate of void volume growth is given by:

$$\frac{dV}{dt} \sim total\ flow\ rate \cdot \Omega \sim \frac{(D_{gb}\delta_{gb})(\sigma - P_h) \cdot 2\pi\rho\Omega}{KTb}$$

3.5

where Ω is the atomic volume.

Hull and Rimmer obtained the rupture time, t_r, for an array of equally sized nuclei lying on a square lattice of size b (see Figure 3.4):

$$t_r = \frac{KTb}{2\pi\Omega(D_{gb}\delta_{gb})(\sigma - P_h)}\left\{\frac{b^2}{8} + \frac{\gamma b}{(\sigma - P_h)} + \frac{4\gamma^2}{(\sigma - P_h)^2}ln\left[\frac{\frac{1}{2}b(\sigma - P_h) - 2\gamma}{\rho(\sigma - P_h) - 2\gamma}\right]\right\}$$

3.6

Figure 3.5 shows the expected variation of t_r with $(\sigma - P_h)$ where the curve has an almost hyperbolic shape down to $(\sigma - P_h) = 2\gamma/\rho_0$. At this point the applied stress is balanced by the surface tension of the void nucleus, so that the voids never grow.

Later other researchers (Raj and Ashby, 1975, Beere and Speight, 1975, Beere and Speight, 1978, Riedel, 1987) obtained similar equations for diffusion-controlled cavity growth. An analysis by Speight and Harris (1967) considered a cavity occupying a circular area of boundary and highlighted the effect of relative void

spacing. Their study showed that the Hull and Rimmer equation underestimated the growth rate for both small and large void spaces. This conclusion was later verified by Weertman (1973) for widely spaced voids.
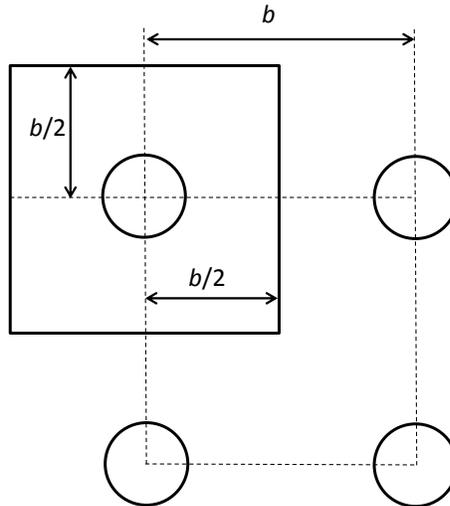
Figure 3.4*- Cross section through voids lying in the plane of the grain boundary (Hull and Rimmere, 1959).*
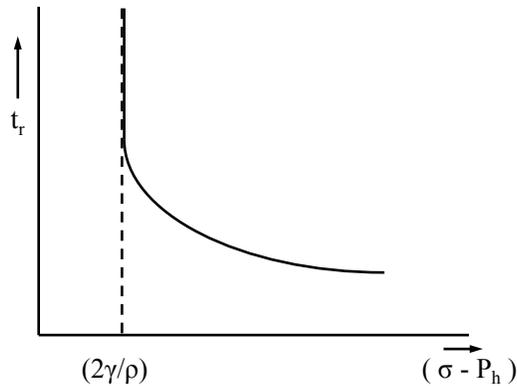
Figure 3.5*- Variation of t_r with $(\sigma - P_h)$.*

3.2.2 Beere and Speight model

Combining matrix creep and diffusion for cavity growth has been the subject of study for many researchers (Edward and Ashby, 1978, Cocks and Ashby, 1982a, Cocks and Ashby, 1982b, Chen and Argon, 1981, Kassner and Hayes, 2003, Lee et

al., 1993). Beere and Speight (Beere and Speight, 1978, Beere and Speight, 1975) modelled the total void growth rate as the sum of diffusion and creep. Their model predicted the cavity growth on a grain boundary. Beere and Speight were the first to investigate the effect of creep on the diffusion distance. Their approach was later adopted by other researchers such as Chen and Argon (1981) and Guo and Ridley (1987) to predict the diffusion bonding process. The model of Chen and Argon (1981) will be explained in the next section.

Beere and Speight's (1978) growth model assumes the grain boundary containing a cavity is divided into two regions (Figure 3.6). In region I, near the cavity (see Figure 3.6), vacancies form uniformly and diffuse to the cavity. In region II, away from the cavity, no vacancies are formed but dislocation creep matches the displacement rate in region I, resulting in an increased growth rate. They indicated that local creep causes the diffusion distance to be less than $c - \rho$ (see Figure 3.6) and found a relationship between the extent of diffusion region, l_I, and a dimensionless quantity Λ:

$$\Lambda = \frac{G \Omega D_{gb} \delta_{gb}}{B K T c^3} \qquad 3.7$$

where G is the shear modulus, Ω is the atomic volume, K is the Boltzmann constant and B is a defined as:

$$B = A \left(\frac{G b_v D_v}{K T} \right) \qquad 3.8$$

where A is a dimensionless constant and b_v is Burger's vector (see pages xviii-xxii for definition of other parameters).

Λ depends on the ratio of diffusion rate to creep rate (the creep rate is defined by a relationship of the type $\dot{\varepsilon} = B(\sigma/G)^n$). Figure 3.7 shows this relationship for copper crept at P = 20.67 MPa and with a grain boundary radius of 10 μm. It is evident from this figure that decreasing the diffusion rate compared to the creep rate (decreasing Λ) reduces the diffusion distance, l_I, or in other words, l_I approaches ρ; whereas increasing the diffusion rate, the diffusion region, l_I, approaches c.
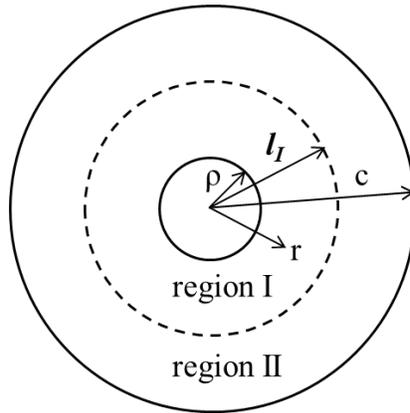
Figure 3.6- *A cavity of radius ρ is associated with an area of grain boundary of radius c. Vacancies created in Region I diffuse to the cavity (Beere and Speight, 1978).*
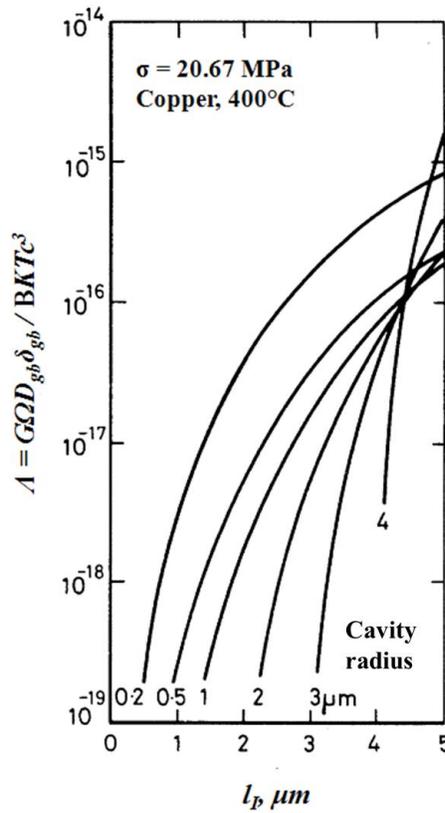
Figure 3.7- *Variation of diffusion region specified by l_I, with Λ (Λ represents ratio of grain boundary diffusivity to creep rate)(Beere and Speight, 1978).*

The normal stress acting across the boundary at the cavity equals $2\gamma/\rho$, where γ is the surface energy and ρ is the cavity radius. In region I, the normal stress (σ_n^I) increases with distance from the cavity; this explains the diffusion of vacancies to the cavity. The stress rate in this region is approximately linearly proportional to the diffusion rate. In region II the deformation rate is associated with a high power of normal stress (σ_n^{II}) due to dislocation creep, but since no vacancy diffusion exists the stress gradient is zero and the stress profile is flat. This is shown in Figure 3.8.
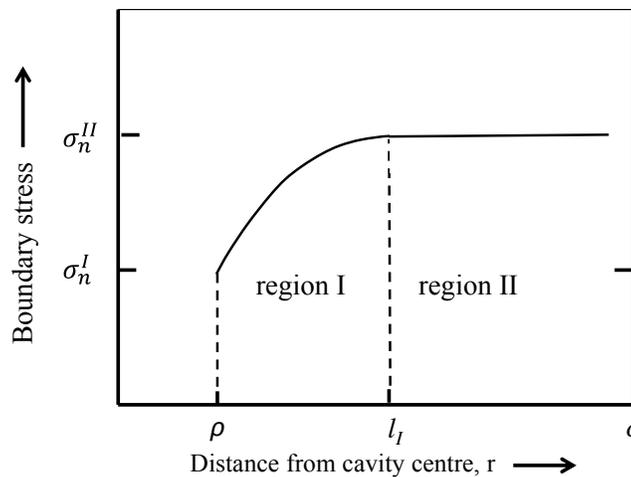
Figure 3.8- Normal stress acting across the grain boundary.

Beere and Speight developed solutions at high and low dislocation creep rates. For high diffusion rates the cavity growth rate depends on diffusion and is independent of power law creep. As the diffusion rate reduces, the dislocation creep rate becomes dominant in the growth of large cavities; the growth of smaller cavities however still remains partly dependent on the diffusion mechanism.

Figure 3.9 shows the sensitivity of the normalised cavity growth rate to stress. At high stresses the cavity growth rate is proportional to σ^n, where n is the stress exponent of power law creep. At low stresses the normalised cavity growth rate shows a linear relationship with stress (dotted lines on this figure).

The cavity growth rate is calculated by adding the rates due to the low- and high-stress asymptotic behaviour.

$$\frac{dv}{dt} = \frac{8K_1}{\left[4ln\frac{c}{\rho} - \left(1 - \frac{\rho^2}{c^2}\right)\left(3 - \frac{\rho^2}{c^2}\right)\right]} + 2\pi\rho^3\dot{\varepsilon}_c\left(\frac{c^2}{c^2 - \rho^2}\right)^n \qquad 3.9$$

where K_1 is defined as $K_1 = \pi\Omega D_{gb}\delta_{gb}(\sigma - 2\gamma/\rho)/KT$. D_{gb} and δ_{gb} are grain-boundary diffusivity and grain boundary thickness and $\dot{\varepsilon}_c$ is creep strain rate. The derivation of equation 3.7 can be found in (Beere and Speight, 1978)
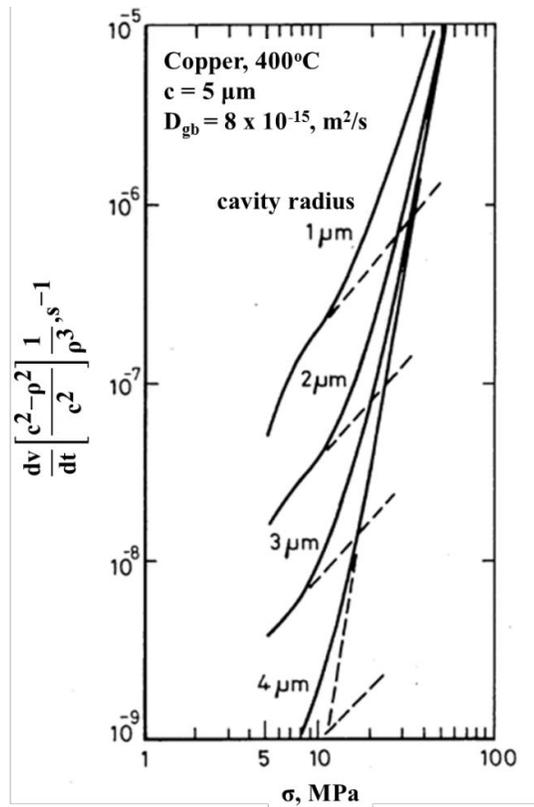
Figure 3.9*- Stress sensitivity of the normalised cavity growth rate. At low stresses the normalised growth rate is proportional to stress to the power of unity (shown as dotted lines). At high stresses the normalised cavity growth rate depends on σ^n (Beere and Speight, 1978).*

Generally, using the Beere and Speight model, finding a real value solution for the diffusion distance under normal conditions is difficult. Beere and Speight have used

numerical solutions to a set of simultaneous equations which leads to the selection of a diffusion distance. Chen and Argon (1981) however have managed to obtain an equation for the diffusion distance. A comparison of Beere and Speight model with other diffusive void growth models is done in section 3.2.4.

3.2.3 Chen and Argon model

Chen and Argon (1981) used a coupled creep-diffusion approach to develop a model for growth of grain boundary cavities. They considered the effect of creep rate on diffusion distance, l_I, initially defined by Beere and Speight (1978). Beere and Speight considered $0 < l_I < c$ (see Figure 3.6) but Chen and Argon assumed $\rho < l_I < c$ and obtained:

$$l_I = \left(D_{gb}\delta_{gb}\Omega\sigma/K\mathrm{T}\dot{\varepsilon}\right)^{1/3} \qquad\qquad 3.10$$

They ignored the surface energy term, γ, and replaced the grain boundary radius, c, in equation 3.9, with the creep-flow-modified diffusion distance $(l_I + \rho)$ and obtained:

$$(\rho^3\dot{\varepsilon})^{-1}\frac{dv}{dt} = 8\pi(l_I/\rho)^3\left[4ln\frac{c}{\rho} - \left(1 - \frac{\rho^2}{(\rho + l_I)^2}\right)\left(3 - \frac{\rho^2}{(\rho + l_I)^2}\right)\right]^{-1} \qquad 3.11$$

Grain-boundary diffusion in region l_I dominates the power-law creep in transporting matter to the grain boundary.

Chen and Argon questioned the quasi-equilibrium void shape assumption made in other models and highlighted the observation of flat, disk-shaped cavities in experiments. They therefore obtained growth rate equations for both quasi-equilibrium and crack-like voids.

A comparison of Chen and Argon (1981) model with other diffusive void growth models is done in the next section.

3.2.4 Other models

In addition to the diffusive void growth models explained so far, some other models also exist. Some of these models are developed by assuming power law creep only (Hancock, 1976) whereas some others are only based on grain boundary and volume diffusion processes (Raj and Ashby, 1975). Coupled diffusion-creep processes in parallel to rigid collapse and diffusion has also been considered (Guo and Ridely, 1987). The void volume change rate as a function of bonding pressure for these models and also for the Beere and Speight (1978) and Chen and Argon model (1981) is shown in Figure 3.10. On this figure dotted lines represent power law creep based models and solid lines represent diffusion (coupled diffusion-creep or diffusion only) based models. The Guo and Ridley (1987) model shown in Figure 3.10 is a coupled diffusion-creep model. Individual constituents of this model are shown in Figure 3.10, where the dotted line represents the power law creep part and the solid line represents the diffusion part of this model. The creep part of this model (shown by dotted line in Figure 3.10) is out of agreement with other creep-based models in this figure (e.g. Hancock (1976)). This is explained in terms of different assumptions made by these models. For example, the Gue and Ridley (1987) model assumes that during the initial stages of bonding, the void edge (tip) adjacent to the bonded interface deforms preferentially. This is referred to as the void crushing effect. Hancock (1978) however, neglects the crushing effect.

It is evident from Figure 3.10 that at low pressures the Beere and Speight (1978) model, Raj and Ashby (1975) model and diffusional part of Guo and Ridley (1987) (solid line) are in good agreement. As explained in section 3.2.2, at low pressures Beere and Speight model neglects the effect of creep and the rate of void growth is only diffusion based. This explains the good agreement of this model with diffusion based models of Raj and Ashby (1975) and Guo and Ridley (1987) on this figure.

At higher bonding pressures, however, the Beere and Speight (1978) model deviates from those models and becomes in good agreement with creep based models such as those of Chen and Argon (1981) and Hancock (1976).

Figure 3.10 also shows a relatively good agreement between the predictions of the coupled diffusion-creep model of Chen and Argon (1981) and the creep model of Hancock (1976). A very good match can also be seen between Chen and Argon (1981) and the Beere and Speight (1978) models at high pressures. The difference between the predictions of these models at low pressures can be explained in terms of the negligibility of creep in the Beere and Speight model at low pressures.
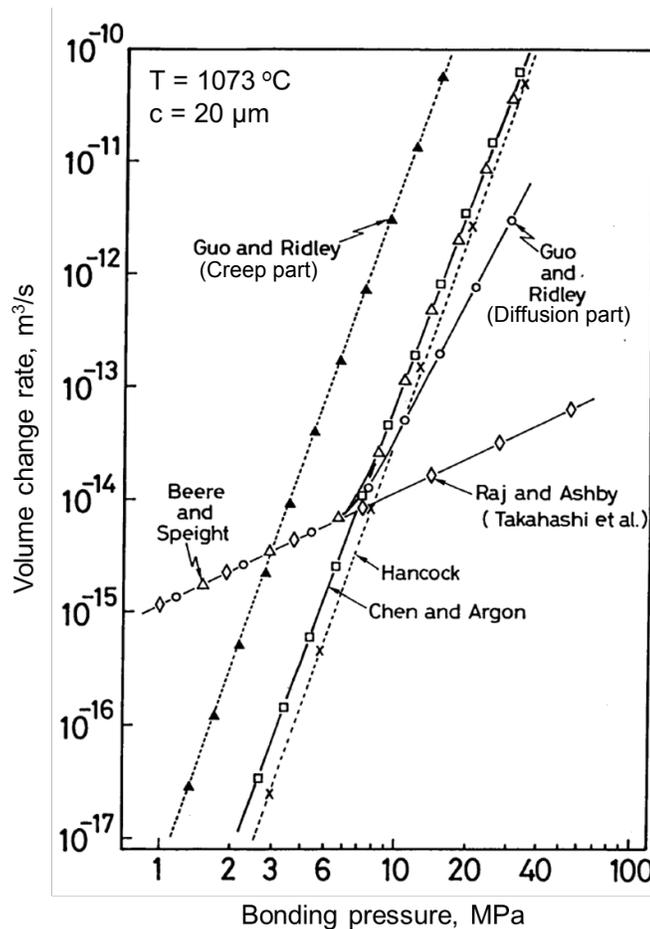
Figure 3.10*- Void volume change rate as a function of pressure for different models. Dotted lines represent power law creep based models and solid lines represent diffusion (coupled diffusion-creep or diffusion only) based models(Takahashi and Inoue, 1992)*

3.3 Models for diffusive void shrinkage

Some void shrinkage models consider the diffusion-bonding process to be analogous to that of pressure sintering. These models apply sintering based mechanisms for elimination of interfacial voids in metal joining and crack healing. Some other void shrinkage models, on the other hand, are solely based on diffusion mechanisms. Some of these models are explained in details here.

3.3.1 Derby and Wallach model

The model of Derby and Wallach (1982, 1984) assumes the surfaces in contact consist of long triangular ridges. As bonding progresses, the channels between the ridges shrink and cylindrical symmetry is obtained. The time from initial surface contact to the point where cylindrical symmetry is reached is modelled as stage I and the final collapse of the cylinders is defined as stage II (Figure 3.11).
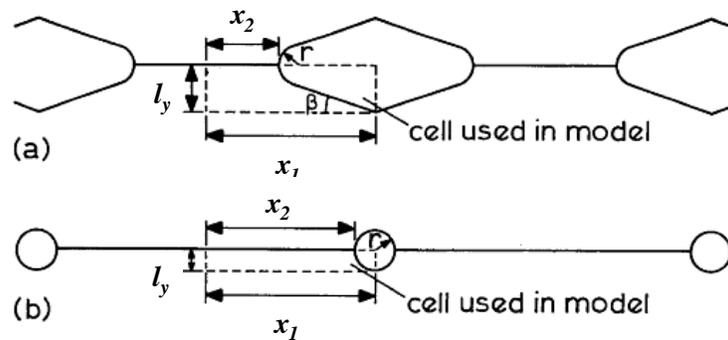
Figure 3.11*- Model geometry used by Derby & Wallach, (a) Stage I (b) Stage II. (Derby and Wallach, 1982).*

Derby and Wallach (1982) defined three categories of bonding mechanisms including surface source mechanisms, bond-line source mechanisms and bulk deformation mechanisms (Figure 3.1). Plastic deformation categorised as a bulk deformation process occurs at the initial stages of bonding, when the contact area is very small. Since it happens at a very high speed compared to the other mechanisms,

it has been considered as an instantaneous process. Diffusion from surface sources to the neck is explained in terms of mass transferred from the void surface to the curved neck by surface diffusion (mechanism 1) and volume diffusion (mechanism 2). Using Kuczynski's (1956) approach for shrinkage of pores during sintering, the surface and volume flux are defined as:

Surface diffusion:

$$\dot{V}_1 = \frac{2\Omega\delta_s D_s}{rK\text{T}}\frac{\gamma}{r}\left(1 - \frac{r}{x_1 - x_2}\right)$$
3.12

Volume diffusion:

$$\dot{V}_2 = \frac{2\Omega D_v}{K\text{T}}\frac{\gamma}{r}\left(1 - \frac{r}{x_1 - x_2}\right)$$
3.13

The diffusion along the bond interface (mechanism 3) and volume diffusion from the interfacial source (mechanism 4) are defined as:

Grain boundary diffusion:

$$\dot{V}_3 = \frac{3\Omega\delta_{\text{gb}}D_{gb}}{2KTx_2}\left(P\frac{x_1}{x_2} - \frac{\gamma}{x_2} - \frac{\gamma}{r}\right)$$
3.14

Volume diffusion:

$$\dot{V}_4 = \frac{3\Omega\text{r}D_v}{Kx_2}\left(P\frac{x_1}{x_2} - \frac{\gamma}{x_2} - \frac{\gamma}{r}\right)$$
3.15

where x_1, x_2 and r (bond neck radius) are shown in Figure 3.11.

At high temperatures and low pressures, bulk deformation of surface ridges occurs by power law creep. Derby and Wallach (1982) used Wilkinson and Ashby's (1975) approach to model power law creep for stages I and II.

$$\dot{V}_{5_I} = \frac{\sqrt{3}SA_c {x_2}^2 \tan\beta}{2\left[1-\left(\frac{x_2}{x_1}\right)^{\frac{2}{n}}\right]^n}\left\{\frac{2\left[\left|P\left(\frac{x_1}{x_2}-1\right)-\frac{\gamma}{x_2}\right|\right]}{\mu n}\right\}^n \qquad 3.16$$

$$\dot{V}_{5_2} = \frac{\sqrt{3}\pi r^2 SA_c}{2\left[1-\left(\frac{r}{x_1}\right)^{\frac{2}{n}}\right]^n}\left[\frac{2\left(\left|P\frac{x_1}{x_2}-\frac{\gamma}{x_2}\right|\right)}{\mu n}\right]^n \qquad 3.17$$

In equations 3.16 and 3.17, S is the sign of the bonding pressure, negative if compressive, A_c is the power law creep pre-exponential constant, β is the angle between a surface ridge and the horizontal, μ is the chemical potential and n is the power law creep exponent.

Considering a parallel operation, this model assumes that the overall rate of bonding will be as the sum of the individual rates of all mechanisms.

Hill and Wallach (1989) later identified the problem of discontinuity arising when the void shape changed from triangle (stage I) to cylinder (stage II). A vapour phase mass transport was later added to the model to reduce the effect of the discontinuity on the void closure rate.

Derby and Wallach's model is in good agreement with experimental results when power law creep is the dominant bonding mechanism. However, when diffusion bonding, especially surface diffusion, becomes dominant, deviations arise between prediction and experimental results. The other problem with this model is the assumption that each mechanism is completely independent of the effect of others, whereas, a quick analysis of the driving forces of each mechanism shows that this assumption may not be true.

3.3.2 Wang and Li model

Despite the assumption of constant void shape made in many void shrinkage models, Wang and Li (2004) considered the variation in the void shape in their model. Figure 3.12 shows the three dimensional model they used for a gas-filled spheroidal void in an elastic solid under triaxial stress.
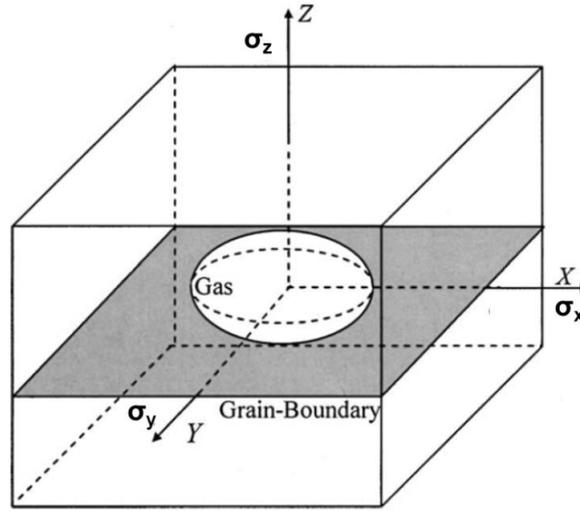
Figure 3.12- The schematic model of a gas-filled spheroidal void in an elastic solid under triaxial stresses $\sigma_x = \sigma_y \leq \sigma_z$ (Wang and Li, 2004).

The shape of a spheroidal void having the same volume as a spherical one of radius ρ can be described by a shape factor, *m:*

$$m = \frac{(\rho_x/\rho_z)^{2/3} - 1}{(\rho_x/\rho_z)^{2/3} + 1}$$

3.18

Therefore:

$$X = Y = \rho \sqrt{\frac{1+m}{1-m}} \cos\theta, \qquad Z = \rho \frac{1-m}{1+m} \sin\theta$$

3.19

In equation 3.19, ρ_x is the radius of the spheroid in *x* and *y* direction and ρ_z is the radius of the spheroid in *z* direction. In this equation $m = 0$ represents a spherical void shape and $m = 1$ corresponds to a crack.

56

The instability of the void is specified by a dimensionless loading parameter, $\psi = (\sigma_3 + p)^2 \rho_o / \gamma E$ which describes the relative importance of elastic energy and surface energy. If ψ exceeds a critical value, ψ_c, the collapse of the spherical void into a crack occurs. Figure 3.13 shows the stability conditions projected on $(m, \alpha\,\psi)$ plane. The heavy solid and the dotted lines correspond to the stable and the unstable equilibrium states, respectively, where α is a volume parameter ($\alpha = \rho/\rho_0$) and ω is the stress ratio defined as $\omega = (\sigma_1 + P)/(\sigma_3 + P)$.
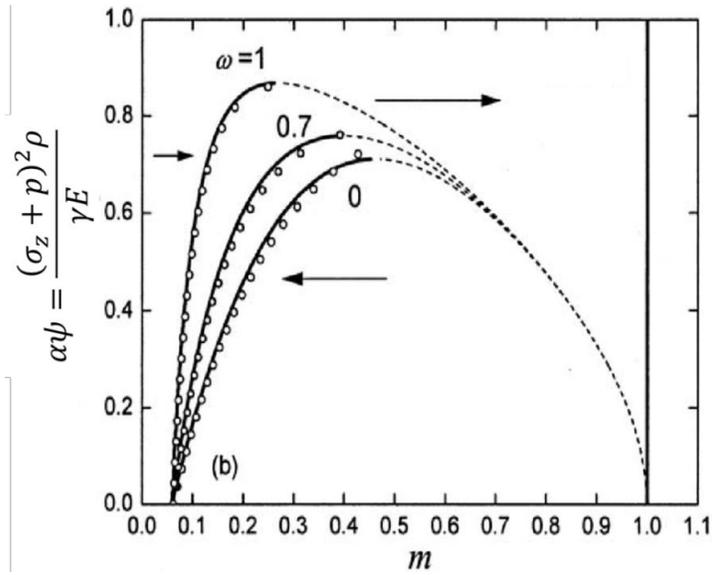
Figure 3.13- *Stability conditions projected on the (m, α ψ) plane. The heavy solid and dotted lines are the numerical solutions to the stable and unstable equilibrium shapes (Wang and Li, 2004).*

In this model an equilibrium state shape parameter, m_e, is defined as a function of ψ. For the elliptical void described by m_e, the curvature of the void apex, k_{apex}, is defined as:

$$k_{apex} = -\frac{\tau^3 + 1}{\rho \tau^{1/2}} \qquad \tau = \frac{1 + m_e}{1 - m_e} \qquad\qquad 3.20$$

The number of atoms removed from the grain boundary per unit time and per unit volume, η also becomes a function of m_e (equation 3.21).

$$\eta = \frac{8D_{gb}\delta_{gb}\gamma}{KTb^2\rho\sqrt{\tau}} \frac{\rho\sigma\sqrt{\tau}/\gamma + x^2\rho\rho_0\,\tau^{3/2}/(\alpha^3\gamma) - (1+\tau^3)(1-x^2\tau)}{3 - 4x^2\tau + x^4\tau^2 + 4ln(x\sqrt{\tau})} \qquad 3.21$$

where $x = \rho/b_1$ (see Figure 3.14).

The removal of atoms from the grain-boundary to the void causes adjacent grains to move close at a rate of $\eta\Omega$, therefore the void volume decreases at a rate of $\pi\xi^2\eta\Omega$. The collection of atoms further decreases the void volume at a rate of $\pi(b_1{}^2 - \xi^2)\eta\Omega$ (ξ is the radius of the spheroid at the void apex as shown in Figure 3.14). So, the total shrinkage rate of the voids can be given as:

$$\frac{dV}{dt} = \eta\pi b_1{}^2\Omega \qquad 3.22$$
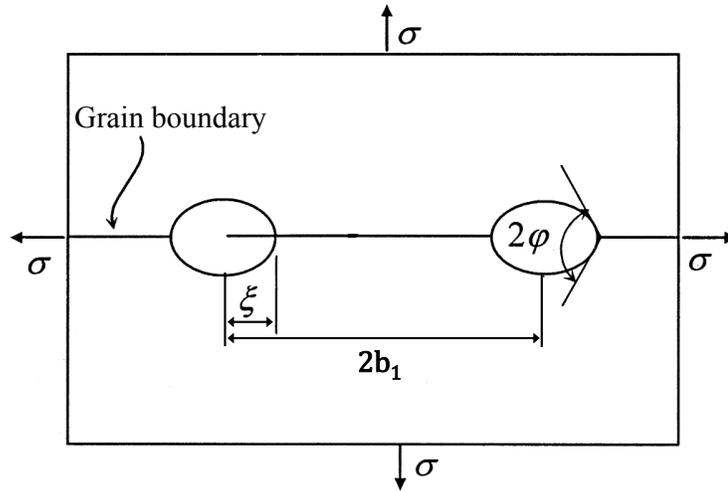
Figure 3.14- *The schematic model for an array of gas-filled voids on a planar grain-boundary under hydrostatic pressure (Wang and Li, 2004) where ξ and φ are the void radius at the void apex and equilibrium angle at the void apex, respectively.*

One of the downsides of the Li and Wang model is the mathematical complication and the very detailed numerical analysis required which are not convenient in practice. It should also be noted that this model is more suitable for an infinite solid containing a single void. For a model with more than one void, the model can be

approximately used only if the initial size of the voids is small enough compared to the void spacing.

3.3.3 Pilling model

The elimination of voids in Pilling's model includes three processes. These processes involve the collapse of the supporting contact area by plastic deformation and creep, diffusion of atoms from the grain boundary to the voids via both volume and interfacial paths and surface diffusion or transfer of mass along the void surface (Pilling and Ridely, 1987, Pilling, 1988, Pilling et al., 1984).

The creep rate in the cylindrical cell is controlled by the effective stress, assuming that the material away from the bond zone is non-deforming and that the external diameter of the cylinder remains constant. The axial plastic collapse of the cylinder causes an increase in both the inner and outer radii and the height of the cylinder. The material in the cylinder walls beyond the outer radius can then be removed and redistributed within the bore of the cylinder, reducing the internal radius and restoring the external radius.
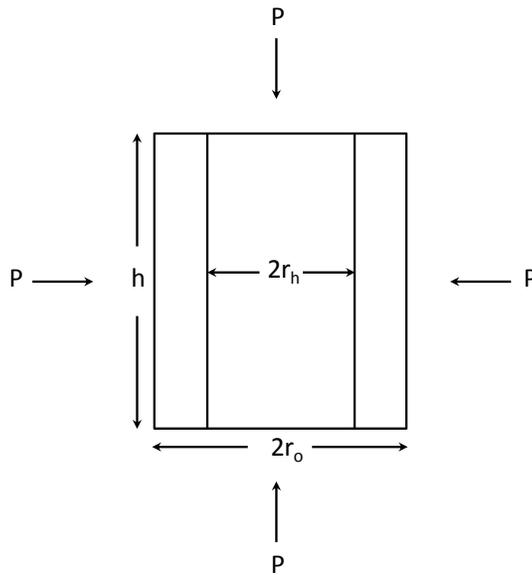
Figure 3.15*- Vertical section through the cylindrical geometry used to define bonding (Pilling, 1988).*

The fractional area, f_h, of non-bonded surface is defined by r_h^2/r_o^2, where r_h and r_o are the inner and outer radii of the cylinder, respectively (see Figure 3.15). Defining the change in fractional area of voids as $df_h = r_i^2/r_o^2 - r_h^2/r_o^2$, the rate of change in the fractional area of non-bonded interface by power law creep and plastic flow can be obtained as,

$$\frac{df_h}{dt}\Big|_{pl} = -2\dot{\varepsilon}_r(1 - f_h) \qquad\qquad 3.23$$

where $\dot{\varepsilon}_r$ is the effective strain rate in the radial direction.

The rate of change in the fractional area of non-bonded interface, as a result of diffusive mass transfer can also be obtained as

$$\frac{df_h}{dt}\Big|_d = -N_i\left(\frac{1}{\pi r_o^2 h_o f_h}\frac{dV}{dt}\right) \qquad\qquad 3.24$$

where,

$$\frac{dV}{dt} = \frac{2\pi D_{gb}\delta_{gb}\Omega\sigma_z}{KT}\frac{1 - f_h}{\ln(1/f_h) - (1 - f_h)/2} \qquad\qquad 3.25$$

In equation 3.24, N_i is the number of interfaces intersecting the void surface ($N_i = $ int$(1 + h_o f_h/d)$, where d is the grain size), D_{gb} is the grain boundary diffusion coefficient, δ_{gb} is the grain boundary thickness, Ω is the atomic volume, K is Boltzmann's constant, σ_z is the axial stress and T is the absolute temperature.

The rate of change of fractional area of non-bonded interface by creep and plastic flow and diffusive flow are shown in Figure 3.16. It is evident that complete interfacial contact is achieved mainly as a result of the time-dependent plastic collapse of the asperities, rather than by diffusion.
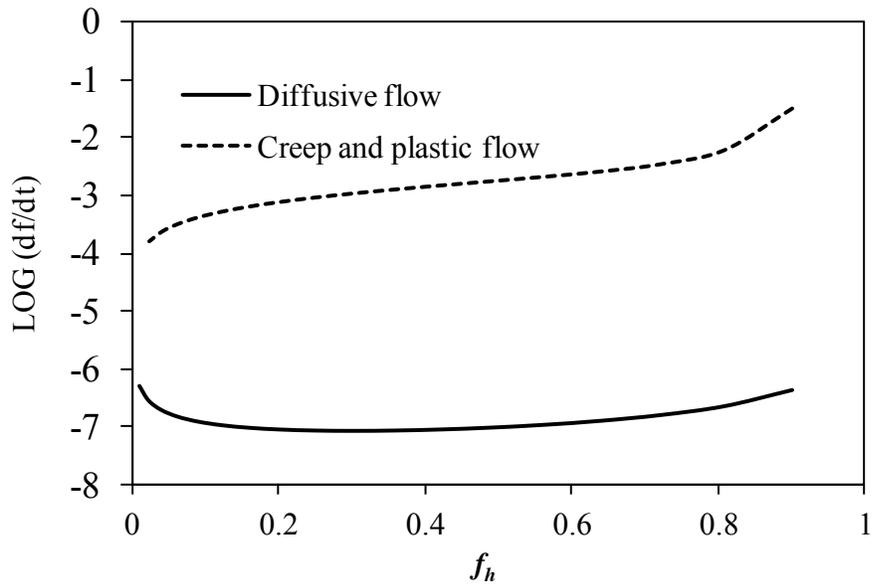
Figure 3.16- *The rate of change in fractional area of non-bonded interface, f_h, with f_h for diffusive flow and creep and plastic flow (P=10MPa, T=1273 K, r_o=28μm).*
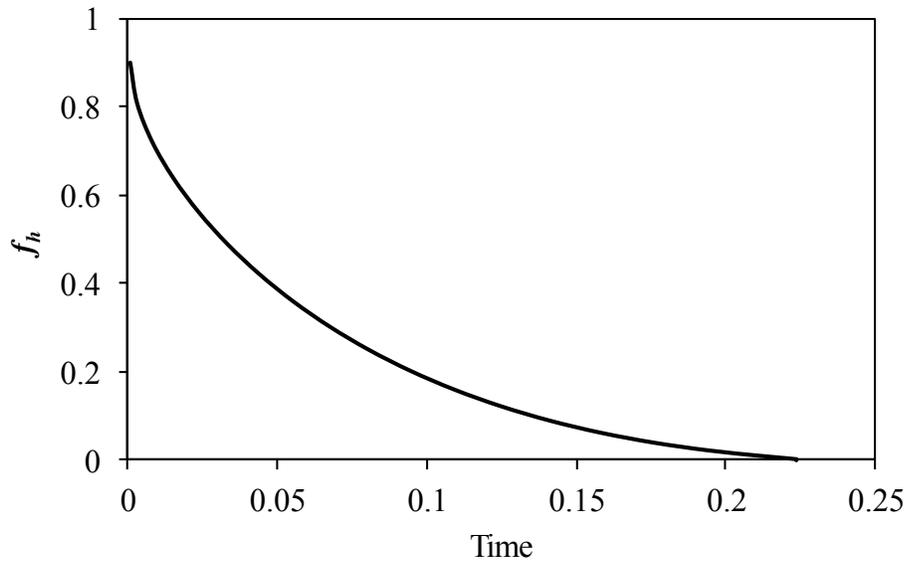
Figure 3.17- *The change in fractional area of non-bonded interface, f_h, versus time (P=10MPa, T=1273K, r_o=28μm).*

61
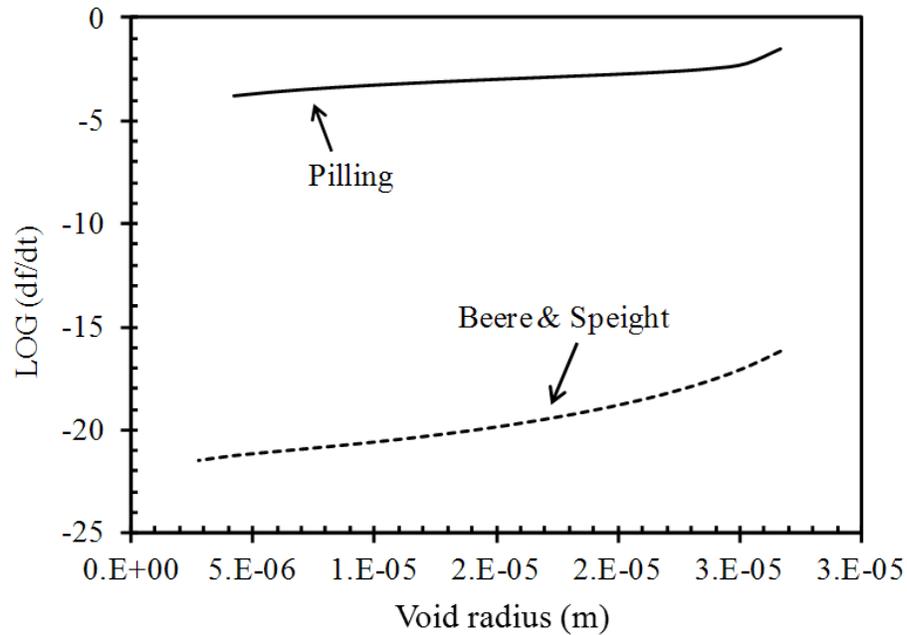
Figure 3.18- *Comparison of predictions of rate of change in fractional area of non-bonded interface, f_h, with void radius for the Beere and Speight and Pilling models.*

The time predictions of the Pilling model are in a better range for rolling applications compared to other models. Figure 3.18 compares the rate of change in fractional area of non-bonded interface predicted by the Beere and Speight model and the Pilling model. Generally models which emphasise more the diffusion bonding and creep than plastic deformation predict very long healing times. Since the void elimination model developed here is for the purpose of rolling applications, shorter healing time predictions are preferable. Furthermore, the Pilling model meets the void shape requirement (cylindrical shape of voids in cast steel). In addition, the Pilling model does not involve the complications of finding a non-real value such as diffusion distance, Λ, as in Beere and Speight model, and the critical loading parameter, Λ_c, as in the Wang model; this makes the implementation of the model simpler.

3.4 Conclusions

The existing void shrinkage models were reviewed and the model discrepancies in shrinkage rate predictions were explained. Factors such as void size, void geometry, the mechanisms involved and the type of operating mechanisms (parallel or series) were suggested to be the causes of the differences in predictions made by the different models.

A range of existing void shrinkage models was reviewed. Some of these models were developed from void growth models where void shrinkage was considered as negative void growth. Other void shrinkage models, on the other hand, considered the diffusion-bonding process to be analogous to that of pressure sintering and applied sintering based mechanisms for elimination of interfacial voids in metal joining and crack healing. In addition to these models, some others were also developed that were solely based on diffusion mechanisms.

From all the existing models, the Pilling model has been selected by the author to be implemented into the FE code and applied in the developed void elimination software for prediction of healing time for the purpose of simulating rolling applications. The reason for this selection has been explained in terms of shorter time predictions (suitable for rolling applications) made by this model, due to its accounting for plastic deformation compared to other models which emphasise on diffusion bonding and creep, and therefore predict very long healing times. The other advantage of the Pilling model is that it meets the void shape requirement (cylindrical shape of voids in cast steel). In addition, the Pilling model does not involve the complications of finding non-real value as in other models.

This model has been implemented into the void elimination software in Chapter 5 to predict healing time.

3.5 References

Artz, E., Ashby, M. F. & Easterling, K. E. 1983. Practical applications of hot-isostatic pressing diagrams: four case studies. *Metallurgical and Materials transactions A,* 14A**,** 211-221.

Ashby, M. F. 1974. A first report on sintering diagrams. *Acta metallurgica,* 22**,** 275-289.

Beere, W. & Speight, M. V. 1975. Vacancy potential and void growth on grain boundaries. *Metal Science,* 9**,** 190-191.

Beere, W. & Speight, M. V. 1978. Creep cavitation by vacancy diffusion in plastically deforming solid. *Metal Science,* 12**,** 172-176.

Chen, I. W. & Argon, A. S. 1981. Diffusive growth of grain-boundary cavities. *Acta Metallurgica,* 29**,** 1759-1768.

Chuang, T. J. & Rice, J. R. 1973. The shape of intergranular creep cracks growing by surface diffusion. *Acta Metallurgica,* 21**,** 1625-1628.

Cocks, A. C. F. & Ashby, M. F. 1982a. On creep fracture by void growth. *Progress in Materials Science,* 27**,** 189-244.

Cocks, A. C. F. & Ashby, M. F. 1982b. Creep fracture by coupled power-law creep and diffusion under multiaxial stress. *Metal Science,* 16**,** 465-474.

Derby, B. & Wallach, E. R. 1982. Theoretical model for diffusion bonding. *Metal Science,* 16**,** 49-56.

Derby, B. & Wallach, E. R. 1984. Diffusion bonds in copper. *Journal of Materials Science,* 19**,** 3140-3148.

Edward, G. H. & Ashby, M. F. 1978. Intergranular fracture during power-law creep *Acta Metallurgica,* 27**,** 1505-1518.

Guo, Z. X. & Ridely, N. 1987. Modelling of diffusion bonding of metals. *Materials Science and Technology,* 3**,** 945-953.

Hancock, J. W. 1976. Creep cavitation without a vacancy flux. *Metal Science,* 10**,** 319-325.

Hill, A. & Wallach, E. R. 1989. Modelling solid-state diffusion bonding. *Acta Metallurgica,* 37**,** 2425-2437.

Hull, D. & Rimmere, D. E. 1959. The growth of grain-boundary voids under stress. *Philosophical Magazine,* 4**,** 673-687.

Kassner, M. E. & Hayes, T. A. 2003. Creep cavitation in metals. *International Journal of Plasticity,* 19**,** 1715-1748.

Kuczynski, G. C. 1956. The mechanism of densification during sintering of metallic particles. *Acta Metallurgica,* 4**,** 58-61.

Lee, Y. S., Kozlosky, T. A. & Batt, T. J. 1993. Effects of grain boundary diffusion and power law creep on cylindrical cavity deformation. *Acta Metallurgica et Materialia,* 41**,** 1841-1854.

Needleman, A. & Rice, J. R. 1980. Plastic creep flow effects in the diffusive cavitation of grain boundaries. *Acta metallurgica,* 28**,** 1315–1332.

Pilling, J. 1988. The kinetics of isostatic diffusion bonding in superplastic materials. *Materials Science and Engineering,* 100**,** 137-144.

Pilling, J., Liversey, D. W., Hawkyard, J. B. & Ridely, N. 1984. Solid state bonding in superplastic Ti-6AI-4V. *Metal Science,* 18**,** 117-122.

Pilling, J. & Ridely, N. 1987. Solid state bonding of superplastic AA 7475. *Materials Science and Technology,* 3**,** 353-359.

Raj, R. & Ashby, M. F. 1975. Intergranular fracture at elevated temperature. *Acta Metallurgica,* 23**,** 653-666.

Riedel, H. 1987. Cavity nucleation at particles on sliding grain boundaries. A shear crack model for grain boundary sliding in creeping polycrystals. *Acta Metallurgica,* 32**,** 313-321.

Rimmer, D. 1959. The growth of grain-boundary voids under stress. *Philosophical Magazine.*

Speight, M. V. & Beere, W. 1975. Vacancy potential and void growth on grain boundaries. *Metal Science,* 9**,** 190-191.

Speight, M. V. & Harris, J. E. 1967. The kinetics of stress-induced growth of grain-boundary voids. *Metal Science,* 1**,** 83-85.

Sun, J. 2002. A model for shrinkage of a spherical void in the center of a grain: Influence of lattice diffusion. *Journal of Materials Engineering and Performance,* 11**,** 322-331.

Takahashi, Y. & Inoue, K. 1992. Recent void shrinkage models and their applicability to diffusion bonding. *Materials Science and Technology,* 8**,** 953-964.

Wang, H. & Li, Z. 2003. Diffusive shrinkage of a void within a grian of a stressed polycrystal. *Journal of the Mechanics and Physics of Solids,* 51**,** 961-976.

Wang, H. & Li, Z. 2004. The three-dimensional analysis for diffusive shrinkage of a grain-boundary void in stressed solid. *Journal of Materials Science,* 39**,** 3425-3432.

Weertman, J. 1973. Grain boundary void growth theory- a correction. *Scripta Metallurgica,* 7.

Wilkinson, D. S. & Ashby, M. F. 1975. Pressure sintering by power law creep. *Acta Metallurgica,* 23**,** 1277-1285.

Chapter 4

Automated calibration of void closure model parameters

The so-called Gurson model is a well-established micromechanical model of the ductile fracture of porous materials. The current work explores the applicability of the Gurson model to predict void closure. The fixed parameters characterising the modified Gurson model are not universal and must be calibrated for a particular material, typically by trial-and-error fitting of FE simulations to experimental data. However, the trial and error approach is expensive and time consuming (one test generally corresponds to only one triaxiality level). A novel approach has been developed in the present chapter to identify the void closure model parameters using Pattern Search method which is a well-known non-gradient based optimisation search method (Pattern Search Method).

4.1 Effect of triaxiality on void closure

Recent works have demonstrated that stress triaxiality is one of the key aspects affecting void coalescence in porous materials (Vadillo and Fernández-Sáez, 2009, Zhang and Chen, 2007). Finite element simulation of void changes under constant triaxiality is a standard numerical procedure in the evaluation of ductile damage of metals. A number of simulations have been performed for uniform loading, and constant triaxiality allowing the effect of triaxiality to be investigated separately (Brocks et al., 1995, Kuna and Sun, 1996, Pardoen and Hutchinson, 2000, Steglich and Brocks, 1997). This study investigates the effect of triaxiality on void closure by predicting the load under which a void with specific volume under certain triaxiality closes.

A pertinent application of the micromechanics of void closure is the thermomechanical processing of steel, e.g. long rolling, in which the rolling stages are ideally optimised to plastically close porosity left in the cast and diffusion bond under temperature and pressure the crack-like interfaces that remain. Complicating matters, the porosity distribution in as-cast steel is generally highly nonuniform; a billet contains more, larger pores at the centre than the surface, therefore the size and distribution cannot be predicted precisely. The voids in different positions experience varying boundary conditions and will close under different loading conditions.

4.1.1 Methodology

To investigate the closure of a void at a specific position in a steel billet, the stress state at that point can be identified and triaxiality obtained by FE modelling of the rolling process. The conditions for plastic closure of a void in that position could therefore be predicted by determining a closure load versus triaxiality relationship obtained from a representative volume element (RVE) model such as that employed in this study. An axisymmetric model of a cylinder of material with a central

spherical hole was constructed using ABAQUS/Explicit (2010), Version 6.10-2, a schematic of which is shown in Figure 4.1. In order to systematically study the dependence of void closure on stress triaxiality, a constant triaxiality (T) is maintained during loading, which dictates a dependency between the boundary stress components. The axial stress σ_y was applied linearly, with lateral stress, σ_x, defined as a function of the axial stress and the triaxiality. At the mesoscopic scale, i.e. on a scale that is large relative to the pore/void, but small relative to that of the billet, the hydrostatic and Von Mises stresses, σ_h and σ_{eq} and the triaxiality T are (the stress is nonuniform on the scale of the void):

$$\sigma_h = \frac{1}{3}\left(\sigma_x + \sigma_y + \sigma_z\right) = \frac{1}{3}\left(2\sigma_x + \sigma_y\right) \tag{4.1}$$

$$\sigma_{eq} = \sqrt{\frac{1}{2}\left(\left(\sigma_x - \sigma_y\right)^2 + \left(\sigma_y - \sigma_z\right)^2 + (\sigma_z - \sigma_x)^2\right)} = |\sigma_y - \sigma_x| \tag{4.2}$$

$$T = \frac{\sigma_h}{\sigma_e} = \frac{2\sigma_x + \sigma_y}{3|\sigma_y - \sigma_x|} = -\frac{2P_x + P_y}{3|P_x - P_y|} \tag{4.3}$$

where the pressures (positive pressure acts in the direction opposite the normal) are $P_x = -\sigma_x$ and $P_y = -\sigma_y$. Only positive values of P_y are considered, i.e. compressive axial stress. Assuming that $P_y > |P_x|$, two loading boundary conditions can be considered: Condition (A) $P_x > 0$; Condition (B) $P_x < 0$, as depicted in Figure 4.1, both of which reduce equation 4.3 to:

$$T = -\frac{2P_x + P_y}{3\left(P_y - P_x\right)} \tag{4.4}$$

The range of triaxialities must be identified that satisfies either Condition A or B, hence leading to compression and closure of the void. Under Condition A, the numerator and denominator in equation 4.4 are both positive, hence triaxiality is negative. However, under Condition B, the denominator could be negative (triaxiality positive) or positive (triaxiality negative) depending on the applied

pressures: when $P_y > -2P_x$, the triaxiality is negative, and when $P_y < -2P_x$, the triaxiality is positive. To satisfy the condition of $P_y > |P_x|$ for positive triaxialities, an upper limit on the triaxility must be determined by substituting $P_y = -P_x$ into equation 4.4, which gives $T = 1/6$. Table 4.1 summarises the process of defining the allowable triaxiality range.
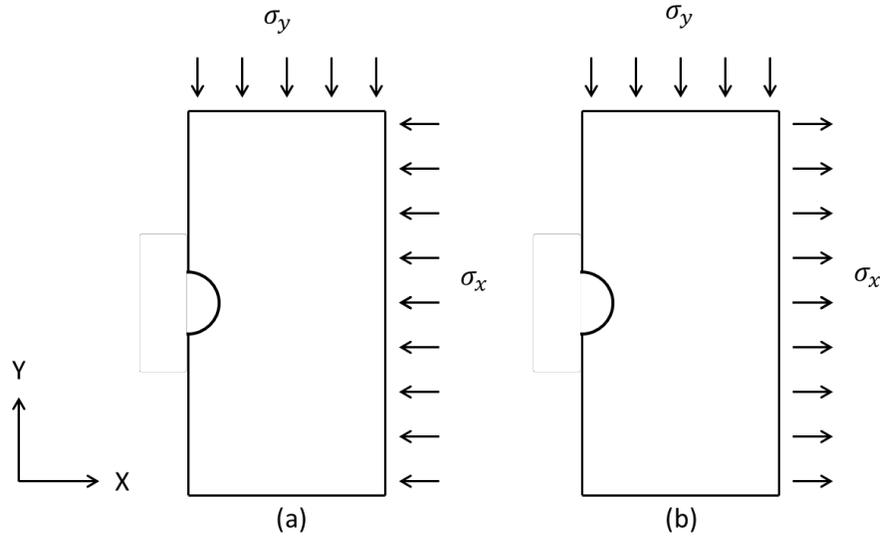
Figure 4.1- *Possible loading boundary conditions*.

Table 4.1- *Process of defining the allowable triaxiality range*.

	$P_y = -2P_x$		$P_x = 0$	
P_x	$-$	$-$		$+$
P_y	$+$	$+$		$+$
$2P_x+P_y$	$-$	$+$		$+$
$3(P_y\text{-}P_x)$	$+$	$+$		$+$
T (triaxiality)	$+$	$-$		$-$

Condition A   Condition B

Figure 4.2 shows the triaxiality range of $-\infty < T < \frac{1}{6}$ which has been obtained and will be used for the simulations.
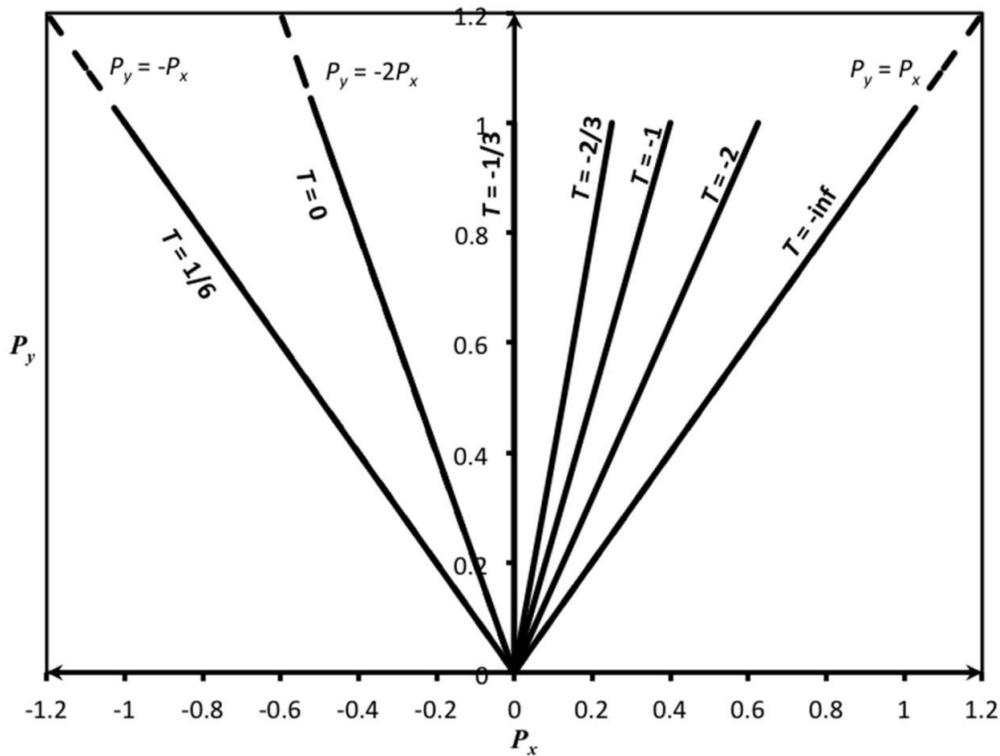
Figure 4.2- *Possible triaxialities leading to compression under the assumption of $P_y > |P_x|$.*

4.1.2 The representative volume element (RVE) model

As explained in the previous section, an axisymmetric model of a cylinder of material with a central spherical hole was constructed using ABAQUS/Standard. A detailed schematic of this is shown in Figure 4.3, where H and W are the height and radius of the cylinder, respectively, and the radius of the void, ρ, can be expressed in terms of the void volume fraction and geometry as:

$$\rho = \left(\frac{3 \cdot vvf \cdot HW^2}{4}\right)^{1/3} \qquad\qquad 4.5$$

The effects of strain rate and temperature on void closure are investigated for an RVE with equal height to diameter ($H = 5.0$ and $W = 2.5$) and an initial void volume fraction of $f_0 = 0.002$.

The material properties used in the model were the experimental data obtained from high-temperature tensile testing of free cutting steel at 900°C and 1100°C (Foster, 2007) (see Appendix A), which are temperatures typical of hot rolling, with the elastic-plastic properties input to the simulations as tabulated stress versus strain data for a given strain rate and temperature (i.e. a given simulation does not include material rate dependence, although the effect of strain rate is studied by multiple simulations).

To account for large deformations, the finite strain/deformation theory was used in the simulation (NLGEOM was set to ON in ABAQUS).

Instead of modelling only half of the cell, the entire cell was modelled to allow for the void closure process, as the top surface of the void touches the lower surface. The detailed description of the model is provided in the following subsections.
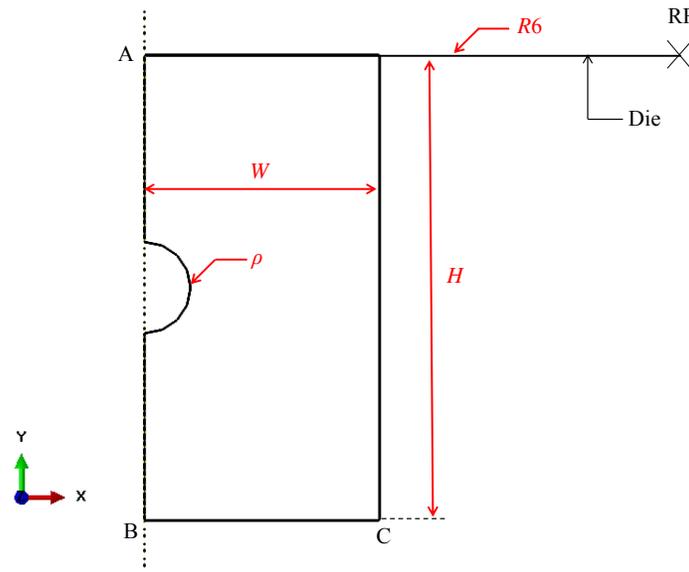
Figure 4.3- *The RVE model*.

Boundary conditions:

In Figure 4.3, line AB is the rotational axis of symmetry to generate axisymmetric boundary conditions ($U_x = U_z = UR_y = UR_z = 0$, where U and UR represent

displacement and rotation, respectively). The bottom of the cylinder (line BC) is fixed in the *y* direction ($U_y = 0$).

The die in Figure 4.3 is an axisymmetric discrete rigid body, which is in contact with the top of the cylinder and is only free to move in the *y* direction ($U_x = U_z = UR_y = UR_z = 0$). This boundary condition was applied to the reference point (RP) of the rigid body.

Loading conditions:

Computations were carried out at a prescribed fixed triaxiality. Several methods can be used to impose constant stress triaxiality on the cell. In the present work the pressure P_y was increased linearly with P_x depending on P_y and T according to equation 4.4 until the point of incipient contact between opposing faces of the void, at which point P_y and P_x were recorded as the axial and lateral pressures, respectively, required for void closure under triaxiality $T = \sigma_h / \sigma_e$.

A frictionless surface to surface type of contact was defined between the die and the cylinder. 'Hard' type of contact (default assumption in surface to surface contact in ABAQUS) was used in defining the interaction in the normal direction. Thermal interactions between the die and the cylinder were ignored in this simulation.

Mesh and element type sensitivity:

To select an appropriate mesh size for the model, a mesh sensitivity analysis was performed where four different mesh densities were inspected. Initially a coarse mesh was applied to the model producing 90 elements. Finer meshes resulting in 250, 1000 and 2250 elements were also examined. These meshes are shown in Figure 4.4. The variation of void volume fraction, *vvf* with time during compression was obtained for these meshes, as shown in Figure 4.5. This figure shows a good convergence of the results for a mesh density of 1000 elements corresponding to the mesh shown in Figure 4.4c. This mesh was therefore selected to be applied to the model.

For axisymmetric conditions, two options are available for the element type in ABAQUS. These include CAX4, which is a 4-node bilinear axisymmetric quadrilateral element and CAX8 which is a 8-node biquadratic axisymmetric quadrilateral element (see the ABAQUS theory manual for a complete description of the elements). To select the suitable element type for the model, the variation of void volume fraction, *vvf*, (see equation 4.6 in section 4.1.3) with time during compression was obtained for these two element types. The results of this analysis are shown in Figure 4.6, where no significant differences are observed when using CAX4 and CAX8 elements. However, CAX4 element type is preferred over CAX8 due to faster computations.

In order to be computationally efficient, reduced integration elements (CAX4R and CAX8R) are a good option. However, before using these elements the effect of using these elements on the accuracy of the results has to be checked. The variation of void volume fraction, *vvf*, with time during compression was obtained for CAX4R elements. The result of this is also shown in Figure 4.6, where no significant difference can be observed in the obtained results for CAX4R and CAX4.

Based on the above findings, CAX4R element type was selected for the model.

| (a) | (b) | (c) | (d) |
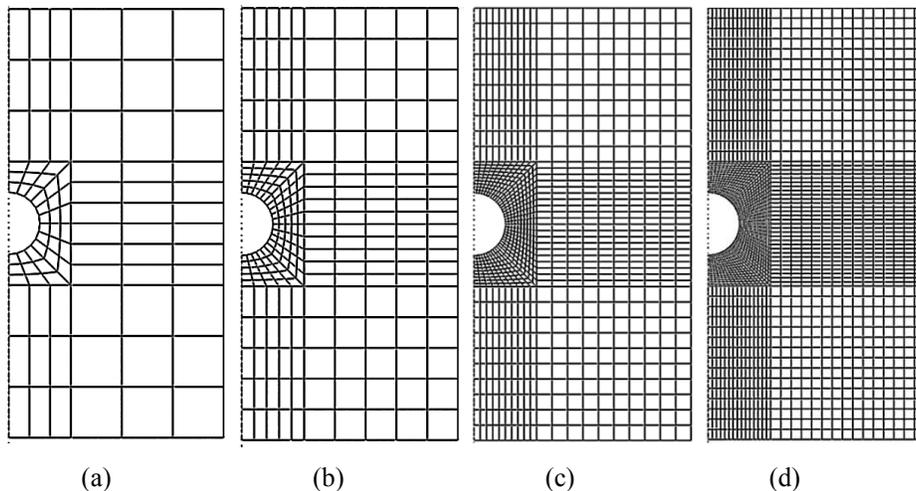
Figure 4.4*- Finite element mesh for the voided cell model with total number of elements equal to (a) 90, (b) 250, (c) 1000, (d) 2250.*
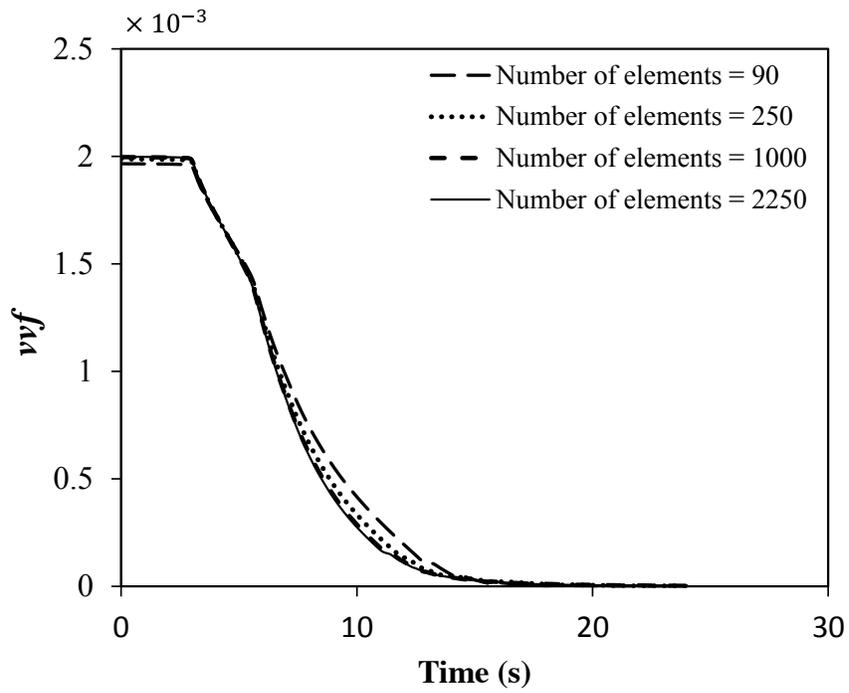
Figure 4.5*- The effect of mesh size on the prediction of vvf versus time for the RVE model.*
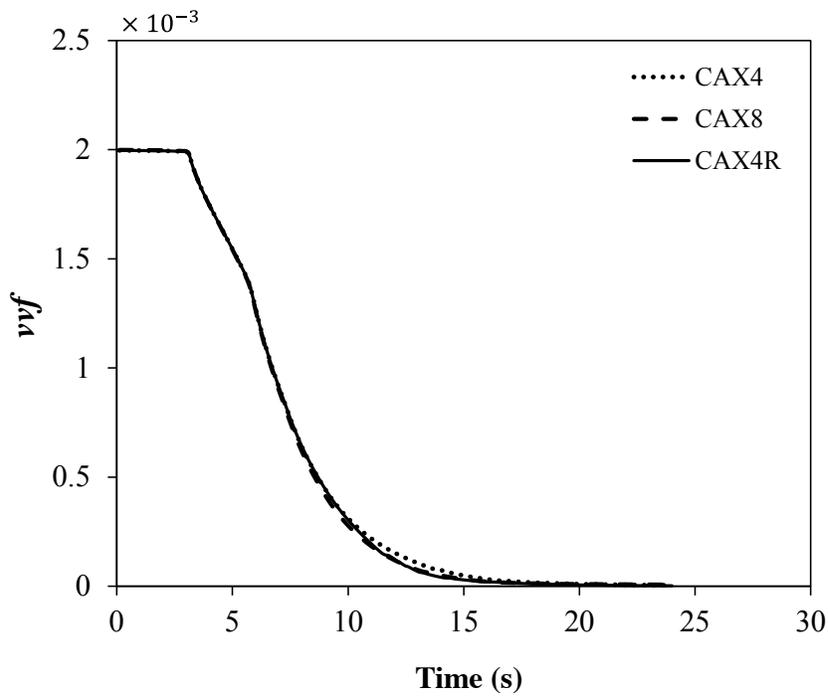
Figure 4.6*- The effect of element type on the prediction of vvf versus time for the RVE model.*

4.1.3 Calculation of void volume fraction, *vvf*

This section explains how void volume fraction is calculated for the RVE model.

For the voided unit cell, the initial void volume fraction can be written as:

$$vvf = \frac{\frac{4}{3}\pi\rho^3}{\pi W^2 H} \tag{4.6}$$

where ρ is the void radius and W and H are the radius and height of the cylindrical cell, respectively as defined in Figure 4.3.

To obtain the volume of the void at any time increment during deformation, the *x-y* coordinates of the void surface are extracted using Python coding. Python is a powerful, interactive, object-oriented programming language. ABAQUS makes extensive use of Python for accessing ABAQUS results on the output database (.odb) file, scripting with ABAQUS/CAE and ABAQUS/Viewer, and generally to accomplish varied programming tasks. Here for simplification the coordinates are extracted for a quarter of the model only, as shown in Figure 4.7.
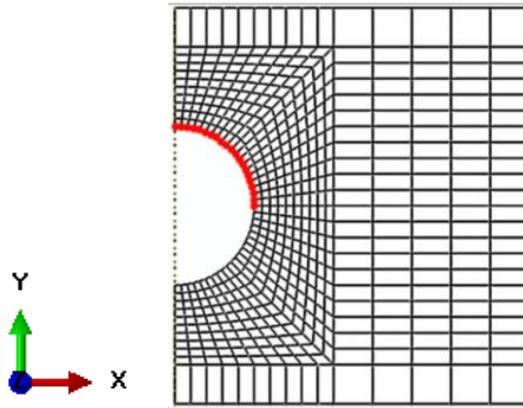
Figure 4.7*- Selected nodal points for calculation of void volume.*

Rotation of the *x-y* profile of the hole about the *y*-axis produces a half sphere (see Figure 4.8), the volume of which can be calculated as an integration of frustums of cones along the *y*-axis.
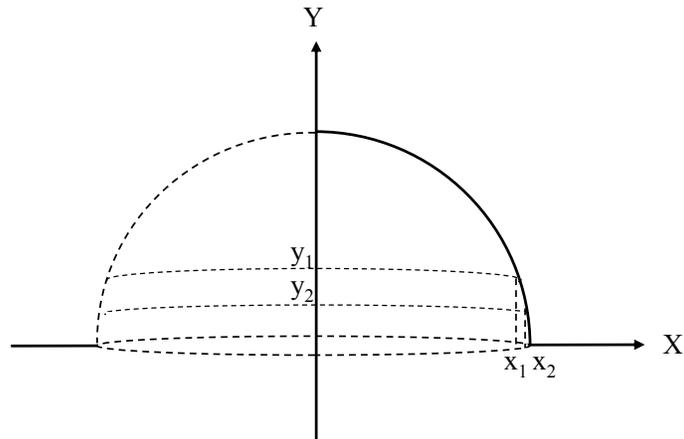
Figure 4.8*- Void volume calculation as an integration of frustums of cones along the*

y-axis.

$$volume\ of\ half\ sphere = \sum_{i=0}^{i=N} \frac{\pi}{3}(x_i^2 + x_iy_{i-1} + x_{i-1}^2)\text{abs}(y_i - y_{i-1}) \qquad 4.7$$

The integration approach used to calculate the void volume can also be used to obtain the volume of the cell. The *x-y* coordinates of the edge (see Figure 4.9) of the cell are extracted during the deformation and the volume of the cell is calculated.
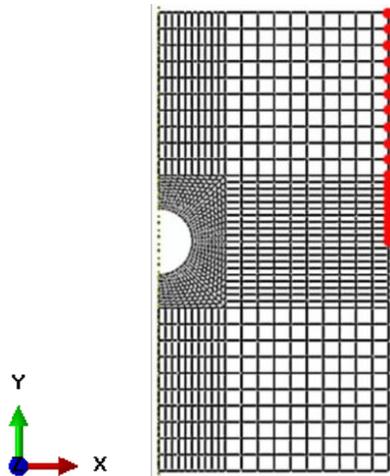
Figure 4.9*- Selected nodal points for calculation of the volume of the cylinder*.

Figure 4.10 illustrates a schematic of the voided cell during the compression, where the *x-y* coordinates of the void surface (profile of the void) and *x-y* coordinates of the edge of the cell (profile of the cell) are shown at three stages of the deformation process.
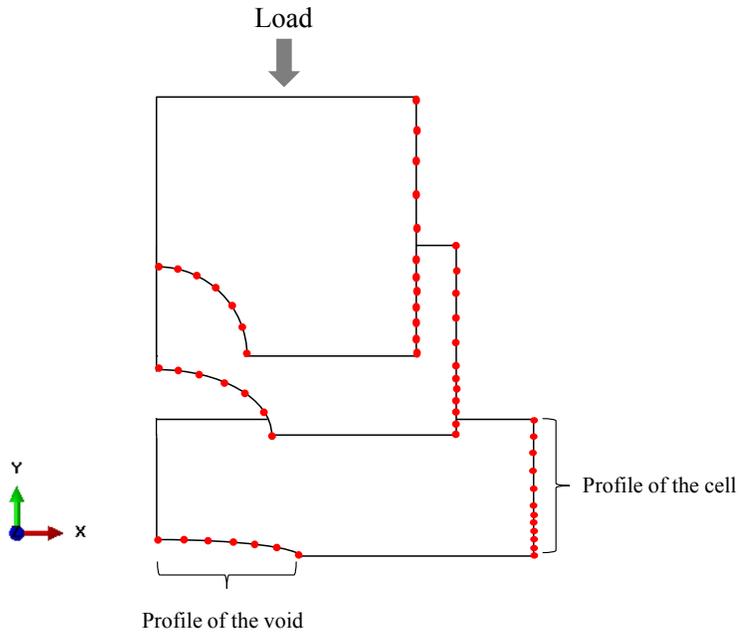
Figure 4.10*- The x-y coordinates of the selected nodes for the void (represented as the profile of the void) and for the cell (represented as the profile of the cell) at three different stages during the deformation process*.

MATLAB 2011a was used to calculate the volume of the void, volume of the cell and therefore the void volume fraction at any time during the deformation. A user interface was created which requires the user to upload the *x-y* profiles of the void and the cell and calculates the *vvf* at any time during the deformation. This user interface is shown in Figure 4.11. As mentioned previously the *x-y* profiles of the void and the cell are extracted from the model in the format of ABAQUS report, using Python coding. The developed Python and MATLAB codes for calculation of *vvf* are provided in Appendix B.
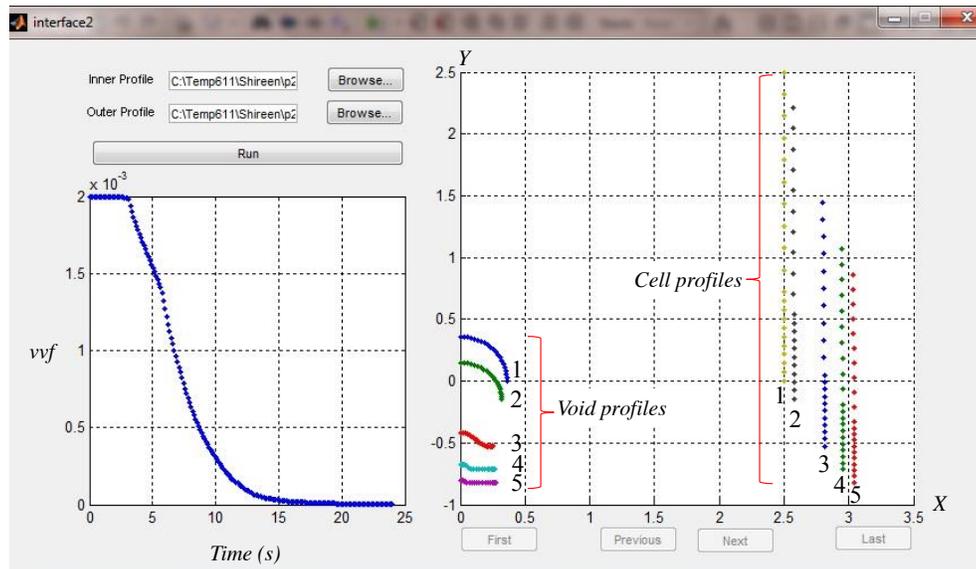
Figure 4.11 - *The MATLAB user interface for calculation of vvf. Void and cell profiles are shown at five stages during deformation on the right and vvf versus time is shown on the left.*

4.1.4 Satisfaction of incompressibility

The aim of this section is to study the satisfaction of incompressibility in the RVE model during plastic deformation.

Figure 4.12 shows the variation in the volume of the matrix, V_m, normalised by the volume of the matrix at the start of plastic deformation, V_{mo}, with time during compression for three different initial void volume fractions, f_o. The total volume of the matrix is obtained by summing the volumes of all elements (EVOL in ABAQUS). Figure 4.12 shows 0.46% reduction in the total volume of the matrix over the deformation for all cases of f_o. To study the effect of this volume reduction on the *vvf* calculation, a comparison should be made between the *vvf* versus time obtained based on the assumption of constant V_m, or incompressible matrix, and *vvf* versus time when there is 0.46% reduction in the volume of the matrix. The volume of the cell is defined as:

$$cell\ volume = \ V_m + V_v$$

where V_m and V_v are volume of the matrix and the void, respectively.

78

The void volume can be obtained using the integration method explained in section 4.1.3. The change in *vvf* with time obtained based on this assumption is represented as case 1 in Figure 4.13.

For an incompressible matrix (case 1), V_m is constant during the deformation and is obtained by adding the volume of all elements at the start of plastic deformation. As previously shown in Figure 4.12, the matrix volume is varying slightly. The volume of the matrix, V_m can either be obtained from the integration method, case 2, explained in section 4.1.3 or by adding the volume of the elements (EVOL in ABAQUS), case 3. The *vvf* versus time obtained for both cases is shown in Figure 4.13. It should be noted that for all cases the volume of the void is calculated from the integration method. As it is evident from Figure 4.13, there is no difference in the *vvf* versus time prediction for constant V_m (case 1) and slightly varying V_m (case 2 and case 3). Therefore, the slight variation in matrix volume shown in Figure 4.12 does not have an influence on the *vvf* versus time analysis.
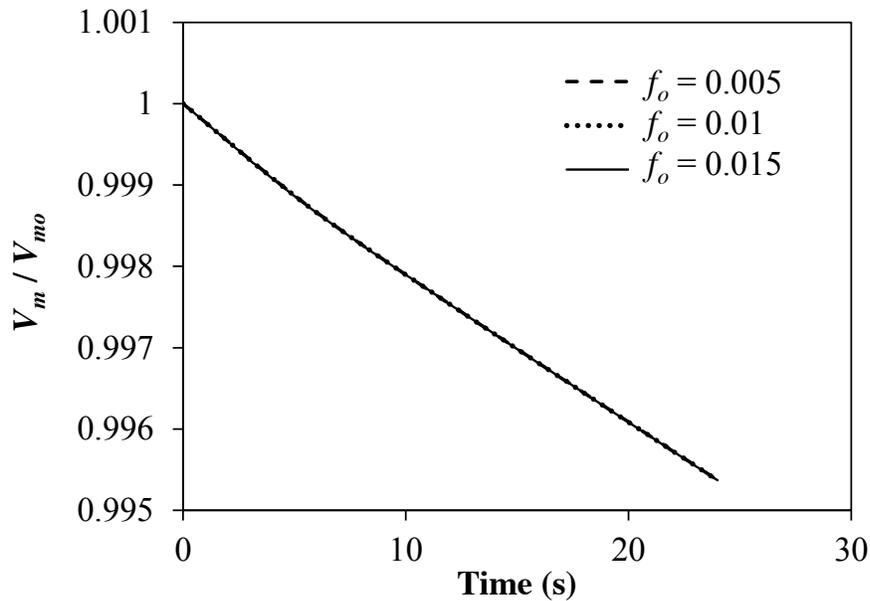
Figure 4.12 *- Variation of the matrix volume, V_m, normalised by the matrix volume at the start of plastic deformation, V_{mo}, with time.*
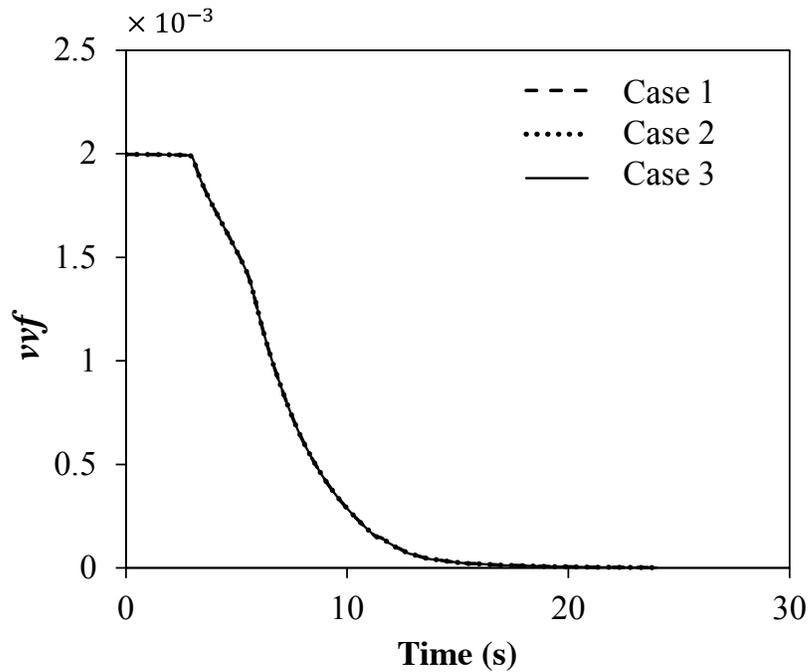
Figure 4.13 *- Comparison of vvf versus time for case 1, case 2 and case 3. In case 1 matrix volume is assumed constant whereas in case 2 and case 3 the matrix volume varies by 0.46%. In case 2, matrix volume is calculated from the integration method explained in section 4.1.3 and in case 3 matrix volume is obtained by adding the volume of the elements (EVOL in ABAQUS).*

4.1.5 Results

Figure 4.14 and Figure 4.15 illustrate the influence of temperature on the axial and lateral pressures required for void closure under different triaxialities for a strain rate of $\dot{\varepsilon} = 0.1$. The closure pressures for temperatures 900°C, 1000°C and 1100°C have been obtained from FE simulations. Figure 4.14 shows that an increasingly negative triaxiality leads to larger required closure pressures, an effect resulting from the plastic incompressibility of the material surrounding the void. Although not evident in these figures for the triaxialities shown, the closure pressures also approach one another as Triaxiality, T tends to -∞, hence the void tends to an increasingly equiaxed shape throughout the closure process as T becomes very small (negative). The results

in Figure 4.14 and Figure 4.15 also quantify the degree to which the higher temperature, 1100°C, reduces the pressures relative to 1000°C (e.g. P_y and P_x each reduce by roughly 32% for a triaxiality T = -2) and 900°C (e.g. P_y and P_x each reduce by roughly 55% for a triaxiality T = -2) required to close the void, with an increasing effect for smaller triaxialities.

The effect of strain rate on the closure pressures for the same initial void volume fraction ($f_0 = 0.002$) and a temperature of 900°C is studied here. The mean strain rate experienced by a work-piece during rolling varies with the roll rotation speed, separation distance and radius. Initially the material has a lower velocity through the rolls, resulting in lower strain rates but the velocity increases as the work-piece is rolled leading to higher strain rates. Decreasing the separation distance of the rolls and increasing the roll's radius increases the strain rate. Applying mean strain rate equations such as the one derived by Sims (1954) for the initial rolling/roughing stage provides a strain rate envelope with minimum and maximum values of 0.1 and 10 s^{-1} respectively. To cover this range of strain rate, three orders of magnitude, 0.1, 1 and 10 for the strain rate have been considered here.

Increasing the strain rate from $\dot{\varepsilon} = 0.1$ to $\dot{\varepsilon} = 10$ significantly increases the pressures required to close the void (e.g. P_y and P_x each increase by approximately 33% for a triaxiality T = -2).
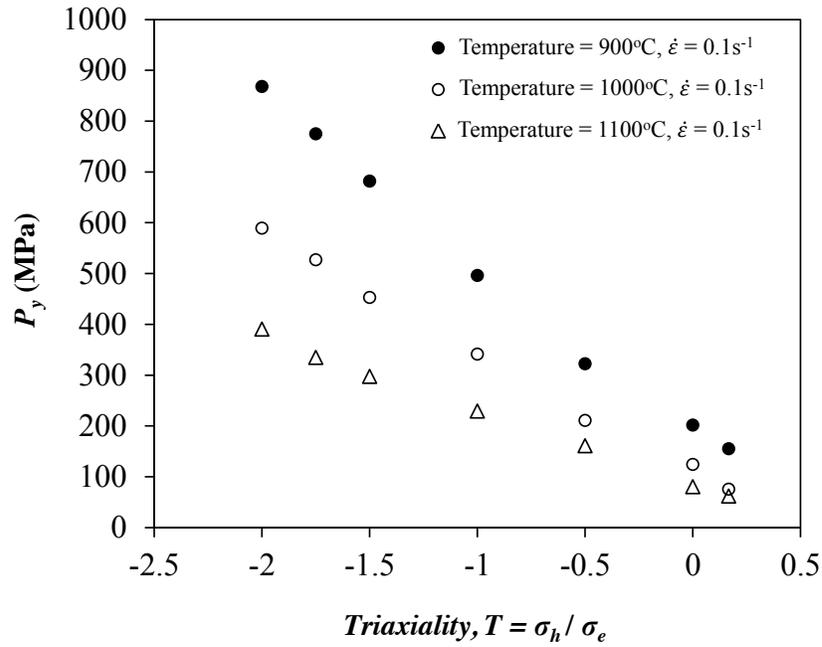
Figure 4.14- *Axial pressures at the onset of void closure versus triaxiality, showing the effect of temperature.*
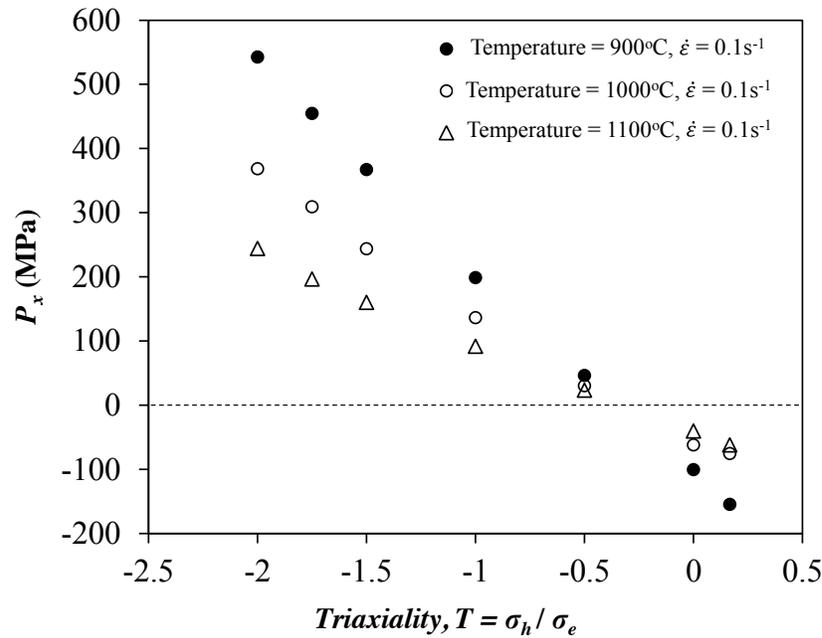
Figure 4.15- *Lateral pressures at the onset of void closure versus triaxiality, showing the effect of temperature.*
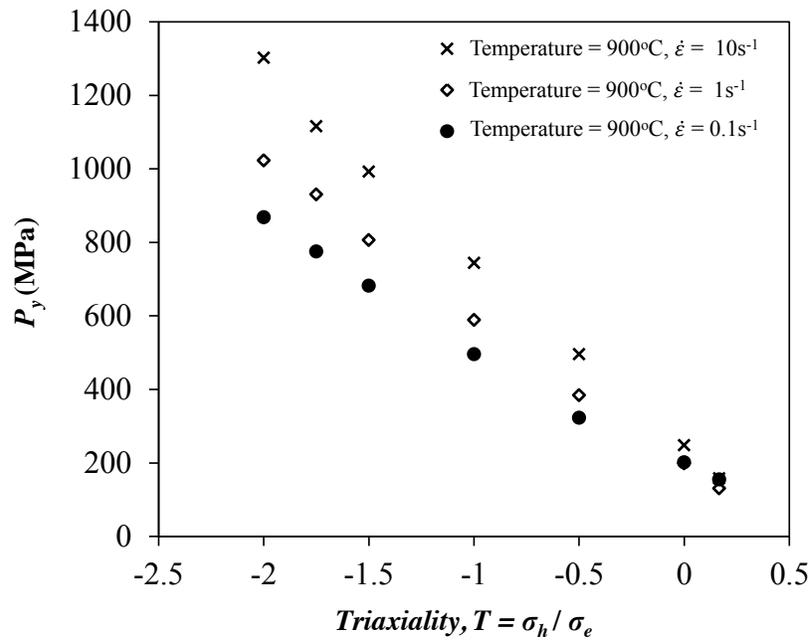
Figure 4.16- *Axial pressures at the onset of void closure versus triaxiality, showing the effect of strain rate.*
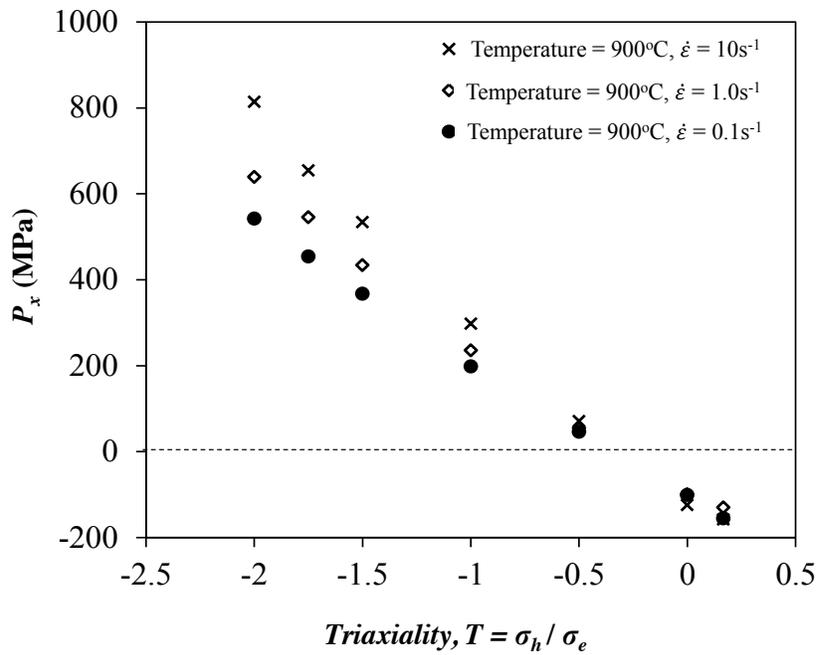
Figure 4.17- *Lateral pressures at the onset of void closure versus triaxiality, showing the effect of strain rate.*

4.2 Calibration of GT model parameters for void closure

Gurson model is a widely known and used micromechanical model for the ductile fracture of porous materials. The yield surface of the Gurson-Tvergaard model, including the fitting parameters, q_1, q_2 and q_3, is given in terms of the equivalent and hydrostatic stress measures by:

$$\Phi(q, \bar{\sigma}, f^*, \sigma_h) = \frac{\sigma_{eq}^2}{\bar{\sigma}^2} + 2q_1 f^* \cosh\left(\frac{3q_2\sigma_h}{2\bar{\sigma}}\right) - 1 - q_3 f^{*2} = 0 \qquad 4.8$$

where f^* is a damage parameter that depends on the void volume fraction f, which is an average measure for a void-matrix aggregate, σ_h is the mean normal stress, σ_{eq} is the Von Mises stress and $\bar{\sigma}$ is the flow stress of the matrix material. The model has been explained in more details in Chapter 2.

4.2.1 Fitting parameters q_1, q_2 and q_3

The parameters q_1 and q_2 were introduced by Tvergaard (1981, 1982) in what is now commonly known as the Gurson-Tvergaard (GT) model in order to account for the effect of plastic work hardening and to improve the model predictions relative to finite element analysis of void growth. Tvergaard defined the set of parameters as q_1 = 1.5, q_2 = 1.0 and $q_3 = q_1^2$ for a range of materials defined by $\sigma_Y/E = 0.004, \upsilon = 0.3$ and n = 5, 10, 20 and $\sigma_Y/E = 0.002, 0.008\ \upsilon = 0.3$ and n = 10 (where σ_Y is the yield stress, E is the Young's modulus, υ is the Poisson's ratio and n is hardening exponent). Søvik (1996) later suggested that the q_1 parameter can be considered to be a function of hardening exponent, and found that q_1 deviates from 1 when a material's hardenability decreases. Søvik (1996) explained this behaviour to be a result of considerable void shape change in materials with low hardening. Therefore, in experimental applications a fixed pair of (q_1, q_2) is often taken to ignore the effect

of strain hardening on void growth. Studies by Koplik and Needleman (1988) found that values of q_1 = 1.25, q_2 = 1.0 and $q_3 = q_1^2$ provide the best agreement between the GT model and the finite element results for power law hardening materials defined by $\sigma_Y/E = 0.002, v = 0.3$ and n values of 0.2, 0.1 and 0. Other q parameters can be found in the literature for different materials, Kuna and Sun (1996) found q_1 = 1.15 and q_2 = 1.0 and $q_3 = q_1^2$ for an elastic perfectly plastic material defined by $\sigma_Y/E = 0.0018, v = 0.3$. Mahnken (1999) found q_1 = 1.5 and q_2 = 0.5 and q_3 = 1.0 for a material defined by $\sigma_Y/E = 0.0017$ and $v = 0.3$.

Therefore a fixed set of q parameters cannot be applied to all materials. The q parameters vary for different material properties and also other parameters such as triaxiality. Kim et al. (2004) outlined the dependency of the q_1 and q_2 parameters on stress triaxiality and the initial void volume fraction, by studying the influence of stress triaxiality and initial void volume fraction on void growth and coalescence.

Although the Gurson model is widely used to model ductile fracture, its application to void closure has not received any attention. The aim of this study is to investigate the applicability of a modified Gurson model to predict void closure under compressive stress states. The diffusion processes explained in Chapter 2 and 3 are not considered in this chapter but will be studied in the following chapters.

The parameter q_1 affects the yield region by changing the dependency of the damage parameter on the void volume fraction f. High values of q_1 reduce the plastic limit and lower the strength of the porous material. Tvergaard originally assumed q_1 = 1.5 as an optimal value to produce good agreement between numerical and experimental results on porous solids. According to equation 4.8, q_2 influences the hydrostatic component of the current elastic domain. Increasing the value of q_2 significantly decreases the yield limit. Corigliano et al. (2000) showed that for high values of q_2 the strain hardening properties of the matrix material are almost completely eliminated by softening due to enhanced void growth, which is induced by the reduction of the overall strength of the porous material.

In this section finite element analyses are performed for a representative volume element (RVE) containing a spherical void and an equivalent cell of Gurson-Tvergaard (GT) material subject to a range of stress triaxialities. The free cutting steel (FCS) experimental material properties for 900 °C and $\dot{\varepsilon} = 0.1$ were used for both models (Foster, 2007) (see Appendix A for material data). The numerical results of the voided RVE and the corresponding GT cell model are compared, and an optimization method is used to find the GT model parameters that best fit the results.

4.2.2 The GT cell model

Finite element analyses were performed on a representative volume element (RVE) with a discrete spherical void at the centre, voided cell model (VC model) (Figure 4.4), and an equivalent cell of Gurson-Tvergaard (GT) material (Figure 4.18). Both models were axisymmetric cylinders with the same dimensions and equal height to diameter.

The void volume in the discrete void model was the same as the total volume of the pores distributed in the GT cell model. The matrix material was assumed to be strain hardening and isotropic, with free cutting steel (FCTS) experimental material properties used for 900 °C and $\dot{\varepsilon} = 0.1$ (Foster, 2007). To compare the behaviour of cell models, the same load, contact, geometric dimensions (radius and height of the cell) and boundary conditions used for the voided cell (VC) model as explained in section 4.1.2 were applied here. The method presented in Section 4.1.1 has been used to keep the triaxiality constant during loading. Figure 4.18 shows a schematic of the GT cell.
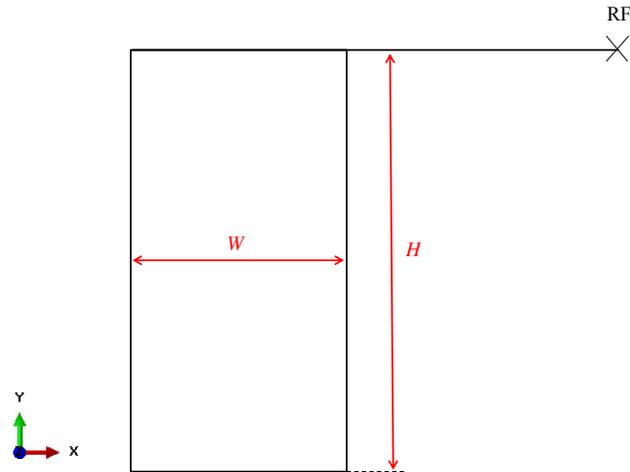
Figure 4.18- The equivalent Gurson-Tvergaard (GT) cell model.

Mesh and element type sensitivity analyses are performed for the GT cell to find a suitable mesh density and element type. Coarse, medium and fine meshes (corresponding to in 510, 1020 and 2040 elements, respectively) were examined. These meshes are shown in Figure 4.19. The variation of void volume fraction, *vvf* with time during compression was obtained for these meshes as shown in Figure 4.20. This figure shows a good convergence of the results for all considered mesh densities. The medium mesh however was applied to the GT model.

(a)   (b)   (c)
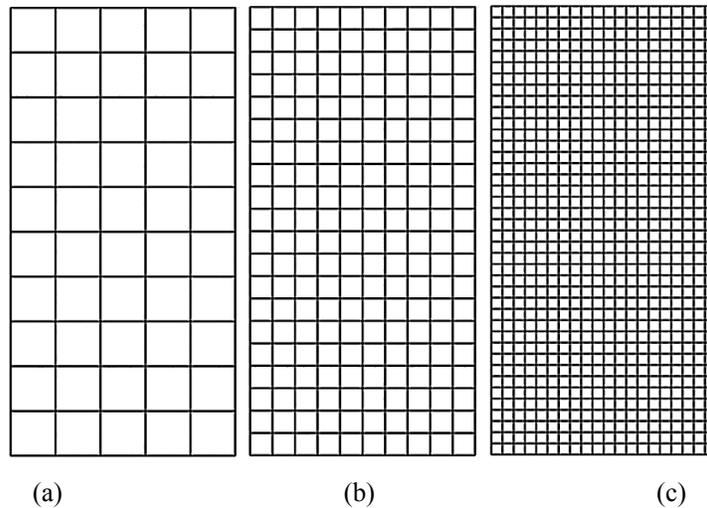
Figure 4.19- Finite element mesh for the GT model (a) Coarse mesh size (510 elements), (b) Medium mesh size (1020 elements) and (c) Fine mesh size (2040 elements).
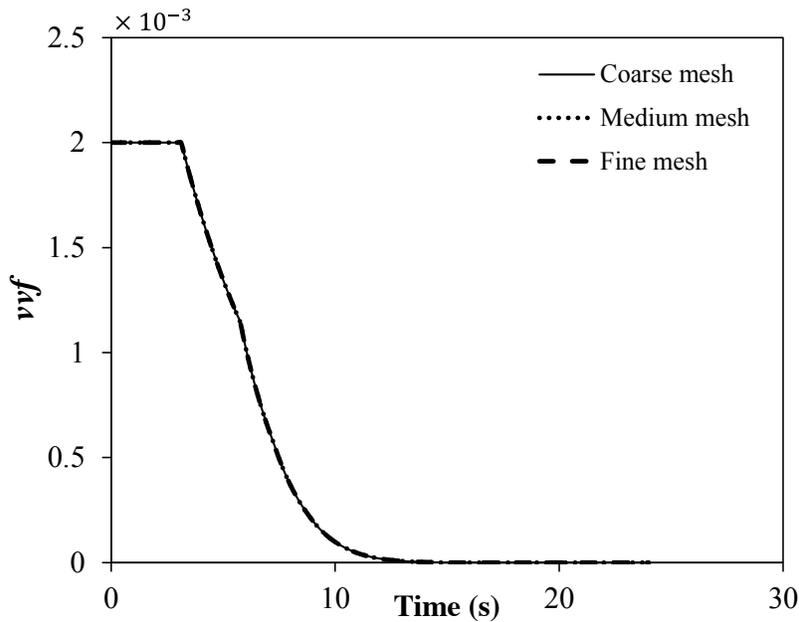
Figure 4.20- *The effect of mesh size on the prediction of vvf versus time for the GT model. Coarse, medium and fine meshes correspond to 510, 1020 and 2040 elements as shown in Figure 4.19 (a), (b) and (c), respectively.*

Similar to section 4.1.2, CAX4 (4-node bilinear axisymmetric quadrilateral element) and CAX8 (8-node biquadratic axisymmetric quadrilateral element) and also CAX4R (reduced integration CAX4 elements) were examined. To select the suitable element type for the model, the variation of void volume fraction, *vvf* with time during compression was obtained for these element types. The results of element type sensitivity are shown in Figure 4.21, where no significant difference is observed in the obtained results for CAX4, CAX4R and CAX8 elements. Similar to the voided cell model in section 4.1.2, CAX4R element was adopted for the FE model developed herein.
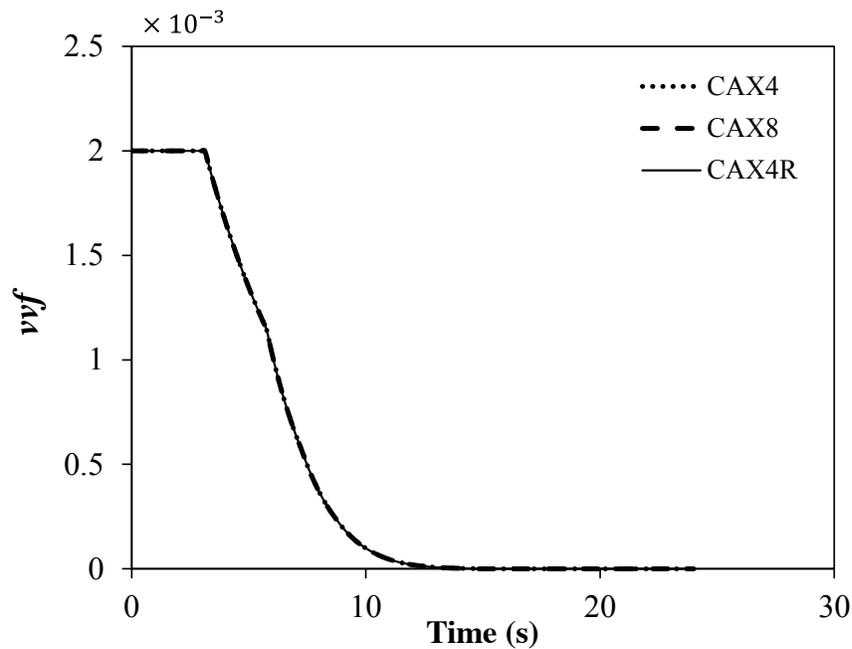
Figure 4.21- *The effect of element type on the prediction of vvf versus time for the GT model.*

4.2.3 The calibration process

A program was developed which optimises the *q* parameters, finding the values that minimise the difference in a least squares sense between the void volume fraction predicted by the GT FE model and that of the corresponding discrete void RVE FE model.

In order to create a model in ABAQUS, the common option is to use ABAQUS/CAE which is the pre-processor for ABAQUS. However, using CAE, any model creation or amendment has to be done manually. For parametric studies when the effect of a parameter change in the model is investigated, the manual amendments will be time consuming and sometimes impossible. To avoid the need for manual changes and to automate the process, Python was used herein. Python is a powerful, interactive, object-oriented programming language. ABAQUS makes extensive use of Python for scripting with ABAQUS/CAE, accessing ABAQUS results on the output database

(.odb) file, and ABAQUS/Viewer, and generally to accomplish varied programming tasks.

A simple MATLAB graphical user interface (GUI) was created to control the calibration process (see Figure 4.22). The developed MATLAB code for the GUI is provided in Appendix B part vi. The user is required to input values for initial void volume fraction, f_o, and triaxiality and set up a directory path, where all codes are stored. The process starts as the Run button is clicked.
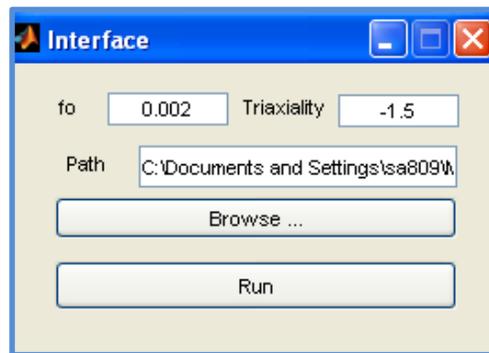
Figure 4.22- *The developed MALAB graphical user interface, GUI.*

Figure 4.23 contains a flow chart that illustrates the steps required to calibrate the *q* parameters. The information such as geometry, boundary conditions, loading and mesh are required to create the GT and VC models (see section 4.1.2 and 4.2.2). This information is hard coded in two Python pre-processor codes that are read by ABAQUS to create the GT and VC models (the developed codes are provided in Appendix B part i and ii). The values for the initial void volume fraction and triaxiality input by the user into the GUI are stored in a text file. The Python pre-processor accesses this text file (this information is also needed for the creation of the GT and VC cell models). The GT and VC models are solved in ABAQUS and a set of separate Python postprocessor codes is used to access the results. The first Python postprocessor code reads the *vvf* from the database of the GT cell model (the developed code is provided in Appendix B part v). The other two codes read the *x-y*

coordinates of the void and the cell from the database of the VC model (the developed codes are provided in Appendix B part iii and iv).

In the next step the difference between the void volume fraction predicted by the GT FE model and that of the corresponding discrete void RVE FE model is calculated in MATLAB. The Pattern Search optimisation method, which is a built-in optimisation function in MATLAB, is then used to optimise the q parameters. The current values of the q parameters, the maximum number of iterations, the optimisation tolerances and the current error (cost function defined in section 4.2.5) are input to the optimisation algorithm (see Appendix B part vi for further details). The optimisation algorithm determines a new set of q parameters and the process continues until the optimisation tolerances are achieved. The tolerances usually used as the termination criteria for the optimisation process have been taken as the default parameters suggested in MATLAB documentation (2011) and are summarised in Appendix C part i. The details of the optimisation procedure are explained in section 4.2.5.
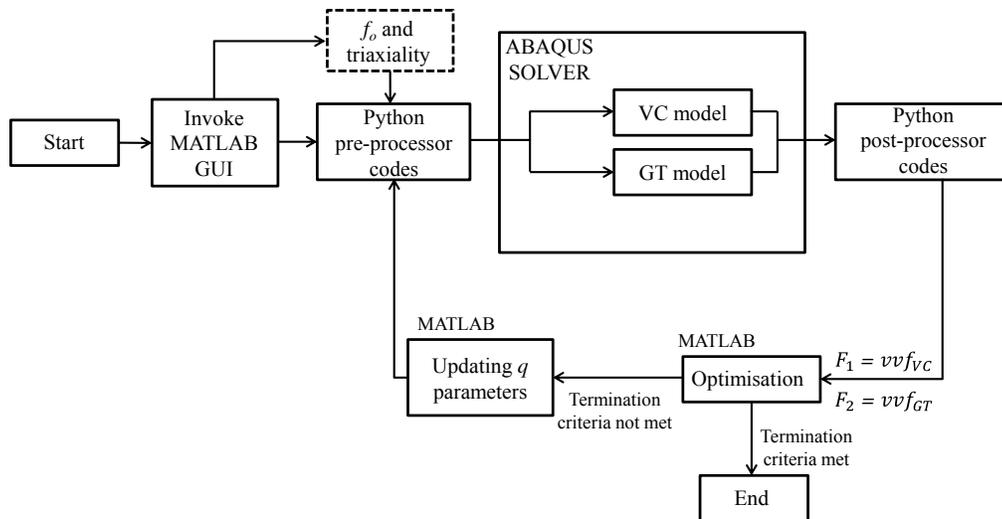
***Figure 4.23**- Illustration of the calibration procedure.*

4.2.4 The effect of *q* parameters on *vvf* rate

Before optimising the q parameters, it is necessary to study the individual effects of q_1, q_2 and q_3 on the void volume fraction rate (*vvf* versus time). Initially the void volume fraction versus time for the GT model explained in section 4.2.2 is obtained for a triaxiality, $T = -1$, initial void volume fraction, $f_o = 0.005$, and $q_1 = 1.5$, $q_2 = 1$ and $q_3 = 2.25$. To investigate the effect of q_1, parameters q_2 and q_3 have been fixed (to 1.0 and 2.25, respectively) and q_1 is changed by 50% ($q_1 = 2.25$) and 100% ($q_1 = 3.0$). The *vvf* versus time curves for these q parameters are shown in Figure 4.24a. It is apparent from this figure that increasing the q_1 parameter noticeably increases the void volume fraction rate.

Figure 4.24b shows the effect of q_2 parameter, where q_1 and q_3 have been fixed to 1.5 and 2.25, respectively and the q_2 parameter is changed by 50% ($q_2 = 1.5$) and 100% ($q_2 = 2.0$). It is evident from this figure that increasing the q_2 parameter significantly increases the *vvf* rate.

To study the effect of q_3 parameter, q_1 and q_2 have been fixed to 1.5 and 1 respectively and the q_2 parameter is changed by 50% ($q_3 = 3.375$) and 100% ($q_3 = 4.5$). The *vvf* versus time curves for these q parameters are shown in Figure 4.24c, where almost no sensitivity to the q_3 parameter is observed. To further investigate this, the q_2 parameter is changed by -50% ($q_3 = 1.125$) and 500% ($q_3 = 11.25$) as shown in Figure 4.24c. However, the *vvf* versus time curve still shows no sensitivity to the q_3 parameter.

To ensure the results obtained in Figure 4.24c are not only specific to triaxiality $T = -1$, another triaxiality was also examined. In Figure 4.25, for a triaxiality of $T = -0.5$ the q_1 and q_2 parameters have been fixed to 1.5 and 1, respectively, and the q_2 parameter was changed by 50%, 100%, -50% and 500%. This figure also shows that the *vvf* rate is almost insensitive to q_3. Therefore, the conclusion made from Figure

4.24c on the insensitivity of the *vvf* versus time to q_3 is not only specific to one triaxiality.
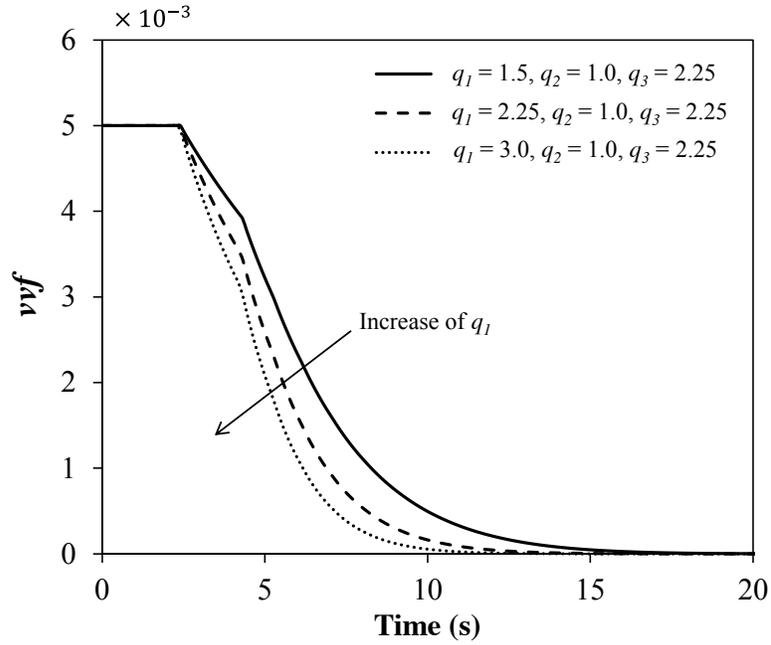
(a)
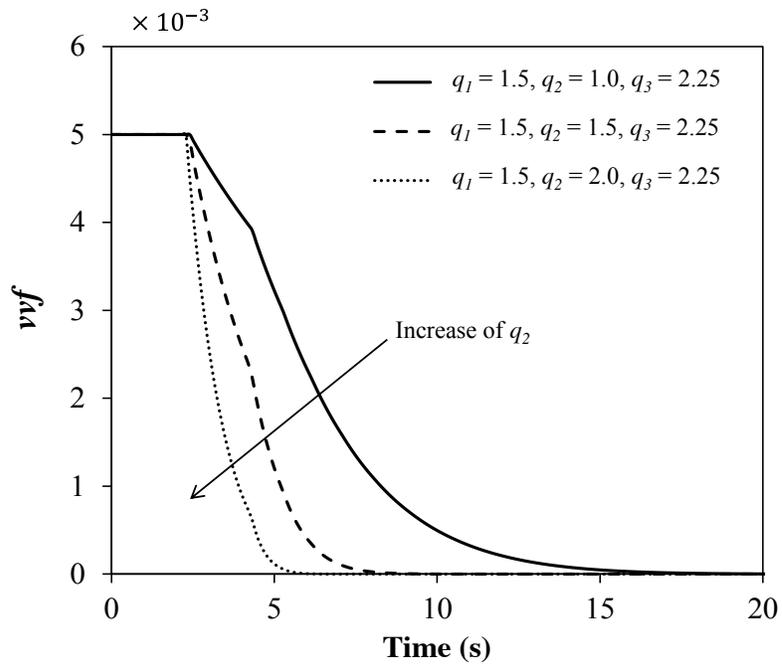
(b)

Figure continued on the next page.
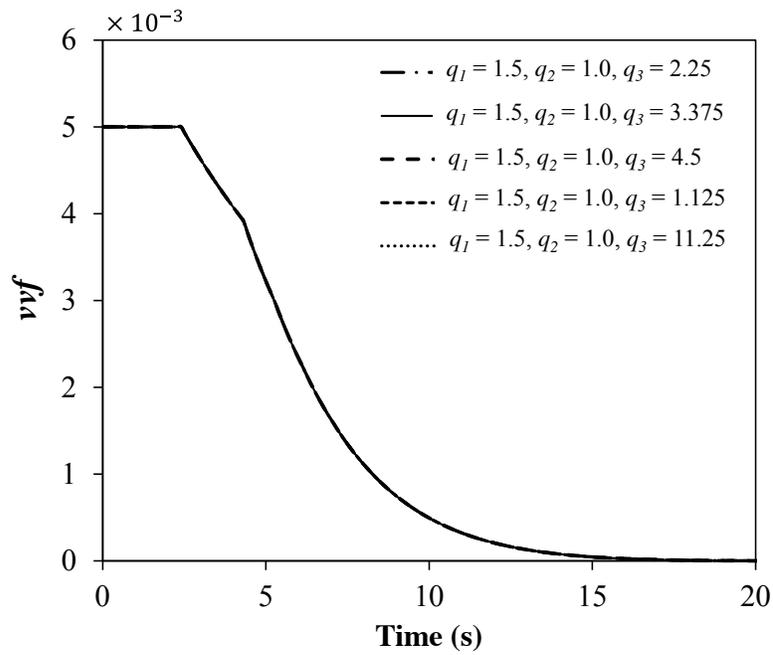
(c)

Figure 4.24*-The effect of (a) q_1, (b) q_2 and (c) q_3 on vvf versus time for triaxiality $T = \sigma_h/\sigma_e=$*

-1.
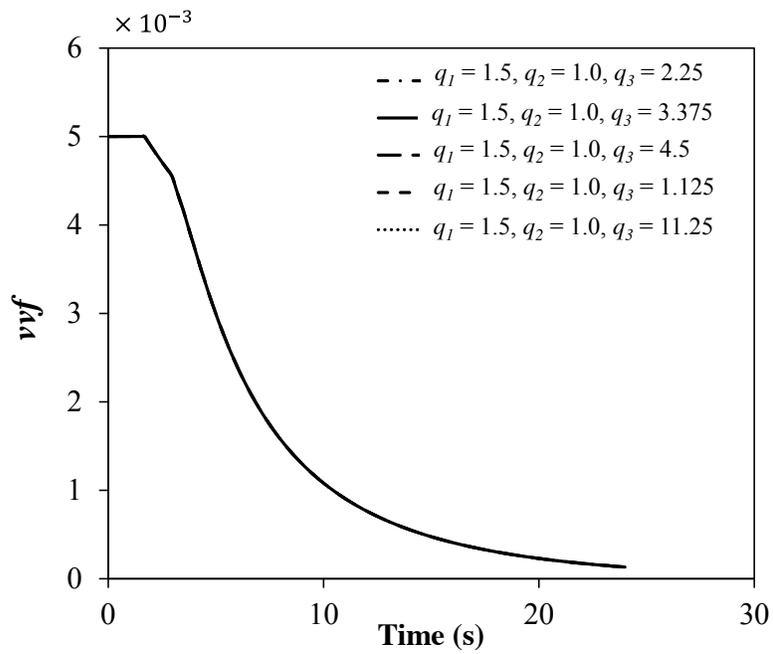
Figure 4.25*- The effect of q_3 on vvf versus time for triaxiality $T = \sigma_h/\sigma_e = -0.5$.*

94

4.2.5 Optimisation procedure

The aim of the optimisation process is to find the q parameters that minimise the difference in a least squares sense between the void volume fraction predicted by the GT FE model (shown as F_1 in Figure 4.26) and that of the corresponding discrete voided cell (VC) FE model (shown as F_2 in Figure 4.26). It should be noted that the curves in Figure 4.26 are just schematics used for illustration purposes and are not representing the output results of any FE simulation.
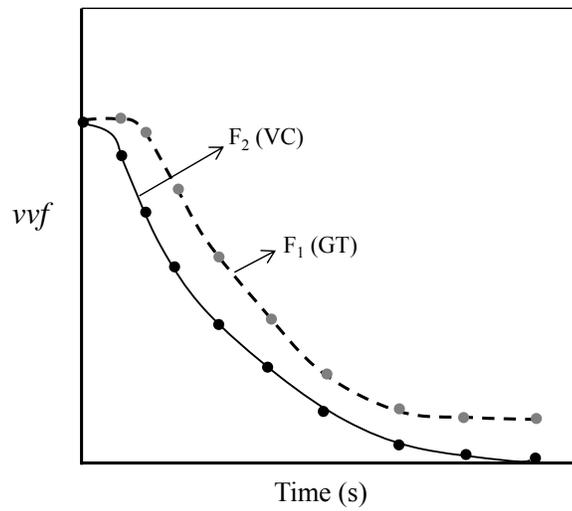
Figure 4.26- *Schematic illustration of the optimisation problem, where* F_2 (VC) *is a fixed curve (target curve) and* F_1 (GT) *is a function of the q parameters.*

The GT model takes the q parameters as input to predict the void volume fraction:

$$F_1 = F_1(q); \qquad q = q(q_1, q_2, q_3) \tag{4.9}$$

Therefore, the objective is to find $q = q^*$ in such a way that the difference between the two curves is minimised. This will be achieved by minimising a cost function (objective function), $R(q)$ defined in equation 4.10,

$$R(q) = \frac{1}{2} \sum_{i=1}^{M} [g_i]^2 \tag{4.10}$$

where M is the total number of data points ($M = 10$ in Figure 4.26) and $g_i(q)$ is:

$$g_i(q) = F_1^i - F_2^i \qquad\qquad 4.11$$

There are various types of optimisation methods available in the literature. These methods can be categorised into gradient based and direct search methods depending on the procedures employed to find the optimised values. In gradient based methods both the cost function and the gradient of the cost function are required to evaluate the optimum point (Newton type methods belong to this category). Search methods, on the other hand, do not require knowledge of the gradient of the objective function and conduct the optimisation only based on the cost function value. Simplex method, Pattern Search and Genetic Algorithm (GA) are examples of this category. In terms of the controls required for optimisation, the methods are categorised into constrained and unconstrained. Simplex method is an example of an unconstrained method where no search range or constraint is required to be defined to find the optimum point. Therefore, the outcome of these optimisation methods may sometimes be totally unreasonable and may converge to an unrealistic value.

Furthermore, some optimisation methods converge to a local minimum rather than a global minimum. Gradient based methods and some of the search methods such as Simplex method are of this type. Well-known examples of methods that are designed to converge to global minimum include Pattern Search, Genetic Algorithm, and Simulated Annealing. These methods work for both smooth and non-smooth cost functions. However, there are some other global methods that are only limited to smooth cost functions and thus have not been mentioned here.

The cost function in this study (defined in equation 4.10) does not have an explicit analytical definition and properties (it is a black-box function with only the input and output known). The gradient of this function is only available if numerical differentiation methods such as finite difference are employed. For this reason and also due to the local minima convergence issues of the gradient based methods, they

were ruled out for use in this study. The Simplex method, as already mentioned, is unconstrained and is a local solver and therefore is not suitable for this application which requires finding the global minimum. Amongst the global methods, GA is more efficient than the Simulated Annealing but less efficient than Pattern Search. GA is also a stochastic method and by repeating the optimisation, the solution might converge to a different value. Therefore, the Pattern Search method was chosen as the optimisation method for all the studied cases and will be compared with GA for one case to demonstrate its superiority for this application.

It is important to note that although the global methods attempt to locate a global solution, no solver employs an algorithm that can absolutely guarantee a solution as global (MATLAB documentation (2011)) and sometimes might converge to a local minimum. This is an outstanding area of research, and is an issue with all current optimisation methods. The impact of this inherent uncertainty is discussed at the end of section 4.2.

The expression 'Pattern Search' was first used by Hooke and Jeeves (1961). However one of the most basic forms of Pattern Search method had been used before in 1959 (Davidon, 1991).

The Pattern Search method is in the category of direct search methods that requires neither analytical nor explicit derivations of the cost function. The Pattern Search finds a sequence of points $x^1, x^2, \ldots, x^k$, where k is the iteration number. The update of the q parameters is done based on a series of exploratory moves to reach the optimal point, and iterations are maintained until a predefined condition based on changes in the parameters, or the cost function, is satisfied (Torczon, 1997).

Assume that x^0 is the initial estimate of the N parameters. A pattern of points is defined as,

$$[q^k]_{2N \times N} = \begin{bmatrix} x^{k-1} \\ x^{k-1} \end{bmatrix}_{2N \times N} + [s^k]_{2N \times N} \qquad 4.12$$

in which s is the mesh size and is defined as,

$$s_m = \begin{bmatrix} \propto I \\ -\propto I \end{bmatrix}_{2N \times N}, I = \begin{bmatrix} 1 & 0 & \cdots \\ 0 & 1 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}_{N \times N} \qquad 4.13$$

The parameters for next iteration would be updated as follows,

$$q_{min} = \mathrm{argmin} F(q^k) \qquad 4.14$$

Initially, the scaling factor, $\propto$, is considered as one. If $F(q_{min}) < F(x^{k-1})$ the poll is called a *successful* poll, otherwise the poll would be an *unsuccessful* one. For each successful poll, $\propto$ will be doubled and for each unsuccessful poll, $\propto$ will be halved. Further details on Pattern Search method can be found in (Torczon, 1997).

The Pattern Search algorithm is available in MATLAB as a built in function. All the parameters required for the optimisation (such as the cost function and optimised value tolerances usually used as the termination criteria for the optimisation process) have been taken as the default parameters suggested in MATLAB documentation (2011). These parameters are summarised in Appendix C part i.

4.2.6 Testing the optimisation method

To investigate the performance of the Pattern Search method for the calibration of the q parameters, a void volume fraction versus time curve was obtained by running the GT FE model (explained in section 4.2.2) for $q_1 = 1.5$, $q_2 = 1$ and $q_3 = 2.25$ (stress triaxiality, $T = -1$, and initial void volume fraction, $f_o = 0.005$, were used for the model); This curve is marked as the 'Target curve' in Figure 4.28. The q values were then set to $q_1 = 1.0$, $q_2 = 1.0$ and $q_3 = 1.0$ (as the initial guess for the optimisation process) and the optimisation was conducted (as in section 4.2.3) to determine whether the optimisation method is capable of finding the target q parameters (i.e. $q_1 = 1.5$, $q_2 = 1$ and $q_3 = 2.25$) and reproducing the target curve. A search range of $0.9 < q < 4.0$ was used for all q parameters (the effect of search range is explored later). Figure 4.27a shows the optimisation process, where at each

iteration the cost function defined in equation 4.10 (where F_1 and F_2 are the target and the guess curves, respectively) is calculated. In this test the best point was found in 17 iterations: $q_1 = 1.5$, $q_2 = 1$ and $q_3 = 2.25$ (Figure 4.27b). These values are in exact accord with the q parameters used for the Target curve. In each iteration, if the cost function is not reduced by the new q parameters, the old set and the corresponding cost function are retained. Figure 4.28 compares the *vvf* versus time for $q_1 = 1.5$, $q_2 = 1$ and $q_3 = 2.25$ (Target curve), for initial guess values of $q_1 = 1.0$, $q_2 = 1.0$ and $q_3 = 1.0$ and the optimised values of q. This figure shows that the optimised curve is identical to the target curve.

(a)

(b)
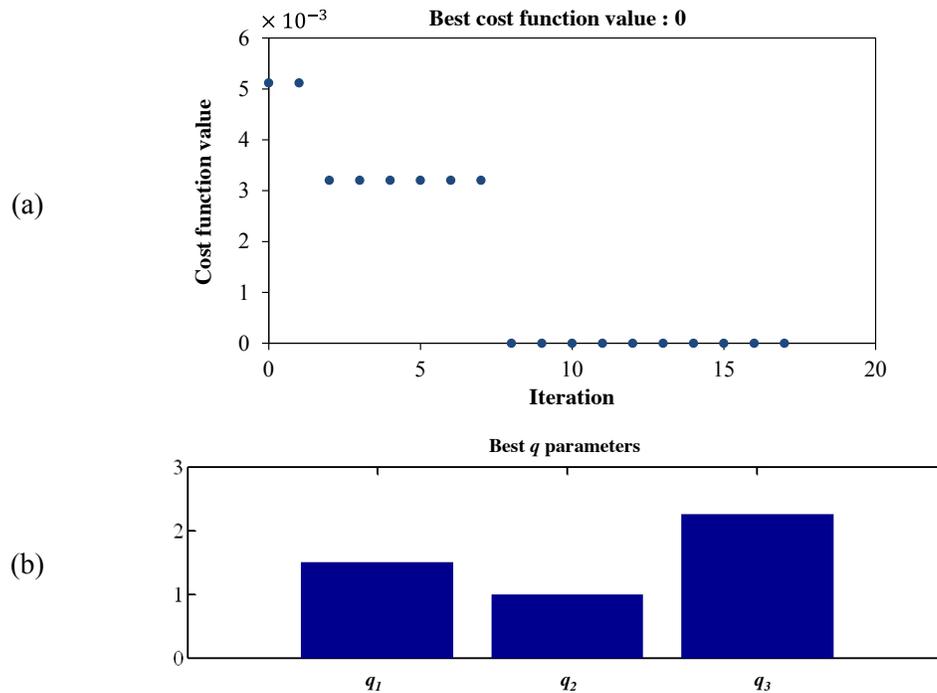
Figure 4.27*- (a) Cost function value at each iteration, (b) The best q parameters*.

The above findings confirm that the Pattern Search method is perfectly capable of finding the target values, at least for the range and guess values used. It is well known that some optimisation algorithms are sensitive to the initial guess. It is therefore important to investigate the initial guess sensitivity on the optimisation method used. This has been investigated later in this section.
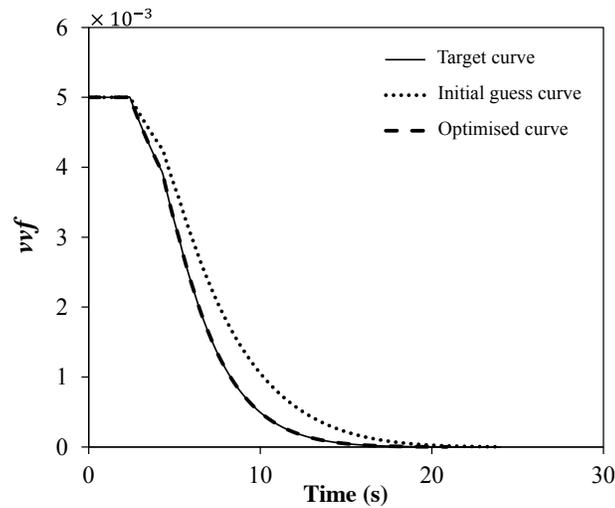
Figure 4.28*- Comparison of target, initial guess and optimised curves.*

Comparison with Genetic Algorithm (GA)

This section compares the performance of Pattern Search method with Genetic Algorithm (GA) which is an evolutionary algorithm to search for global minima. GA works based on techniques inspired by natural evolution. It is an iterative process starting from a population of randomly generated individuals. The population in each iteration is called a generation. GA only passes the best population of each generation to the next generation by using the cross-over and mutation of individuals of a generation to make children for the next generation. Further details on genetic algorithms can be found in (Conn et al., 1991, Goldberg, 1989).

The target curve explained in the previous section was used and the optimisation was conducted using Genetic Algorithm to determine whether the optimisation method is capable of finding the original q parameters. The same search range and the same termination criteria as in the previous section were used (full description of the parameters used for both GA and Pattern Search can be found in Appendix B).

Figure 4.29 shows the optimisation process, where each data point corresponds to the cost function defined in equation 4.10 for the best individual in the generation (an individual is any point to which you can apply the cost function and a generation is an array of individuals). It should be noted that as indicated in Appendix C, part ii

the size of the population of 20 per variable is suggested as the default value in MATLAB. Since the optimisation is conducted for 3 variables (q_1, q_2 and q_3) the total population of each generation is $3\times20 = 60$. This means each data point shown in Figure 4.29 represents up to 60 function evaluations (ABAQUS runs). In practice the total function evaluations are less than this maximum value. This is due to the fact that some of the individuals of the current generation are present in the next generations. In this study a total number of 1347 ABAQUS runs were performed. This shows how computationally expensive this method is.

In Figure 4.29 the best fitness value (cost function) of 1.01×10^{-5} was found after 51 generations. This corresponds to the optimised parameters of $q_1 = 1.52$, $q_2 = 0.99$ and $q_3 = 1.48$. This indicates errors of 1.33%, -1.0% and -34.2% in q_1, q_2 and q_3 determinations, respectively. Comparing these results to the Pattern Search results which has 0% error in finding the target values and require about 60 ABAQUS runs, it is concluded that the Pattern Search is more suitable for this application.

(a)

(b)
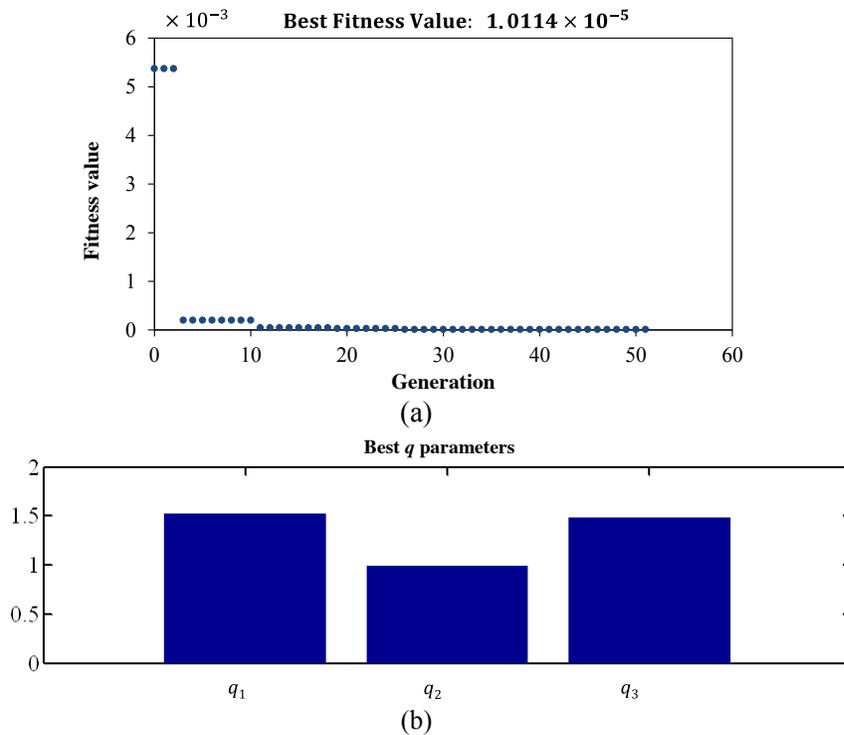
Figure 4.29-*(a) Best cost function value for each generation, (b) The best q values.*

The effect of initial guess

Previously the q parameters were set to $q_1 = 1.0$, $q_2 = 1.0$ and $q_3 = 1.0$ as the initial guess for the optimisation process. To investigate the sensitivity of the Pattern Search method to the initial guess, two additional sets of q parameter, as summarised in Table 4.2 were also examined for the same search range. The optimisation was performed for all the considered cases. The cost function values at each iteration are shown in Figure 4.30.

Table 4.2-The initial guess values tested.

	Initial guess values		
	q_1	q_2	q_3
Case I (studied previously)	1.0	1.0	1.0
Case II	2.0	2.0	2.0
Case III	3.0	3.0	3.0

(a)

(b)
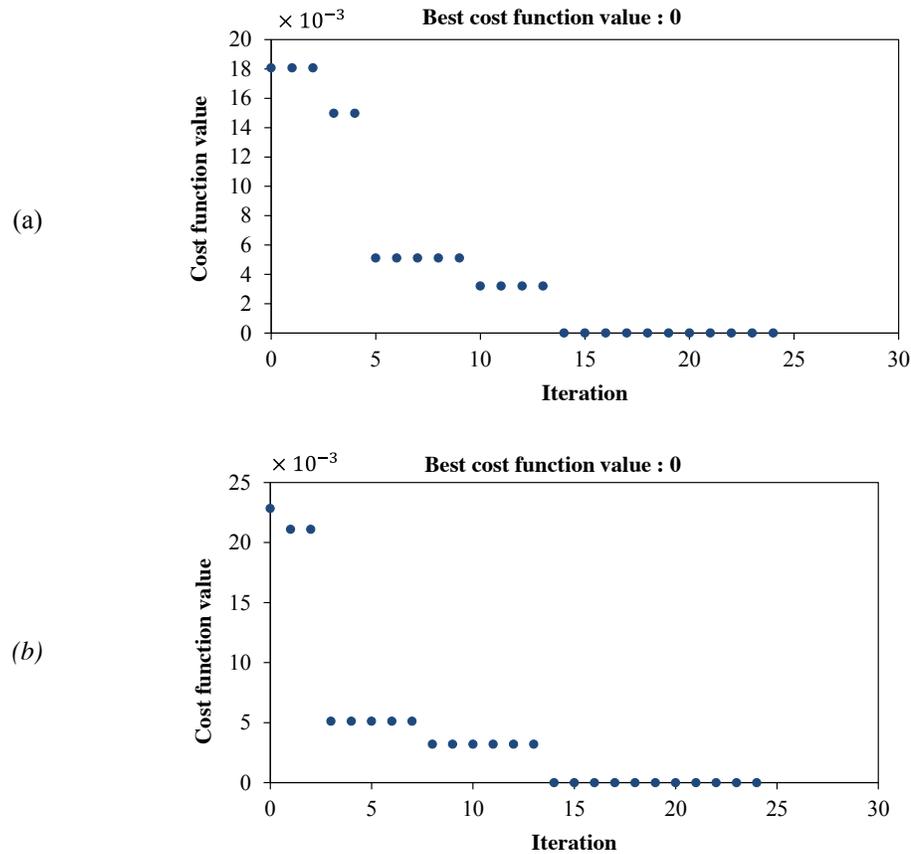
Figure 4.30- Cost function values at each iteration (a) Case II, (b) Case III.

Figure 4.31a compares the *vvf* versus time before and after optimisation for Case II with the target curve ($q_1 = 1.5$, $q_2 = 1$ and $q_3 = 2.25$). Similarly, Figure 4.31b compares the *vvf* versus time before and after optimisation for case III and the Target curve. It is evident from both figures that the optimised curves are identical to the target curve. The q parameters obtained from optimisation for both cases are identical to the q parameters used for the target curve. Therefore, the optimisation method used in this study is not sensitive to the initial guess values for the studied cases. However, it is evident that cases II and III converge more slowly than case I.

(a)

(b)
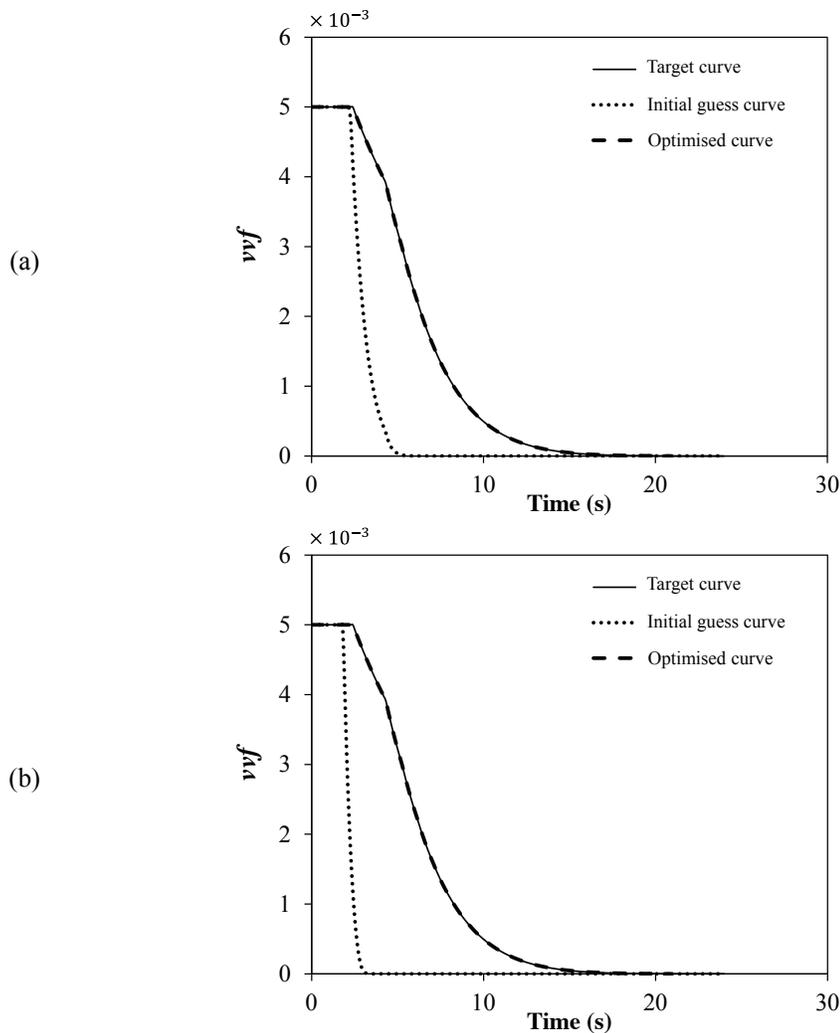
Figure 4.31 - *Comparison of the target, initial guess and optimised curves for (a) Case II (initial guess: $q_1 = 2.0$, $q_2 = 2.0$ and $q_3 = 2.0$) and (b) Case III (initial guess: $q_1 = 3.0$, $q_2 = 3.0$ and $q_3 = 3.0$.)*

The effect of search range

So far all optimisations were performed for a search range of $0.9 < q < 4.0$. To investigate the sensitivity of the method to the search range, the optimisation of the q parameters for Case I (initial guess values of $q_1 = 1.0$, $q_2 = 1.0$ and $q_3 = 1.0$) is repeated but for an expanded search range of $0.1 < q < 10.0$.

Figure 4.32a shows the optimisation process, where at each iteration the cost function defined in equation 4.10 (where F_1 and F_2 are the target and the guess curves, respectively) is calculated. The best point was found in 37 iterations: $q_1 = 1.5$, $q_2 = 1$ and $q_3 = 2.25$ (Figure 4.32b). These values are in exact accord with the q parameters used for the Target curve. Therefore, by expanding the search range, the applied optimisation method is still capable of finding the target values.

(a)

(b)
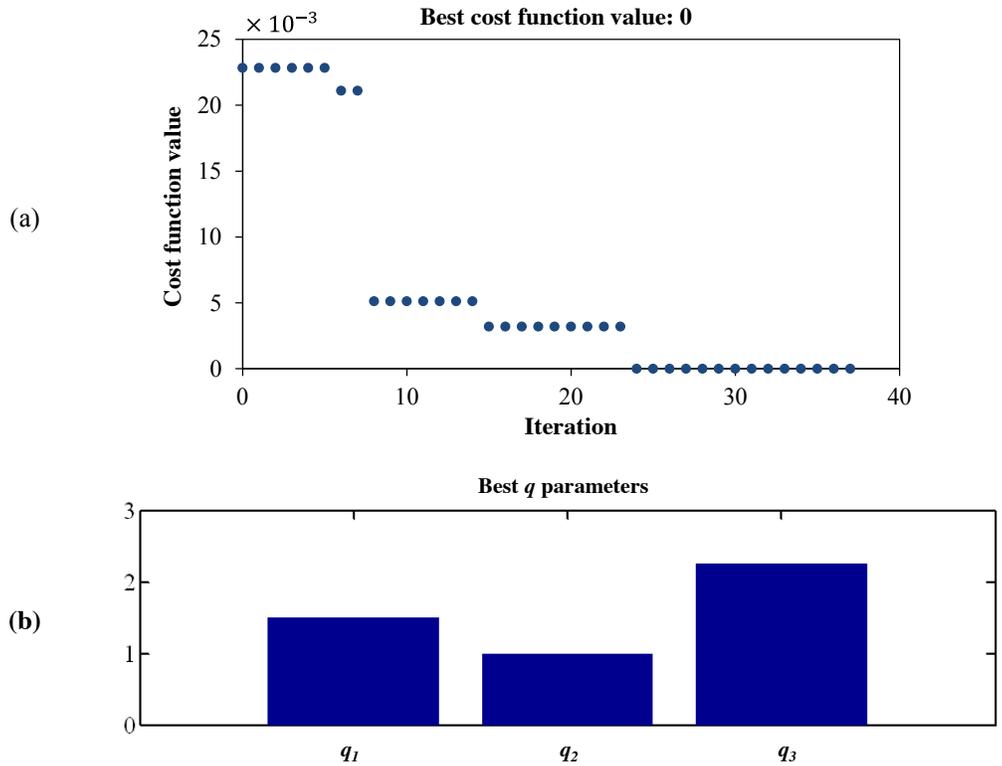
Figure 4.32 - (a) Cost function value at each iteration, (b) The best q parameters.

4.2.7 Results and discussion

As discussed in section 4.2.5, a common concern when dealing with optimisation algorithms is whether the technique is able to find the absolute minima. If the objective function (cost function) has a large multiplicity of local minima, the approach may be trapped very far away from the solution. Knowing the properties of the cost function influences the choice of the best numerical technique used to minimize the cost function. If the cost function has a large multiplicity of local minima the only solution will be to use algorithms which are more likely to find the global minima, such as Pattern Search and Genetic Algorithm.

The cost function optimised in this study (equation 4.10) does not have an explicit analytical definition and properties. Also, as shown in the previous sections, a very large number of function calls (ABAQUS runs) are required for the GA to perform the optimisation. It was shown that considering the running times for the ABAQUS models, application of GA is not practical and the Pattern Search was shown to be more efficient. However, there is no guarantee that the Pattern Search or any other global method always converges to the global minima. There are some situations that the solution is trapped in a local minimum. An example of this has been shown below.

Two different search ranges were studied (triaxiality T = -0.5 was used). Table 4.3 summarises the tested search ranges, the best cost function values and q parameters obtained after optimisation for each case. In Table 4.3, Case A corresponds to a search range which is symmetric with respect to the commonly suggested values, q_1 = 1.5 and q_2 = 1.0 and explores $1.0 < q_3 < 4.0$ based on the assumption of $q_3 = q_1^2$ and Case B corresponds to a very wide search range. The cost function versus iteration plots for both cases is shown in Figure 4.33.

Table 4.3-*Different search ranges tested.*

	Search range			Best (q_1,q_2,q_3)	Best Cost Function value
	q_1	q_2	q_3		
Case A	[1.0, 2.0]	[0.75, 1.25]	[1.0, 4.0]	(2.0, 1.24, 4.0)	0.0035
Case B	[0.1, 10]	[0.1, 10]	[0.1, 10]	(3.13, 1.0, 10.0)	0.0036

(i)

(a)

(b)

(ii)

(a)

(b)
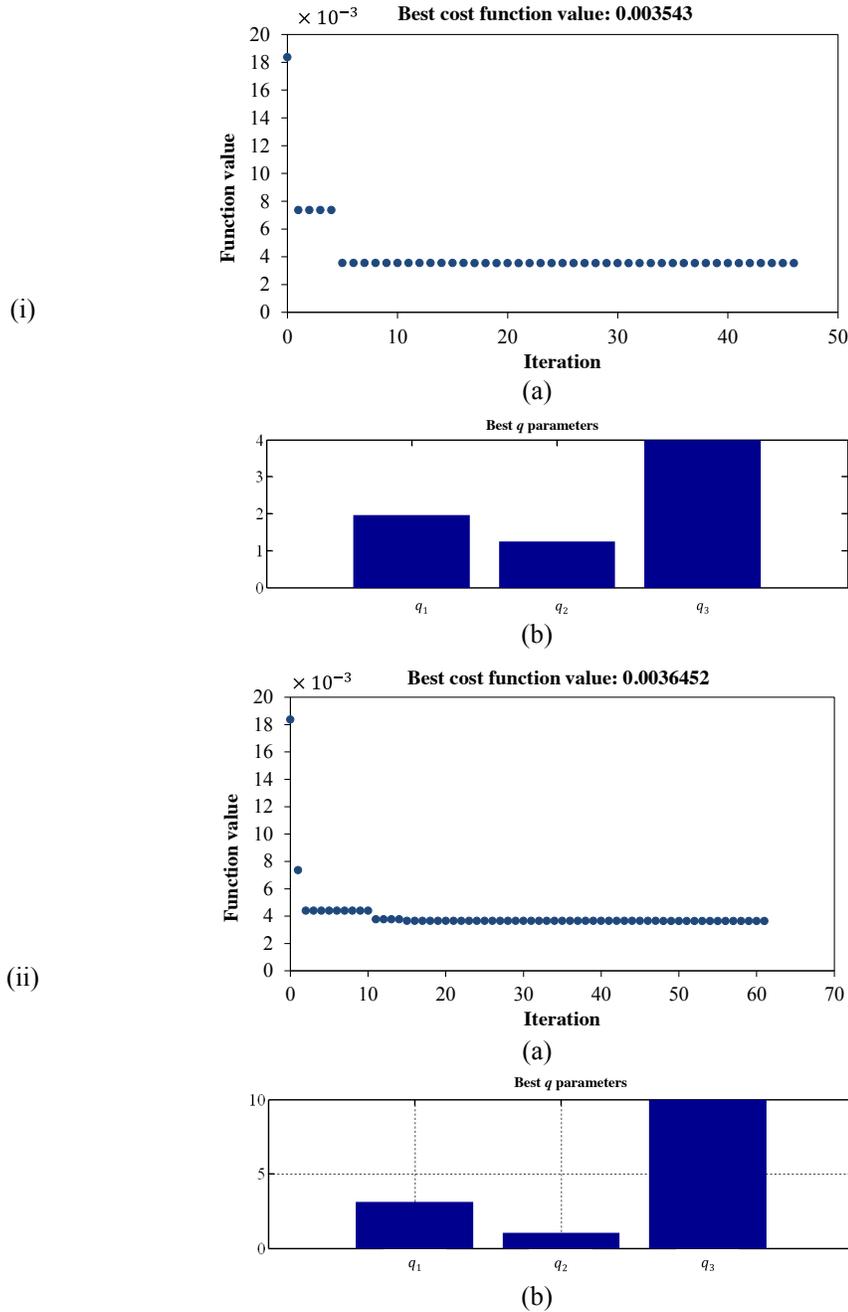
Figure 4.33- *(a) Cost function value at each iteration for (i) Case A and (ii) Case B, (b) The best q parameters for (i) Case A and (ii) Case B.*

The results in Table 4.3 and Figure 4.33 indicate that by expanding the search range, the optimisation has found different optimised q parameters and cost function values for this case. Figure 4.34 compares the void volume fraction, *vvf*, versus time after optimisation for cases A and B. Since the difference in the cost function values for both cases is very small the plots are almost identical. Therefore, the q parameters obtained in both Case A and Case B can be considered suitable for modelling the material, and the set of best q parameters is shown to be not necessarily unique. However, it should be noted that the expanded range studied in Case B is an uncommonly broad range and the q values are far away from common values found in the literature. Therefore, to avoid finding unreasonable optimised values the range of q parameters was set to the one used in Case A. However, it is worth emphasising that the q parameters are dimensionless and do not have physical meaning, so both sets are suitable from the standpoint of the material behaviour.
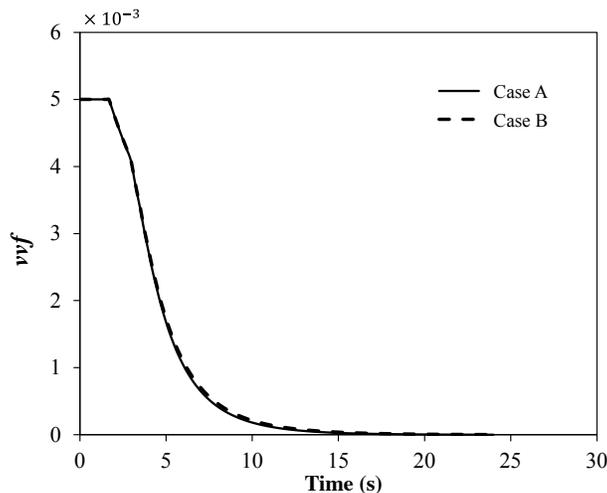
Figure 4.34- *Comparison of vvf versus time for Case A and Case B.*

Figure 4.35 to Figure 4.38 show the results of optimising the GT model parameters for triaxialities in the range $-2.0 \leq T \leq -0.5$. The graphs show void volume fraction (*vvf*) versus time (deformation), beginning from an initial value of 0.005 to a final void volume fraction near zero, for the discrete void cell (VC) model and the GT cell model before and after the optimisation. The values of *vvf* coincide for all models early in the deformation (i.e. the GT parameters have no or little effect for small

strains, when the material is elastic or has a small plastic strain). A considerable improvement in the GT behaviour is obtained upon optimisation of the q_1 and q_2 parameters, although an exact match is not possible.
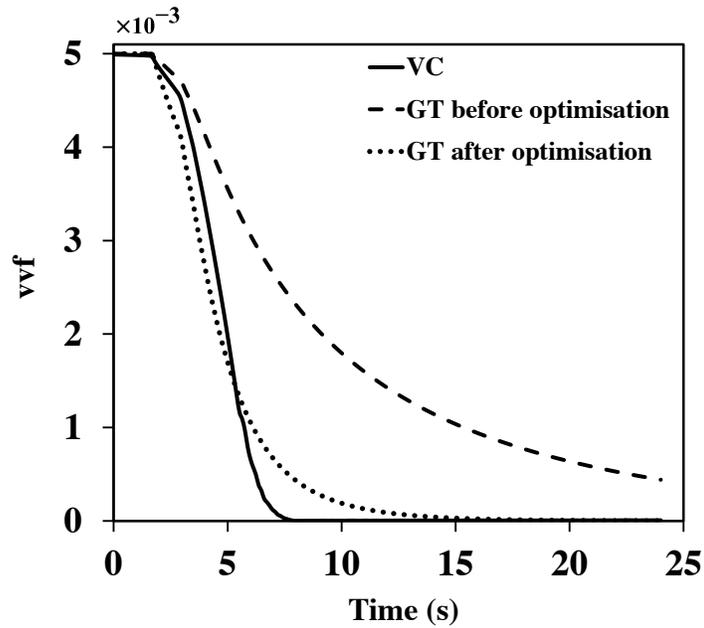
Figure 4.35- *vvf vs time for VC and GT model before and after optimisation at Triaxiality $T = \sigma_h/\sigma_e$ = -0.5.*
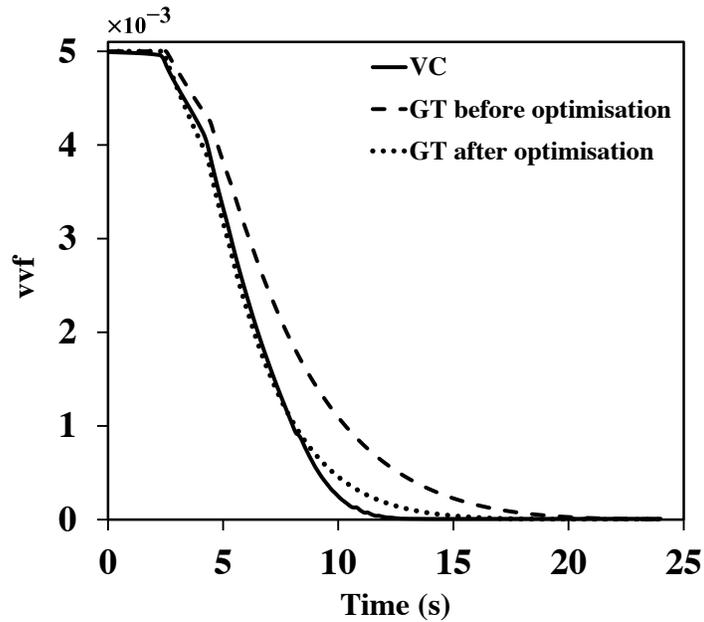
Figure 4.36- *vvf versus time for VC and GT model before and after optimisation at triaxiality $T = \sigma_h/\sigma_e$ = -1.0.*

108
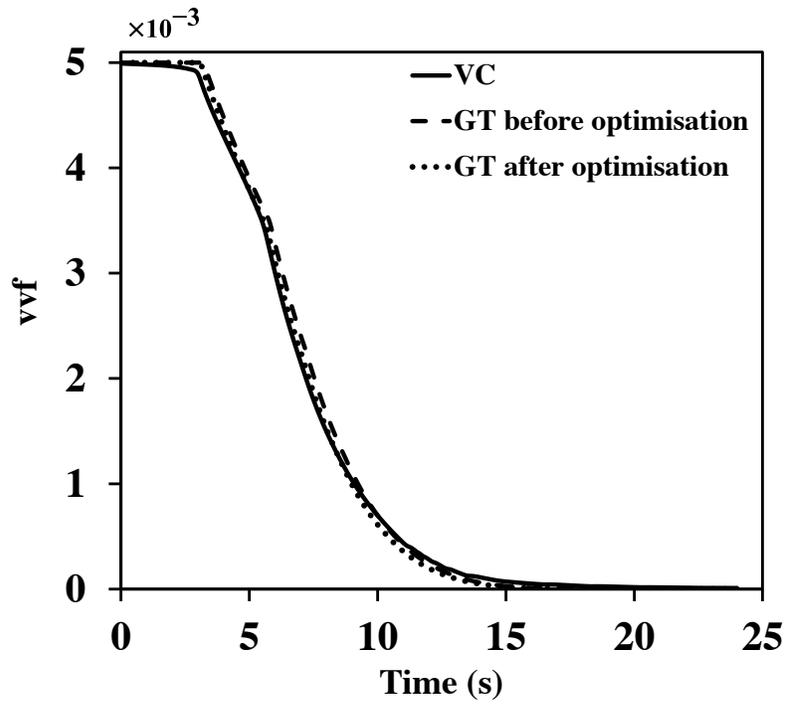
Figure 4.37 *- vvf versus time for VC and GT model before and after optimisation at triaxiality $T = \sigma_h/\sigma_e = -1.5$.*
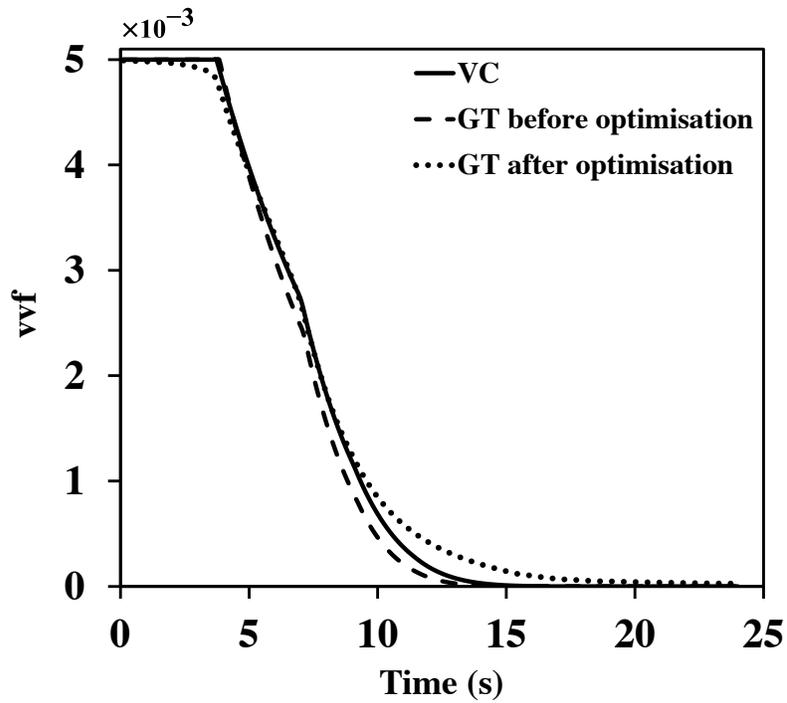
Figure 4.38 *- vvf versus time for VC and GT model before and after optimisation at triaxiality $T = \sigma_h/\sigma_e = -2.0$.*

Performance of the optimised GT model is best when $T = -1.5$, exhibiting close correlation with the discrete void cell model for the entirety of the void closure process, predicting accurately even final void closure, and is generally very accurate for $T \leq -2$. However, as T becomes increasingly negative, the void remains more equiaxed throughout the void closure process, hence the original Gurson values of the parameters ($q_1 = q_2 = q_3 = 1.0$) also provide a reasonably accurate behaviour ($T = -2$ is an example, in Figure 4.38). The results of the parameter fits for various triaxialities, including the final value of the cost function, are summarised in Table 4.4.

Table 4.4. *Optimised q values for different triaxialities*

T	q_1	q_2	q_3	Cost Function (equation 4.10)
-0.3	2.0	1.25	2.0	0.007
-0.4	2.0	1.25	4.0	0.0048
-0.5	2.0	1.25	4.0	0.0035
-0.6	1.62	1.25	4.0	0.0031
-0.7	1.73	1.12	4.0	0.0027
-0.8	1.97	1.0	4.0	0.0024
-0.9	1.74	1.0	4.0	0.0018
-1.0	1.5	1.0	4.0	0.0013
-1.5	1.0	1.0	2.0	0.0008
-2.0	1.04	0.94	2.0	0.0016

Table 4.4 shows that the cost function decreases as triaxiality decreases, indicating the optimized GT model performs better (regarding the discrete VC model as reality) when the triaxiality is sufficiently negative that the void remains roughly spherical for the majority of the deformation. Deviation occurs late in the closure process

when the hole becomes non-spherical prior to closing. The best response can be seen at $T = -1.5$, which is the triaxiality at which the void remains spherical.

Just for the sake of completion, it should be noted that instead of optimising all three q parameters, it is possible to optimise only q_1 and q_2, with q_3 considered to be equal to q_1^2 (Tvergaard, 1981, 1982). The optimisation has been performed based on this assumption for triaxiality of $T = -1$ and initial void volume fraction of $f_o = 0.005$. Figure 4.39 shows the optimisation process where an optimum point of $q_1 = 1.55$, $q_2 = 1.0$ (and $q_3 = q_1^2 = 2.40$) has been reached resulting in a cost function of 0.0013. The values for q_1, q_2 and the cost function are in very close agreement with the results obtained when optimising all three q parameters ($q_1 = 1.5$, $q_2 = 1.0$ and cost function = 0.0013, see Table 4.4) but the q_3 parameters are significantly different. However, as discussed in section 4.2.4, the *vvf* behaviour is almost insensitive to the q_3 parameter. It was predicted that no significant difference should exist between the results obtained from either of the approaches. Figure 4.39 confirms this prediction.
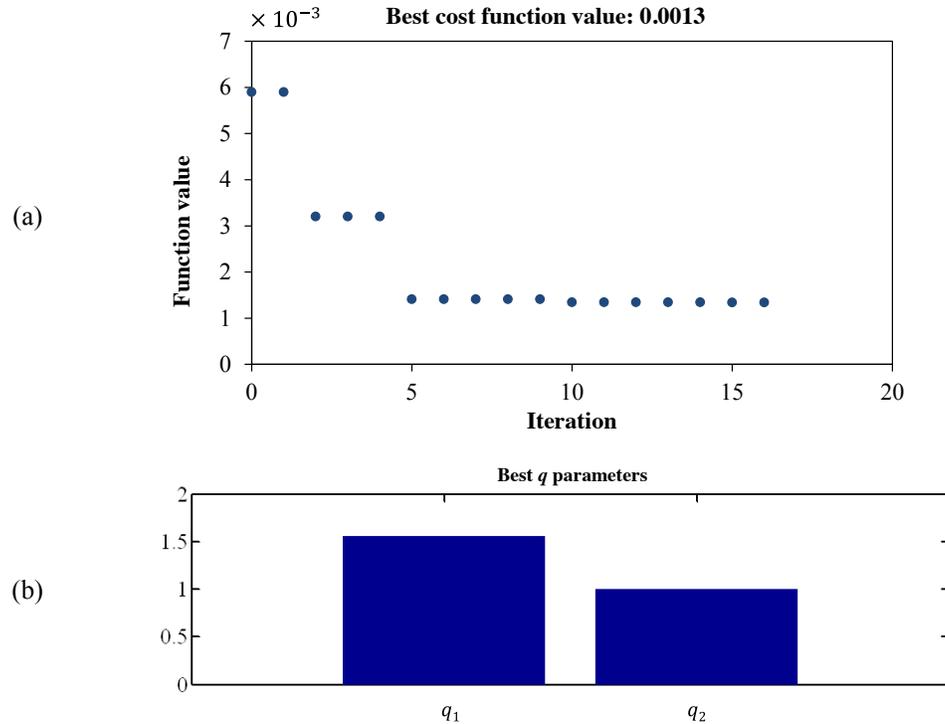
Figure 4.39- *(a) Cost function value at each iteration, (b) The best q parameters.*

111
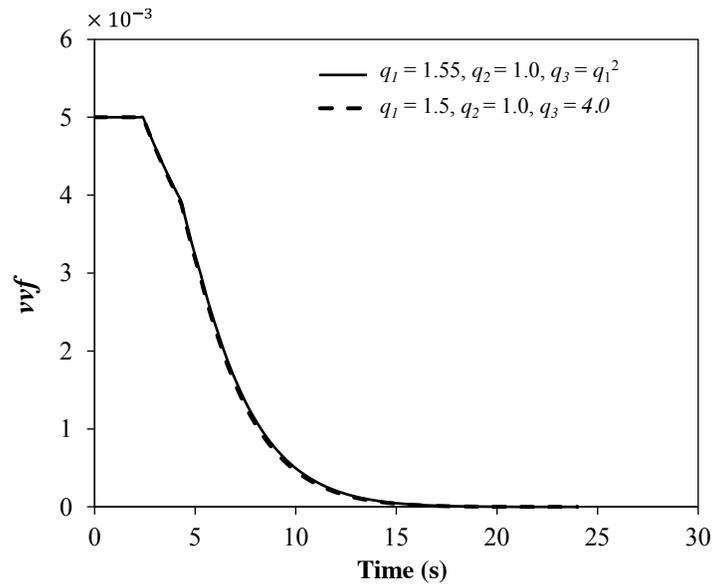
Figure 4.40- The comparison of vvf versus time for the case where only q_1 and q_2 are optimised ($q_3=q_1^2$), and the case where all three q parameters are optimised.

4.3 Conclusions

A model has been created which predicts the closure load for a voided cell of specific void size and certain triaxiality. The effect of temperature and strain rate has been investigated. The results show that for high temperatures and low strain rates, less pressure would be required for void closure.

The applicability of the reverse Gurson model for void closure is examined. Finite element analyses are performed on a representative material volume (RMV) containing a spherical void and an equivalent cell of Gurson-Tevergaard (GT) material. The numerical results of the voided RMV and the GT cell are compared and an optimization method is used to find the *q* parameters which fit the results best.

The effect of triaxiality on the void closure process is investigated and the results show that the Gurson model has better agreement with the voided cell model for higher triaxialities. This is explained by the fact that Gurson assumes a spherical void shape which is more correct at high triaxialities.

112

4.4 References

Abaqus 2010. *In*: Dassault SYSTÈMES. 6.10-2 ed. USA

Brocks, W., Sun, D. Z. & Honig, A. 1995. Verification of the transferability of micromechanical parameters by cell model calculations with visco-plastic materials. . *International Journal of Plasticity*. , 11**,** 971-989.

Conn, A. R., Gould, N. I. M. & Toint, P. L. 1991. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, 28**,** 545–572.

Corigliano, A., Mariani, S. & Orsatti, B. 2000. Identification of Gurson–Tvergaard material model parameters via Kalman filtering technique. I. Theory. *International Journal of Fracture*, 104**,** 349-373.

Davidon, W. 1991. Variable metric method for minimization. *SIAM Journal on Optimisation*, 1**,** 1-17, (the article was originally published as Argonne National Laboratory Research and Development Report 5990, May 1959).

Foster, A. 2007. *Damage modelling in free machining steels*. Ph.D, University of Birmingham.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimzation & Machine Learning*, Reading, Addison-Wesley.

Hooke, R. & Jeeves, T. A. 1961. 'Direct Search' solution of numerical and statistical problems. *Journal of the ACM*, 8**,** 221-229.

Kim, J., Gao, X. & Srivatsan, T. 2004. Modeling of void growth in ductile solids: effects of stress triaxiality and initial porosity. *Engineering Fracture Mechanics*, 71**,** 379-400.

Koplik, J. & Needleman, A. 1988. Void growth and coalescence in porous plstic solids. *International Journal of Solid Structures*, 24**,** 835–853.

Kuna, M. & Sun, D. Z. 1996. Three-dimensional cell model analyses of void growth in ductile materials. *International Journal of Fracture*, 81**,** 235-258.

Mahnken, R. 1999. Aspects on the finite-element implementation of the Gurson model including parameter identification. *International Journal of Plasticity*, 15**,** 1111-1137.

Matlab 2011. Documentation R2011a, Global Optimisation Toolbox: Direct Search. Mathworks.

Pardoen, T. & Hutchinson, J., W. 2000. An extended model for void growth and coalescence. *Journal of Mechanics and Physics of Solids*, 48**,** 2467-2512.

Sims, R. B. 1954. The calculation of roll force and torque in hot rolling mills. *Proceedings of the Institution of Mechanical Engineers*, 168**,** 191-200.

Søvik. 1996. *A damage mechanics approach*. Ph.D, Norwegian University of Science and Technology.

Steglich, D. & Brocks, W. 1997. Micromechanical modelling of the behaviour of ductile materials including particles. *Computational Material Science,* 9**,** 7-17.

Torczon, V. 1997. On the convergence of pattern search algorithms. *SIAM Journal of Optimisation,* 7**,** 1-25.

Tvergaard, V. 1981. Influence of voids on shear band instabilities under plane strain conditions. *International Journal of Fracture* 17**,** 389-407.

Tvergaard, V. 1982. On localization in ductile materials containing spherical voids. *International Journal of Fracture,* 18**,** 237-252.

Vadillo, G. & Fernández-Sáez, J. 2009. An analysis of Gurson model with parameters dependent on triaxiality based on unitary cells. . *European Journal of Mechanics /A Solids*., 28**,** 417-427.

Zhang, Y. & Chen, Z. 2007. On the effect of stress triaxiality on void coalescence. *International Journal of Fracture,* 143**,** 105-112.

Chapter 5

Micromechanical modelling of void elimination: Healing

As mentioned in the previous chapters, the porosity elimination process can be investigated through two stages. The first stage is the pore closure when the pore shrinks to a slit (crack) as a result of compression. Upon initial contact of the pore surfaces, a series of voids form along the bonding surfaces, the scale of which depends on the pore surface roughness. It is assumed that these voids can be removed by diffusion and plastic deformation so allowing an intimate contact along the bond (the healing stage). A strong bond which has the mechanical properties of the original material is therefore obtained in the healing stage when the surfaces of the void are diffusion bonded. The modelling of pore closure has been described in the previous chapter. This chapter continues the porosity elimination process by modelling the second stage, void healing.

5.1 Experimental observation of the voids

The purpose of this section is to characterise the porosity distribution and shape to provide a better understanding of the assumptions made in the modelling process.

In steel casts the porosity is usually localized and non-uniformly distributed relative to the entire part and is more concentrated in the centre. As explained in Chapter 3 this kind of porosity forms due to solidification shrinkage. To observe these pores two billets, billet 1 and billet 2, from the centre of two different blooms were considered. Figure 5.1a shows a schematic of a 750×355×500 mm cast bloom and the cutting locations for billet 1 and billet 2. Macroscopic images of the cross section of billet 1, at two locations specified by cutting planes A and B (see Figure 5.1b) are shown in Figure 5.1c. It is evident from this figure that pores are concentrated in the centre of the bloom (pores are the dark spots in the image). To observe the porosity in the longitudinal direction, macroscopic images were taken from the 'af' edge of billet 2 (Figure 5.1d). It is evident also from this figure that porosity is concentrated in the centre of the bloom.

(a)

(b)

(c)

(d)
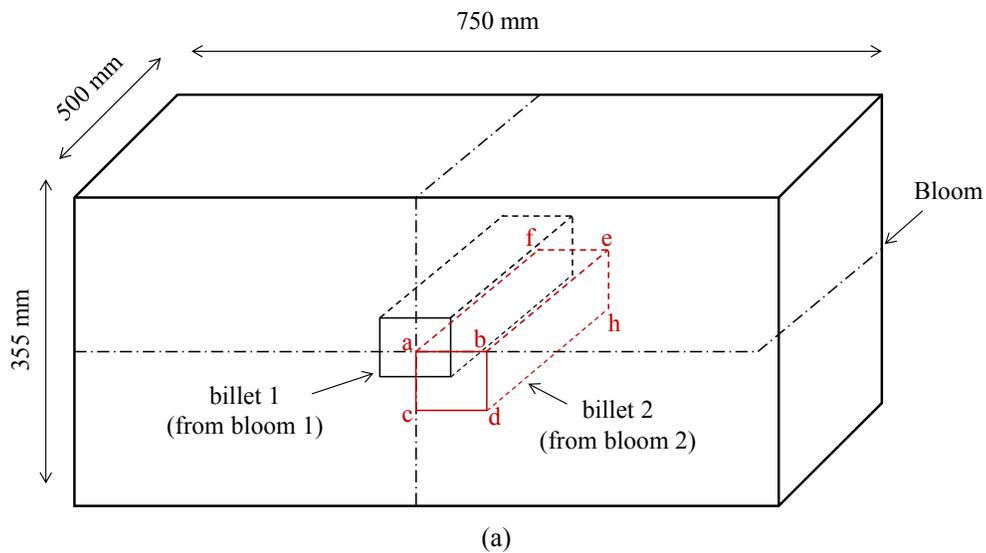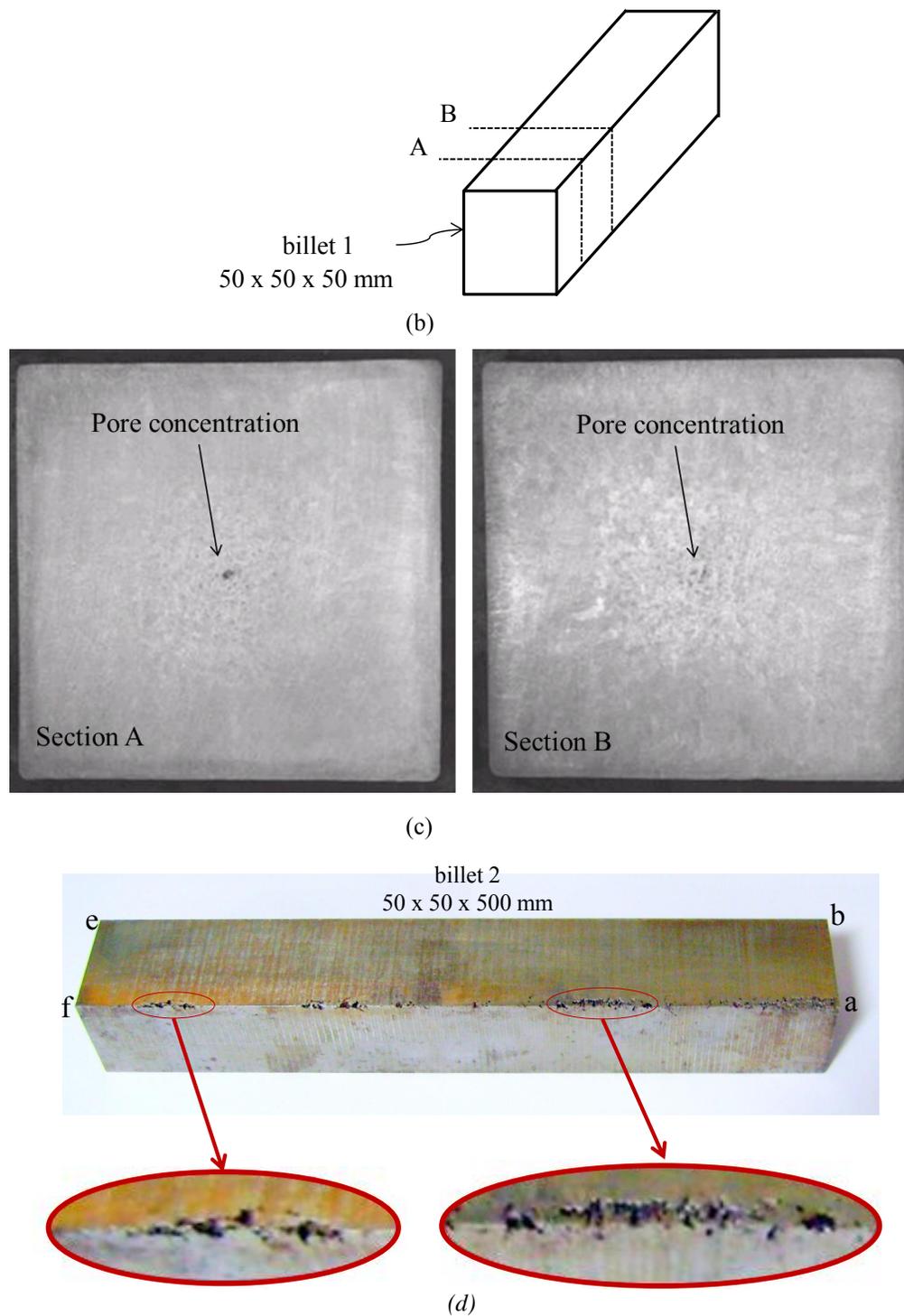
Figure 5.1-*(a) The cast bloom and the location of billets, (b) Billet 1 and the cutting planes*
A and B, (c) Macroscopic images of the cross section of billet 1 at cutting planes A and B,
(d) Pores distributed longitudinally at the centre of the bloom.

117

Although maximum porosity concentration is at the centre of a bloom, porosity is not limited to the central part only. Away from the centre, small patches of micro-porosity are likely to be seen. Porosity in this region reduces in size and concentration and has random shapes. Figure 5.2a shows a SEM image of porosity in a near surface region (billet 11 in Figure 5.2c), scale in μm. Figure 5.2d on the other hand, shows a pore in the central region (billet 31 in Figure 5.2c) scale in mm. It is worth mentioning that the surface of the bloom is very unlikely to contain porosity and is usually assumed to be non-porous (Farrugia, 2012).

(a)   (b)

(c)
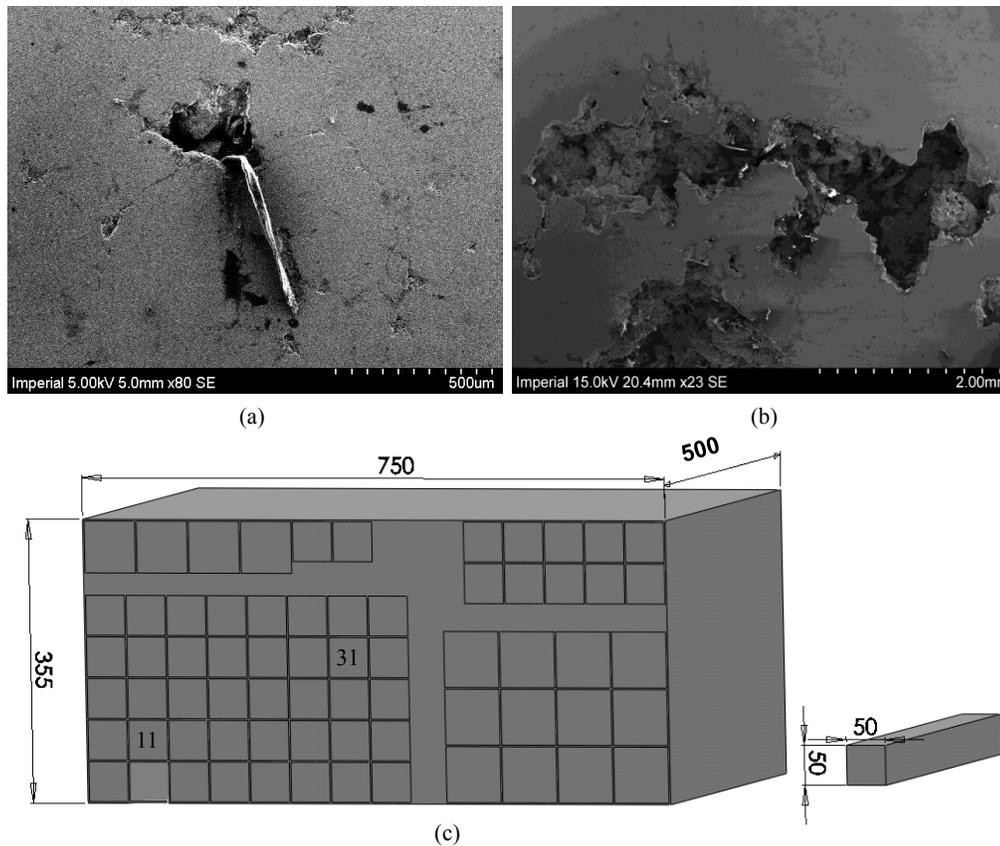
Figure 5.2- (a) Porosity at a near surface region of a free cutting steel (FCS) bloom, (b) Porosity in the central region of a FCS bloom, (c) Locations of billet 11 and billet 31 used for images (a) and (b), respectively.

5.2 Introducing the parameter *vvf$_c$* and its underpinning physics

For a non-porous solid material, plastic deformation will not change material volume. However, for a porous material under plastic straining in either tension or compression, the material volume will change. Figure 5.3a shows a bulk of material containing cavities of different shapes and sizes. Gurson assumes a "homogenisation" process by adding all voids into a single spherical void as shown in Figure 5.3b. In terms of finite element (FE) modelling, these voids will be homogeneously distributed in all elements as shown in Figure 5.3c. The initial void volume fraction of the material prior to plastic deformation is defined as:

$$vvf = \frac{volume\ of\ the\ voids}{volume\ of\ the\ material} \qquad\qquad 5.1$$

In reality	Homogenisation	FE modelling

(a)	(b)	(c)
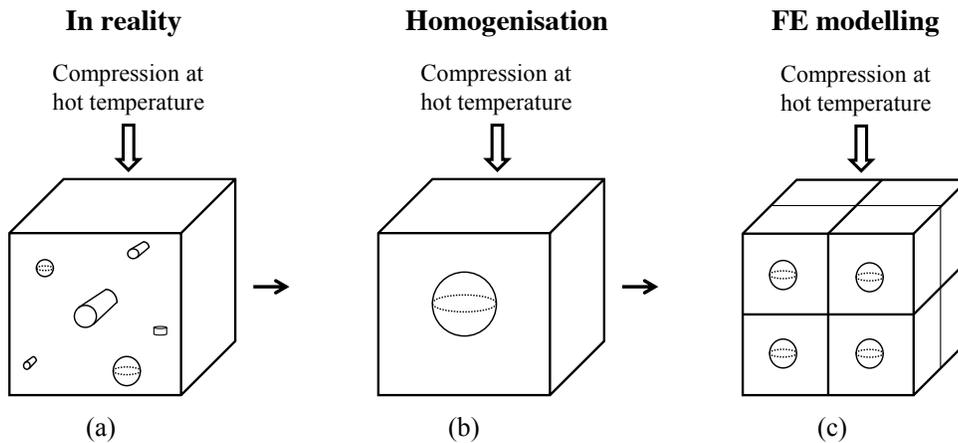
Figure 5.3*-(a) Voids of different shape and size in a material, (b) Homogenisation of voids into a single spherical void, (c) FE modelling of porous material.*

As explained in the previous chapters the first stage of void healing is assumed to be the void closure, when the void reduces to a crack as a result of compression as shown in Figure 5.4a. Following the compression stage, the void surfaces are still apart by their roughness. This is illustrated in Figure 5.4b for a closed cylindrical and spherical void. The initial contact of the asperities on the surfaces of the closed voids

(or cracks) creates a series of cylindrical voids or channels on a scale small compared to that of the original void (Figure 5.4c). It is assumed that these voids can be removed by diffusion bonding so allowing an intimate contact along the bond (the healing stage) assuming no oxidation is present (Derby and Wallach 1982; Monroe 2005). The diffusion bonding of these voids is predicted by the Pilling model here which assumes cylindrical shape for the voids (Pilling et al., 1984; Pilling and Ridely, 1987; Pilling, 1988). Details on the Pilling model can be found in Chapter 3. It is assumed that cylindrical pores within an element are agglomerated into a single cylinder, similar to the Gurson model assumption, as shown in Figure 5.4d.

Figure 5.4a shows a bulk of porous material under compression at hot temperature where pores have been closed and diffusion bonding is initiated. The void volume fraction at this stage is called *vvf$_c$* and is defined as:

$$vvf_c = \frac{volume\ of\ the\ closed\ pores}{volume\ of\ the\ material} \qquad 5.2$$

In order to obtain the value of *vvf$_c$* for a material, the volume of the existing closed pores should be measured while the material is under compression at the test temperature. It is worth mentioning that upon the removal of compressive load and heat, the volume of the existing closed pores will increase, therefore a measured value of *vvf* will not represent the actual value of *vvf$_c$*, as the measured value will be greater than that at compressive load and hot temperature when healing initiates.

Therefore, *vvf$_c$* should ideally be measured while load and temperature are applied. Since porosity is usually concealed within a material, this would require a technique such as in-situ 3D X-ray microscopy. However, currently, it is not possible to distinguish pores in free cutting steel from MnS inclusions, which are much larger than the remnant cylindrical porosity depicted in Figure 5.4c (Kaye et al., 2013). For these reasons, *vvf$_c$* is treated as a fitting parameter to be obtained by matching model predictions to experimental data. This is discussed in greater detail in Chapter 7.
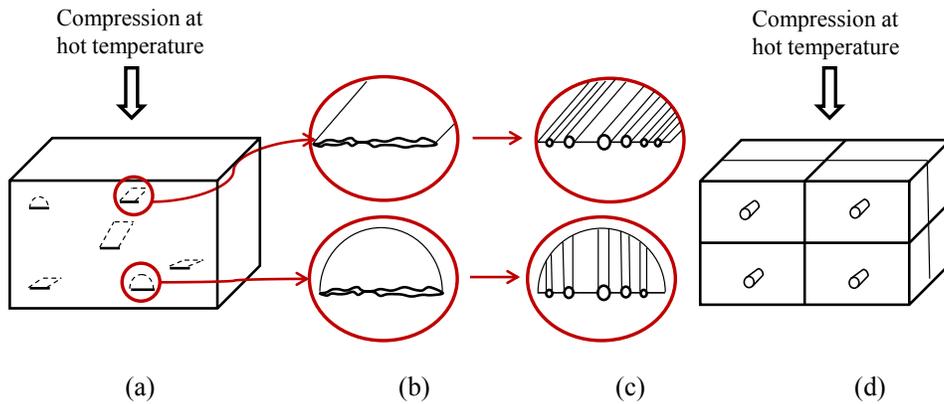
Figure 5.4-*(a) A bulk of porous material, where pores have been closed under compression, (b) A through crack and a penny shaped crack, where the crack surfaces are apart by roughness, (c) Formation of cylindrical voids or channels, (d) Homogenous distribution of cylindrical voids in all elements.*

5.3 The developed User Material (UMAT)

A user material subroutine, UMAT, in ABAQUS/Standard is used to define the mechanical constitutive behaviour of a material. UMAT is used when the existing material models included in the ABAQUS material library cannot accurately represent the behaviour of the material to be modelled. User-defined material models can be used to define any constitutive model of arbitrary complexity.

An understanding of the overall structure of ABAQUS is necessary in developing a user subroutine. Appendix D provides information on the basic flow of data and actions from the start of an ABAQUS/Standard analysis to the end of a step.

Multiple user materials can be implemented in a single UMAT and can be used together. A UMAT subroutine has been developed for the analysis of the material porosity elimination process comprising two stages of void closure and healing. The UMAT consists of three subroutines which are explained below.

5.3.1 Subroutine Gurson-Tvergaard (GT)

The developed approach uses GT under reverse loading to predict void closure. As a result of plastic compression the void volume fraction of the material, *vvf*, reduces to *vvf$_c$*, the void volume fraction at which healing is initiated. Tata steel uses a specific in-house developed UMAT subroutine for the GT constitutive model. This UMAT has been used as a subroutine in the overall void elimination UMAT and contains the GT constitutive equation responsible for the material response.

5.3.2 Subroutine HealTime

The HealTime subroutine is the implementation of the Pilling model. This study applies the model proposed by Pilling (Pilling et al., 1984; Pilling and Ridely, 1987; Pilling, 1988) to predict the healing time. The elimination of voids involves three processes. These processes involve the collapse of the supporting contact area by plastic deformation and creep, diffusion of atoms from grain boundaries to the voids via both volume and interfacial paths and surface diffusion or transfer of mass along the void surface. The model has previously been described in detail in Chapter 3.

To obtain the healing time from the Pilling model, equation 5.4 which is the inverse of Pilling's diffusion bonding rate equation (equation 5.3) (also see Chapter 3 for further details), was integrated using the trapezoidal rule.

$$h = \frac{dvvf}{dt} = -2\dot{\varepsilon}_r(1 - vvf) - N_i\left(\frac{2}{r_o^2 h_o vvf}\frac{D_{gb}\delta\Omega\sigma_z}{KT}\frac{1 - vvf}{\ln(1/vvf) - (1 - vvf)/2}\right) \qquad 5.3$$

$$g = \frac{dt}{dvvf} = [h]^{-1} \qquad 5.4$$

In equation 5.3, σ_z is the axial stress and $\dot{\varepsilon}_r$ is the radial component of the strain rate; equations for σ_z and $\dot{\varepsilon}_r$ can be found in (Pilling, 1988) and in the HealTime subroutine shown in Appendix E. These variables are calculated for each element

based on its stress state and are passed from ABAQUS into the UMAT every time the UMAT is called.

In equation 5.3, N_i is the number of interfaces intersecting the void surface ($N_i = \text{int}(1 + h_o vvf/d)$, where d is the grain size), D_{gb} is the grain boundary diffusion coefficient, δ is the grain boundary thickness, Ω is the atomic volume, K is the Boltzmann's constant, σ_z is the axial stress and T is the absolute temperature. More details on Pilling model can be found in Chapter 3.

The trapezoidal integration method approximates an integral from a to b by approximating the area under a curve using trapezoids. This is shown mathematically in equation 5.5, where N is the number of intervals.

$$t = \int_{a=vvf_c}^{b=0} g \, dvvf$$

$$\approx \frac{\Delta vvf}{2}[g(vvf_0) + 2g(vvf_1) + 2g(vvf_2) + \cdots + 2g(vvf_{N-1}) + g(vvf_N)] \qquad 5.5$$

$$\Delta vvf = \frac{b-a}{N}, t_i = a + i\Delta vvf$$

The HealTime subroutine calculates the total healing time based on the above equations (see Appendix E for the developed Fortran code).

5.3.3 Subroutine FindF

During the void closure process the change in *vvf* is calculated from the Gurson-Tvergaard (GT) subroutine. However, after the healing process starts the change in *vvf, Δvvf,* at each time increment is calculated from the Pilling equation (equation 5.3). This is especially important in multi-pass rolling processes when the final closure and healing states of the current roll pass are inherited as the starting condition of the next roll pass. For example, one element may reach the closure state in the first roll pass but it may not completely heal until the fourth or fifth roll pass.

Therefore, the variation in *vvf* from the first to the last roll pass is important and is calculated from the Pilling model here.

Therefore, the behaviour of the material is updated during the healing process. When healing is completed, *vvf* is set to zero, reducing the GT yield surface to a von Mises yield surface.

The change in *vvf* at each time increment, *Δt*, is defined as:

$$\Delta vvf = h \cdot \Delta t$$

5.6

A subroutine has been developed which calculates the change in *vvf* at each time increment for this purpose (see Appendix E for the developed Fortran code).

5.3.4 Methodology and illustration

This section explains how the subroutines explained above are combined into a UMAT subroutine for prediction of void elimination process.

In a general loading condition, voids in different locations may experience different stress states and will close at different times and under different local pressures. Therefore, at each time increment, the characteristics of each element must be monitored. This requires the definition of an array, *I*, storing the stress conditions, healing process and element status. This array is specific to Pilling model here but it can be adapted to any other diffusion bonding/healing model. Defining *I* as,

$$I_{k_n}^{i,j} = \left[S^{i,j}, t_c^{i,j}, t_h^{i,j}, P_{k_n}^{i,j} \right]$$

5.7

in which i is the element number, j is the integration point and k_n is the increment number. For elements with more than one integration point, an average value could be calculated. This study however uses reduced integrated elements with one integration point (i.e. j=1). From now on as a matter of simplicity j notation is ignored.

In equation 5.7, t_c is the time at which the *vvf* reaches vvf_c, P is the compressive hydrostatic pressure under which this condition occurs, t_h is the time required for healing and S represents the status of the element which is defined as follows,

$$S^i = \begin{cases} 2 & \text{if} & t_{total}^i \geq t_c^i + t_h^i \\ 1 & \text{if} & t_c^i \leq t_{total}^i < t_c^i + t_h^i \\ 0 & \text{if} & \text{otherwise} \end{cases} \qquad 5.8$$

i.e. $S = 1$ when *vvf* is reduced to vvf_c (void is closed but not healed), and $S = 2$ when the element is healed.

Before getting into the details of the implementation of the above concept into a UMAT subroutine, it is better to have a brief introduction on how UMAT subroutines are handled in ABAQUS. More details on how a UMAT is run in ABAQUS can be found in Appendix D.

Figure 5.5 shows a flow chart illustrating the step by step procedure followed by the UMAT for void closure and healing. It is important to know that a UMAT is run for each time increment, for each element integration point and for each iteration (ABAQUS document).

Initially the status of all elements assigned with UMAT properties is set to zero, indicating that these elements contain open voids with a specified void volume fraction. The GT model is the constitutive equation responsible for the material response. It is therefore called to calculate the stress and strain status in the model as well as the change in the void volume fraction of the elements. Void closure can be predicted by calling the GT subroutine. During deformation the void volume fraction of each element is monitored. Once an element's void volume fraction reaches vvf_c, the following actions should be taken:

1) The element status, *S*, will change from 0 to 1, indicating the element has reached the closure state.

2) HealTime subroutine is called to calculate the healing time, t_h^i, (using Pilling model, explained in Chapter 3) for that element.

125

3) FindF subroutine is called to obtain the change in *vvf, Δvvf,* during healing at each time increment according to equation 5.6. This implies that after the closure stage the change in the void volume fraction is controlled by the Pilling model, but the GT model inherits the new value of *vvf.*

Step 3 will continue until the healing time is reached ($t \geq t_c^i + t_h^i$). At this point the element is healed (*vvf* = 0), reducing the GT yield surface to the von Mises yield surface. The element's status should therefore be changed from 1 to 2 (from closed to healed).
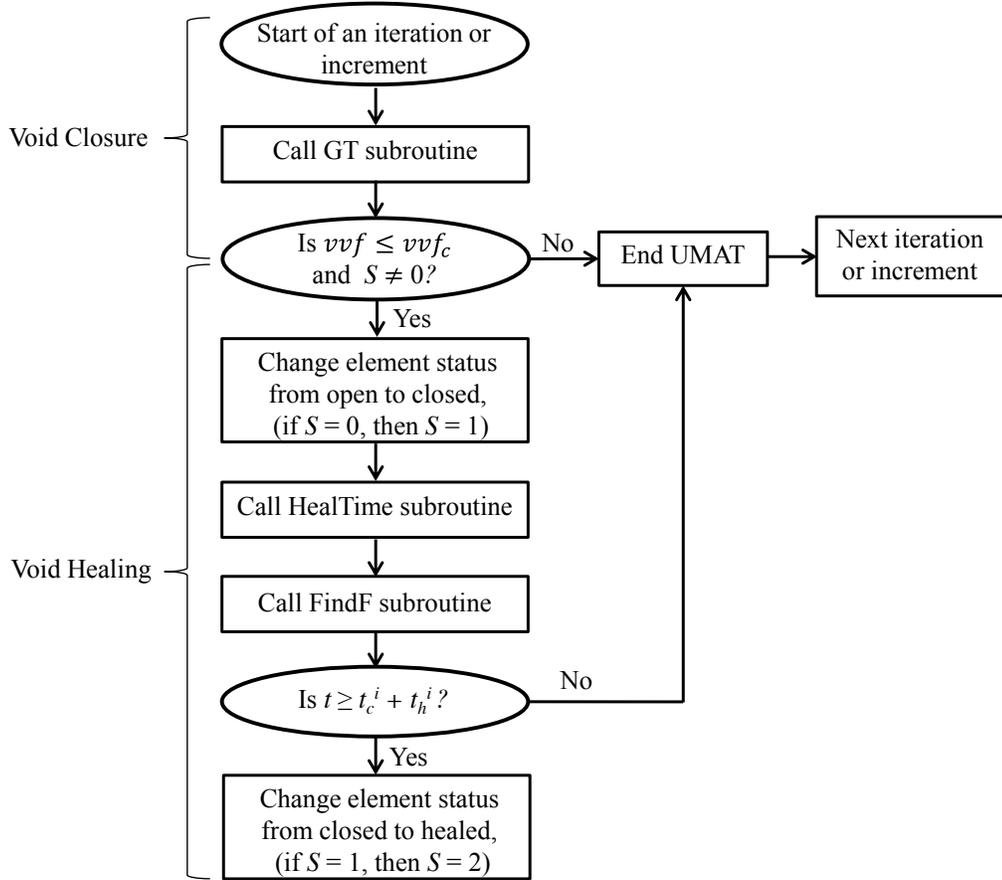
Figure 5.5- *The developed UMAT structure, where an oval signifies a decision point in the code or a specific state (i.e. beginning of an increment) during the analysis and a rectangle signifies an action that is taken during the analysis. More details on how a UMAT is run in ABAQUS can be found in Appendix D.*

126

5.4 The problem of modelling long real times

Complete healing might not occur in some cases when the stress state is non-homogeneous or the simulation time is not large enough for the healing to complete. In principal, when vvf reaches vvf_c, healing should begin and have a duration t_h as predicted by the healing model (equation 5.4) for the local stress. However, since the healing times are very long, the problem of simulating the real time emerges. A time scaling method is proposed to deal with this problem. Consider a model with two elements, where the closure and healing times are t_c^1, t_h^1 and t_c^2, t_h^2 for elements 1 and 2, respectively. The desired simulation time, t_s, is much smaller than the healing times t_h^1 and t_h^2. It should be noted that the closure time for an element, t_c, is very small compared to the corresponding healing time, t_h; t_c is reached as a result of plastic deformation whereas t_h involves slower processes such as creep and diffusion bonding.

Figure 5.6a shows a situation where the closure state for all elements is reached whithin the desired simulation time, t_s ($t_c^1 < t_s$ and $t_c^2 < t_s$). Whereas the healing times t_h^1 and t_h^2 are greater than t_s and therefore healing will not occur before the end of the simulation. One solution is to assume the elements are healed as soon as they are closed. This is illustrated in Figure 5.6b. However, using this approach neglects the effect of the change in material behaviour that would result from sequential healing. This could be important in many applications, e.g. multi-pass rolling where the final closure and healing states of the current roll pass are inherited as the starting condition of the next roll pass. In this application it is necessary to account for the effects of sequential and incomplete healing on the mechanical properties of the material.

A better solution would be to scale the healing time remaining for a given element, measured relative to the time at which the last element closure occurs, $t_h^i - \left(\max(t_c^i) - t_c^i\right)$, using a scaling factor, η_s. The scaling factor, η_s and the simualtion

time t_s should be selected so that the void elimination of all elements occurs within the simulation time and the element void elimination sequence is preserved (see Figure 5.6c). Considering the fact that $t_c \ll t_h$, it can be assumed that the closure state for all elements is reached before the first element's void elimination would occur in real time. Although the possibility exists of void elimination (i.e. reaching the healing time) in some elements causing a change in the stress state of neighbouring elements, and consequently leading to closure of those elements, this is a small effect and is neglected. Based on the above assumptions the following statements can be made:

The scaled void elimination time, t_{es}^i (t_e^i is the real elimination time), for all elements is less than the simulation time:

$$max\left(t_{es}^i\right) < t_s \qquad\qquad 5.9$$

No scaled elimination time is reached before all elements are closed:

$$min\left(t_{es}^i\right) > max\left(t_c^j\right) \qquad\qquad 5.10$$

where *i* and *j* represent the element number (*i* is not necessarily equal to *j*).

Given that $t_h^i = t_e^i - t_c^i$, it follows that the healing time remaining $t_h^i - \left(max\left(t_c^i\right) - t_c^i\right) = t_e^i - max(t_c^i)$. Defining the scaled elimination time as $t_{es}^i = max\left(t_c^i\right) + \eta_s \cdot [t_e^i - max\left(t_c^i\right)]$ would automatically satisfy equation 5.7 for all values of $\eta_s > 0$. Equation 5.6 can be restated as equation 5.8 and can be used to determine an upper bound for η_s:

$$max\left(t_c^i\right) + \eta_s[max\left(t_e^i\right) - max\left(t_c^i\right)] < t_s \qquad\qquad 5.11$$

or

$$0 < \eta_s < \frac{t_s - max\left(t_c^i\right)}{max\left(t_e^i\right) - max\left(t_c^i\right)} \qquad\qquad 5.12$$

$$t_c^1 \quad t_c^2 \qquad\qquad t_s \; t_h^1 \; t_h^2 \qquad \text{(a)}$$

$$t_c^1, t_h^1 \quad t_c^2, t_h^2 \qquad\qquad t_s \qquad \text{(b)}$$

$$t_c^1 \qquad t_c^2 \; t_h^1 \; t_h^2 \qquad\qquad t_s \qquad \text{(c)}$$
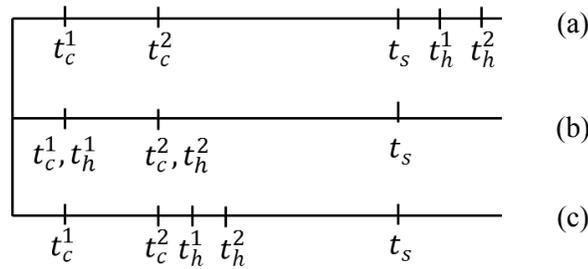
Figure 5.6*- Scaling of healing time to overcome the problem of simulating long real times.*

For most applications it is advantageous to choose η_s as small as possible, hence this demonstrates that scaling only the part of the healing time that would elapse after the final element closure, with $\eta_s = 0$, preserves the sequence of void closure and healing and takes into account the effect healing might have on the closure process of other voids. In other words, with $\eta_s = 0$, all healing can be deemed complete once the last closure is complete. To observe the sequential healing, $\eta_s > 0$ is required.

5.5 Comparison of GT model with void elimination model

This section compares the material behaviour predicted by GT model and the developed void elimination model. This will illustrate the change in the porous material behaviour when healing occurs.

5.5.1 Finite element model

A three-dimensional model of a cylinder with a central porous region was constructed in ABAQUS/Standard, version 6.10-2, a schematic of which is shown in Figure 5.7. The initial void volume fraction was taken to be *vvf* = 0.005, and the aforementioned material properties for free cutting steel (FCS) at 900 °C and $\dot{\varepsilon} = 0.1$ (Foster, 2007) (see Appendix A) were used. The calibrated fitting parameters, $q_1 = 2.0$, $q_2 = 1.25$ and $q_3 = 2.0$ were obtained for an average triaxiality, $T = -1/3$ (see

Chapter 4, Table 4.4). The details of the FE model are explained in the following subsections. To account for large deformations the finite strain/deformation theory was used in the simulation (NLGEOM was set to ON in ABAQUS).

Boundary conditions:

An eighth of a cylinder was modelled with symmetry boundary conditions on the flat faces as shown in Figure 5.7. In this figure, plane 'abcd' is symmetric relative to the z axis ($U_z = UR_x = UR_y = 0$, where U and UR represent displacement and rotation, respectively). Similarly, plane 'dcfe' is symmetric relative to the x axis ($U_x = UR_y = UR_z = 0$) and plane 'bcf' is symmetric relative to the y axis ($U_y = UR_x = UR_z = 0$). The die is a rigid surface in contact with the top of the cylinder and is only free to move in the y direction ($U_x = U_z = UR_x = UR_y = UR_z = 0$).

In order to avoid separation between the die and the cylinder in the tension step, the die was tied to the top surface of the cylinder (plane 'ade'). A frictionless surface to surface type of contact was defined between the die and the cylinder. 'Hard' type of contact (default assumption in surface to surface contact in ABAQUS) was used in defining the interaction in the normal direction. Thermal interactions between the die and the sample were ignored in this simulation.

Loading conditions:

The simulation consists of three steps of compression, unloading and tension. During the compression step, pressure is evenly applied to the die increasing linearly from 0 to 60 MPa. Due to the compression, void closure occurs in this step. In this step, the void volume fraction of the elements in the porous region specified in Figure 5.7, reaches vvf_c. The pressure is then held constant at 60 MPa for the healing time to be reached ($vvf = 0$) for all the elements. The cylinder is then pulled in tension to investigate the change in vvf, once healing was complete. To do so, an upwards

displacement of 1.75 mm in the *y* direction was applied to the reference point (RP) of the rigid die.
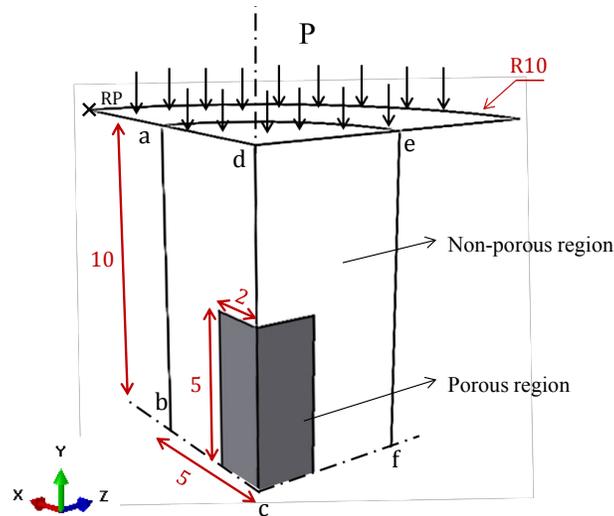
Figure 5.7*- A cylindrical model with central porous region (all dimensions are in mm).*

Mesh and element type:

The one eighth cylinder consists of a total of 1660 elements with 290 elements in the porous region. A mesh with 8-node linear brick, reduced integration elements (C3D8R in ABAQUS terminology) was used. Figure 5.8 represents the mesh configuration. The die was defined as a discrete rigid surface consisting of 98 elements. A quadratic-dominated free mesh with discrete rigid elements was applied to the die.
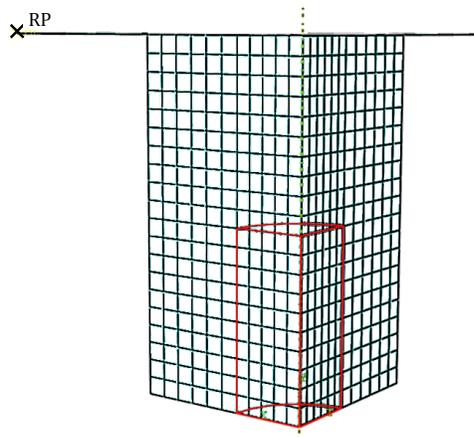
Figure 5.8*- The mesh configuration.*

131

5.5.2 Simulation procedure and results

The porous region of the model shown in Figure 5.7 was given GT material model in one case and the developed void elimination model (closure + healing) in another case. Figure 5.9 compares the change in *vvf* of the porous region for the sequence of compression, holding the compressive load and tension for the developed void elimination model compared to an equivalent GT material without void healing. In both cases the cylinder is compressed where the void volume fraction of the elements in the porous region, reaches *vvf*$_c$, this is represented as the closure stage in Figure 5.9. The compressive load is then held for healing to occur. For the GT material the *vvf* remained constant during this step, whereas for the void elimination material model the *vvf* decreases to zero as a result of healing.
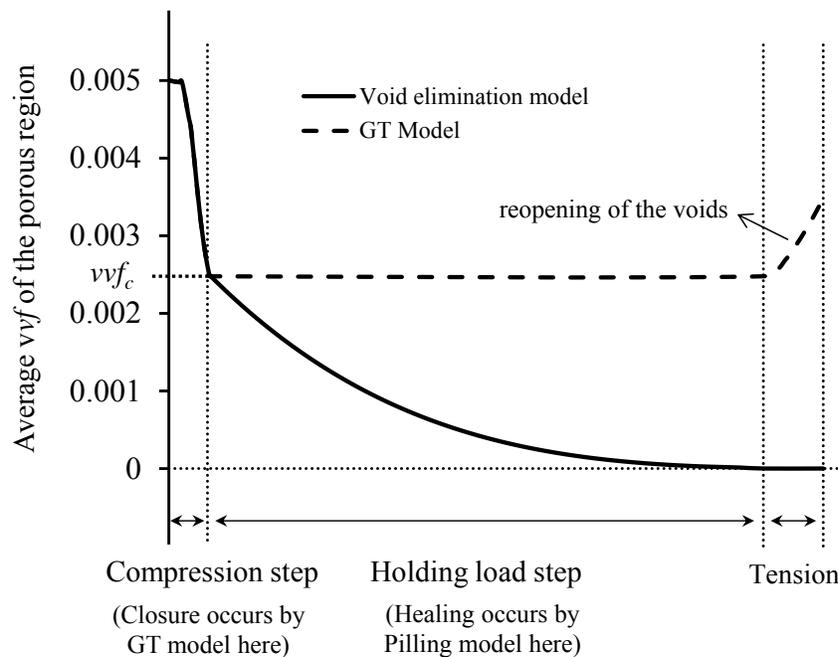
Figure 5.9- *The comparison of the GT model and the developed void elimination model.*

To investigates the change in material behaviour as a result of healing, the load versus displacement plots obtained from the GT material model and the void

elimination model had to be compared. However, before doing so the FE model had to be slightly modified. In order to better reveal the material behaviour difference as a result of healing, a larger initial void volume fraction (*vvf* = 0.02) and a larger porous zone (see Figure 5.10) were considered and a greater displacement was applied in the tension load step. This was initially attempted with the model shown in Figure 5.7. However, there were convergence problems for this geometry; so the dog-bone shape shown in Figure 5.10 was used. Boundary conditions, contact properties and element type used for this model are identical to the previous model shown in Figure 5.7.
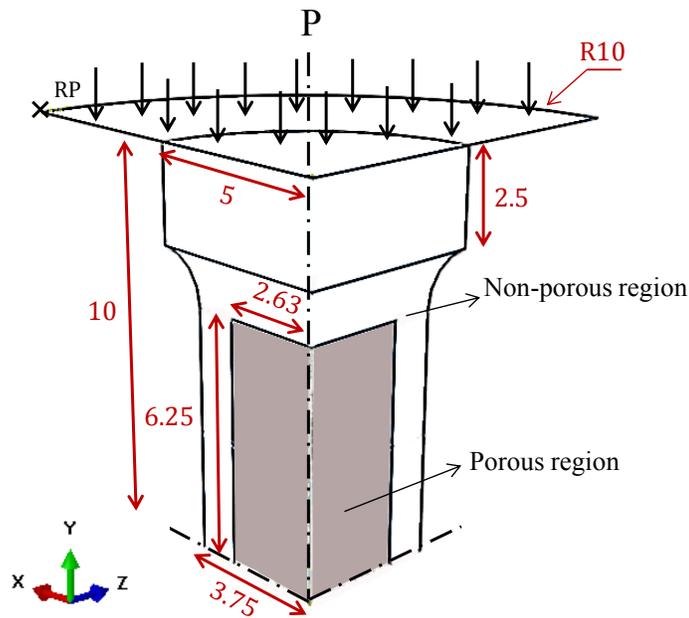
Figure 5.10- *The dog-bone model with central porous region (all dimensions are in mm).*

As in the previous model, the simulation consists of three steps of compression, holding the compressive load and tension. Application of the same pressure level (*P* = 60 MPa) as in the original model resulted in total collapse of the cylinder. This is due to the higher compliance resulting from the larger porous region, larger initial *vvf* and the dog-bone shape. Therefore, a smaller compressive load (*P* = 16 MPa) was applied in this study compared to the previous model (Figure 5.7). The load increased linearly from 0 to 16 MPa during the compression step and was evenly

applied to the die. This is illustrated in Figure 5.11 from point A to B, where a compressive load of 1.25 kN (P = 16 MPa) has resulted in a displacement of -2.27 mm. This displacement is the movement of the reference point (RP) specified on the die (see Figure 5.10). The compressive load is then held constant for healing to occur (point B in Figure 5.11).

The compressive load is then removed (point B to C in Figure 5.11) and the cylinder is pulled in tension (point C to D in Figure 5.11) where an upwards displacement of 1.27 mm in the *y* direction was applied to the RP on the die. Points D to E show the plastic behaviour according to the GT model with healing enabled (according to Pilling model) in tension. Points D to F show the plastic behaviour according to the GT model without healing enabled in tension. It is evident from this figure that the GT model with healing enabled requires a higher load to reach the same displacement as the corresponding GT model without healing enabled in tension, as a result of voids reopening under tension in the latter.
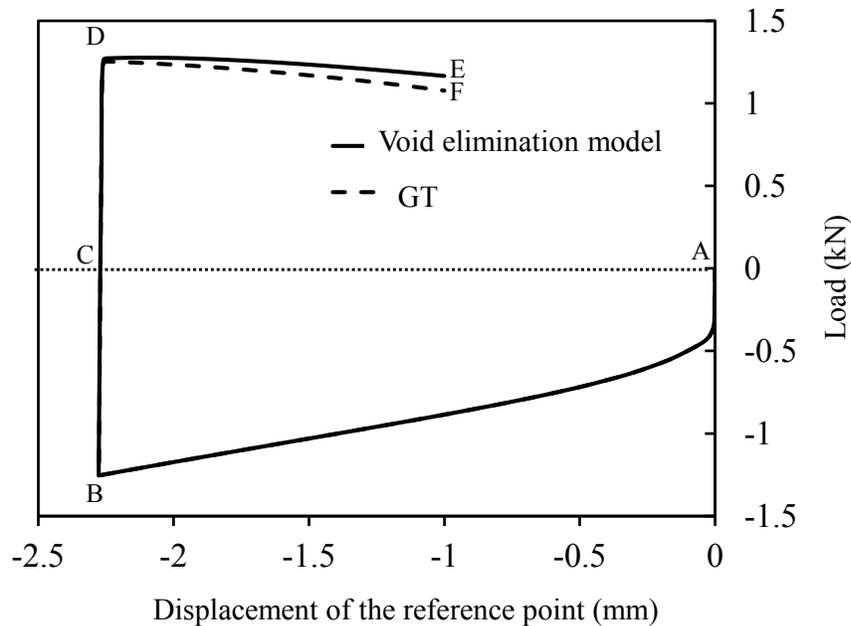
Figure 5.11 *- Comparison of load-displacement behaviour of GT and the developed void elimination model (the load here is the summation of the reaction forces of the bottom surface of the one eighth cylinder in Figure 5.10).*

134

5.5.3 The effect of change in stress state

Depending on the application, the stress state might change during the loading process. For example, in rolling applications the stress state at some locations changes from compression to tension. This will result in opening of the closed but not healed voids in the elements. Therefore, the void elimination model has to be able to capture this situation. This section investigates the capability of the developed void elimination model in correctly predicting *vvf* in situations when the stress state changes.

During the healing process, the stress state of those elements which have already been closed might change from compression to tension. In that case since the elements have not been completely healed reopening of those elements might occur. To check this situation the change in vvf is monitored in each time increment. A positive change in the void volume fraction ($\Delta vvf > 0$) implies that the element is under tension and the void is growing. In that case if the vvf of that element exceeds the vvf_c, the element's status has to be changed from closed to open.

A similar model to the one used in the previous section (Figure 5.10) was used to investigate this situation. However, the tension stage was started earlier than before to ensure some closed but not healed voids still exist in the model.

To illustrate this, the porous region of the model in Figure 5.10 is considered. The reopening of voids for this porous region has been illustrated in Figure 5.12. It can be seen from Figure 5.12a that under compression, part of elements have reached the closure state (light blue elements) whereas some others haven't (dark blue elements). As the healing time is reached, the healing starts for some elements (green elements in Figure 5.12b). The holding time however is not long enough for some other elements and therefore the healing will not occur for all elements with closure state (light blue elements in Figure 5.12c). The cylinder is stretched in the next stage and therefore elements with closed status have reopened (red elements in Figure 5.12d).

Therefore, the model successfully simulates the situation of changing the stress state.
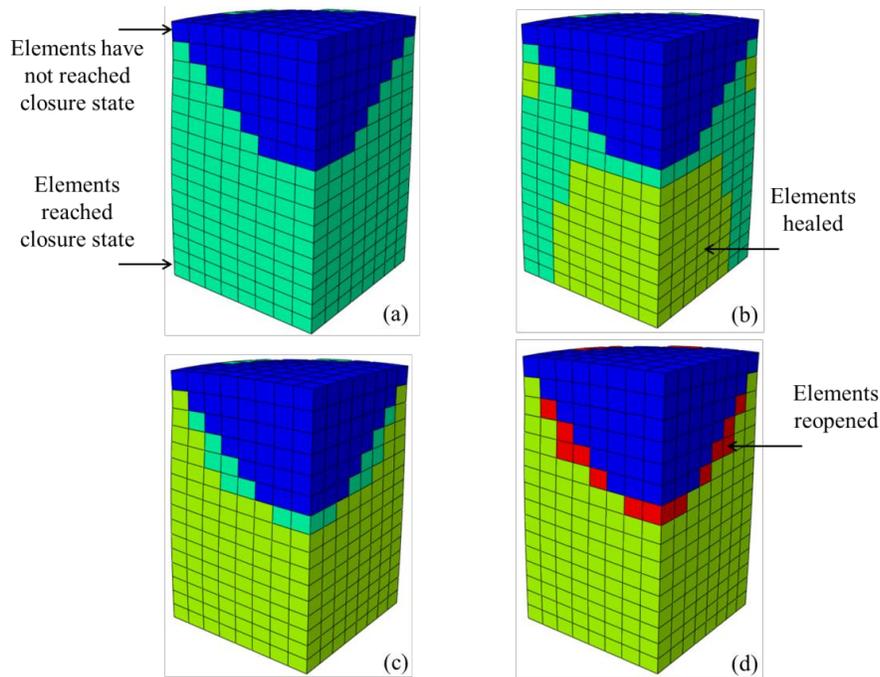
Figure 5.12*- (a) Some elements have reached the closure state. (b) Healing has started from the centre (c) Majority of elements have been healed however a small number of elements are still in closure state and some elements have not even reached the closure state yet. (d) The cylinder has been stretched and the elements with closed status have reopened.*

5.6 Healing under non-uniform stress states

In the previous section (section 5.5) the stress state was almost uniform. Non-uniform stress states are very common in real cases such as rolling and indentation. During rolling the stress state varies at different locations of the bloom/billet. It is therefore important to study the applicability of the void elimination model to non-uniform stress states. For this purpose a three-dimensional work-piece loaded by a rigid roll was created in ABAQUS/Standard under an indentation scenario (no rolling), a schematic of which is shown in Figure 5.13. The reason for selecting such an FE configuration was to create a stress state similar to that in the rolling process in

terms of non-uniformity. However, the FE model created in this section does not simulate the exact rolling process and is more similar to an indentation process where the work piece is stationary (in contrast to a rolling process where the bloom moves horizontally through the rollers) and the roll moves downwards in the *y* direction as shown in Figure 5.13 (in contrast to the rolling process where the rolls rotate). The rolling process will be simulated in Chapter 6.

5.6.1 Finite element model

A three-dimensional work-piece loaded by a rigid roll was created in ABAQUS/Standard. The work-piece was divided into two sections, section I and section II as shown in Figure 5.13. The developed void elimination UMAT subroutine was applied only to section I of the work-piece in contact with the rigid roll. The outer section of the work-piece (section II), where appreciable densification and healing would not occur, was modelled using a von Mises yield surface equivalent to the GT model with *vvf* = 0.

The initial void volume fraction was taken to be *vvf* = 0.005 and the aforementioned material properties for free cutting steel (FCS) at 900 °C and $\dot{\varepsilon} = 0.1$ (Foster, 2007) (see Appendix A) were used. The calibrated fitting parameters, q_1 = 2.0, q_2 = 1.25 and q_3 = 4.0 were obtained for an average triaxiality, *T* = -0.5 (see Chapter 4, Table 4.4).

To account for large deformations the finite strain/deformation theory was used in the simulation (NLGEOM was set to ON in ABAQUS).

Boundary conditions:

In Figure 5.13, plane 'A' is symmetric relative to the *x* axis ($U_x = UR_y = UR_z = 0$, where *U* and *UR* represent displacement and rotation, respectively). Plane 'B' is symmetric relative to the *y* axis ($U_y = UR_z = UR_x = 0$).

The roll (compactor) in Figure 5.13 is a 3D discrete rigid body and is in contact (surface-to-surface contact) with the top of the work-piece. The roll is only free to move in the *y* direction (other degrees of freedom for the roll have been constrained).

A 'tangential behaviour' for the contact surfaces was defined using the penalty method. The friction coefficient between the roll and the steel work-piece was taken to be 0.8 which represents the steel to steel friction coefficient for non-lubricated surfaces (Engineering Toolbox, 2013). Unless otherwise stated, the default values in ABAQUS/Standard for surface to surface contact were used for all other parameters.

Loading conditions:

A compressive force was applied to the reference point of the roll increasing linearly from 0 to 3270 kN over 6 seconds during the compression step. The load was then held constant while densification and healing occurred.

Mesh and element type:

The work-piece consists of 10500 elements with 3000 elements in the porous region (Section I in Figure 5.13). A mesh with 8-node linear brick, reduced integration elements (C3D8R in ABAQUS terminology) was used. Figure 5.14 represents the applied mesh.

The rigid roll consisted of 1080 elements. A quadratic-dominated free mesh with 4-node 3D bilinear rigid quadrilateral elements (R3D4 in ABAQUS terminology) was applied to the roll.
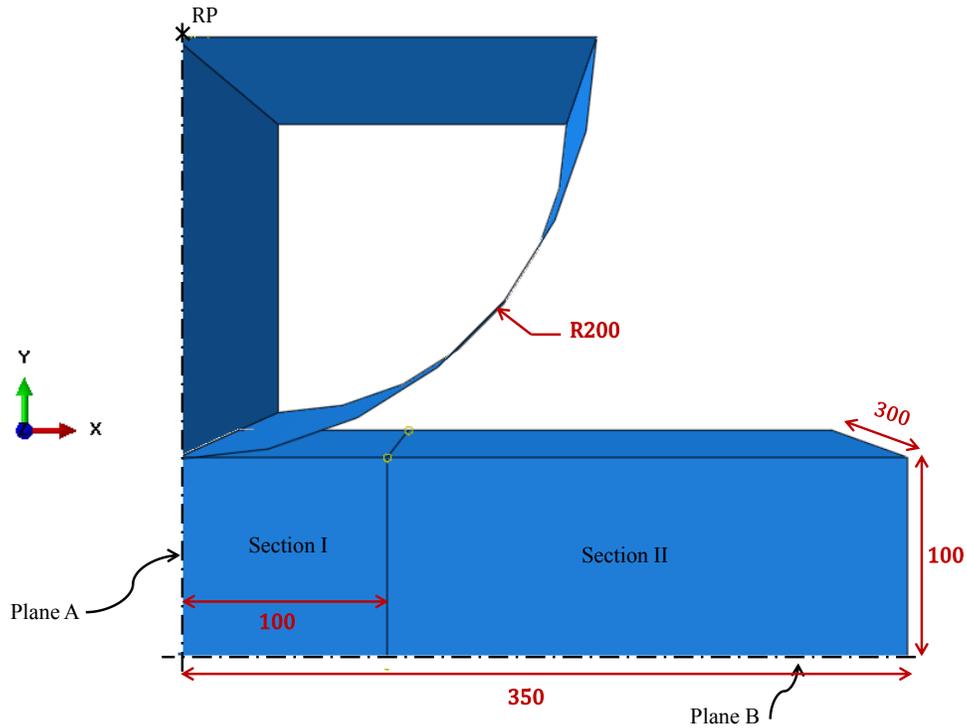
Figure 5.13*- The schematic of a 3D work-piece with porous middle section under compression.*

5.6.2 Simulation results

As it was mentioned in section 5.6.1, the UMAT subroutine was used only in the centre part of the sheet in contact with the rigid roll. The outer section of the sheet, where appreciable densification and healing would not occur, was modelled using a von Mises yield surface equivalent to the GT model with $vvf = 0$. Figure 5.14 shows the stress distribution in the specimen under compression; the middle section is under plane strain but approaches a plane stress condition nearer to the surface.
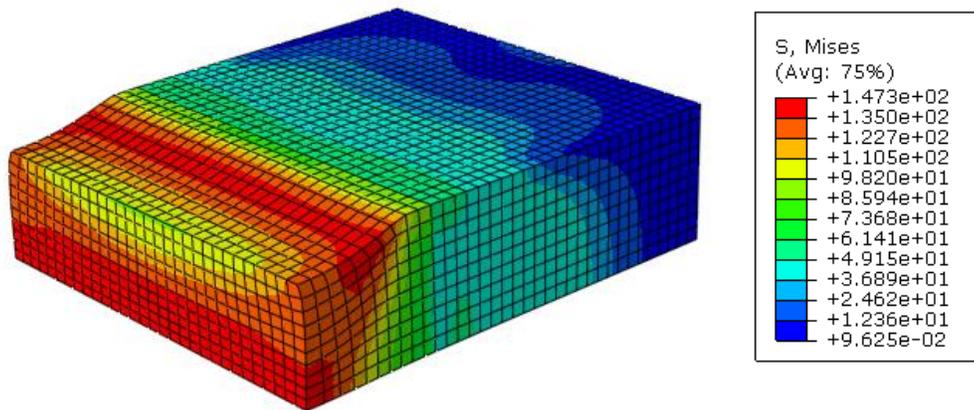
Figure 5.14- The stress distribution in the specimen under compression.

Figure 5.15 shows the healing process of the specimen during the application of the compressive load. When the compressive load is sufficient some elements reach the closure state. The rate of reduction of void volume fraction in the centre of the sheet is higher because the stress in the loading direction is highest there, hence closure occurs more readily in the centre (Figure 5.15a). Figure 5.15b shows more elements have reached the closure state as the compressive load has reached its maximum value (as mentioned in section 5.6.1 load is applied linearly over a period of 6 seconds). Nearer to the free surfaces of the specimen the rate of reduction of void volume fraction reduces, therefore fewer elements reach the closure state near the free surfaces. It is evident that the healing will start from elements which reached the closure state first (Figure 5.15c). It can be seen from Figure 5.15d that a larger number of elements have been healed after 60 seconds, however, some have remained unhealed and will be healed upon holding the load for a longer time.

(a) time = 1.6 s

Element status

+2.0e+00
+1.0e+00
+9.9e-01
+0.0e+00

(b) time = 6 s

(c) time = 22.1 s

(d) time = 60 s
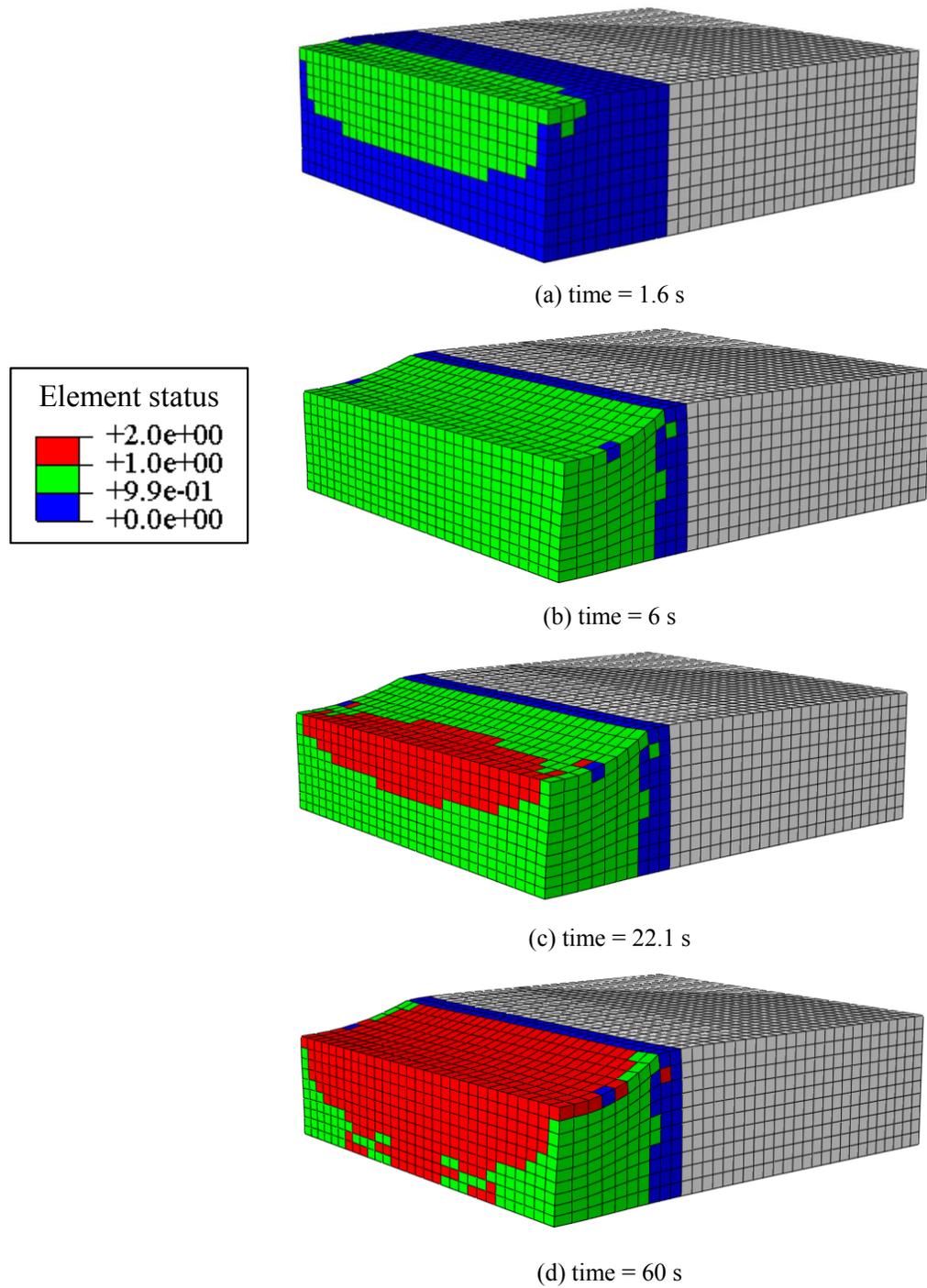
Figure 5.15*- Progress of healing during loading (blue, green and red represent the unclosed, closed and healed elements respectively). (a) closure has started, (b) more elements have reached the closure state, (c) healing has started, (d) healing has progressed with time.*

141

5.6.3 Sensitivity of healing time to vvf_c

The vvf_c strongly affects the healing time: larger values of vvf_c imply coarser void surfaces and therefore longer healing times. To investigate this, the model explained in section 5.6.1 was considered. The simulation was performed for two different *vvf$_c$* values of 0.0045 and 0.0035. The change in *vvf* with time during the healing process was obtained for all elements. To simplify the presentation of the results, the *vvf* versus healing time is only shown for a random element (element A as shown in Figure 5.16) in both cases. The results (*vvf* versus healing time) are presented in Figure 5.17a and b, where for a vvf_c of 0.0045 (Figure 5.17a) the healing time is longer than for $vvf_c = 0.0035$ (Figure 5.17b). It is worth noting that the healing times are long compared to a typical roll pass. This is discussed further in Chapter 6 where rolling is simulated.
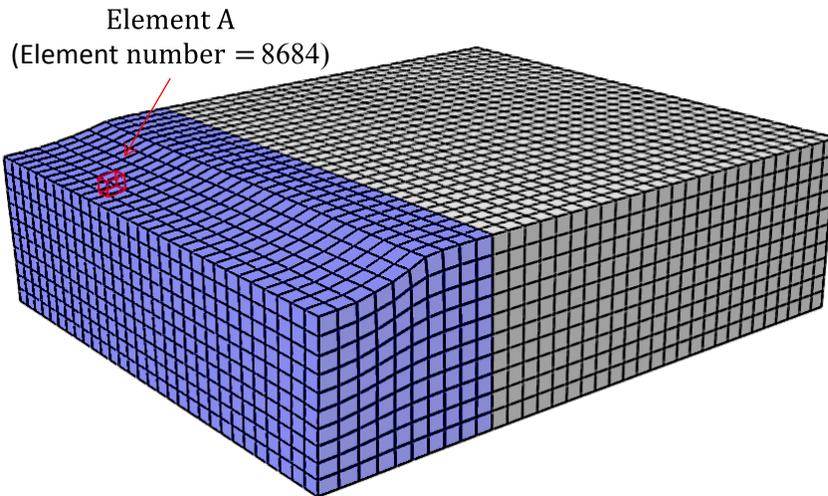
Figure 5.16- The selected element (Element A) used for presentation of healing time sensitivity to vvf$_c$.
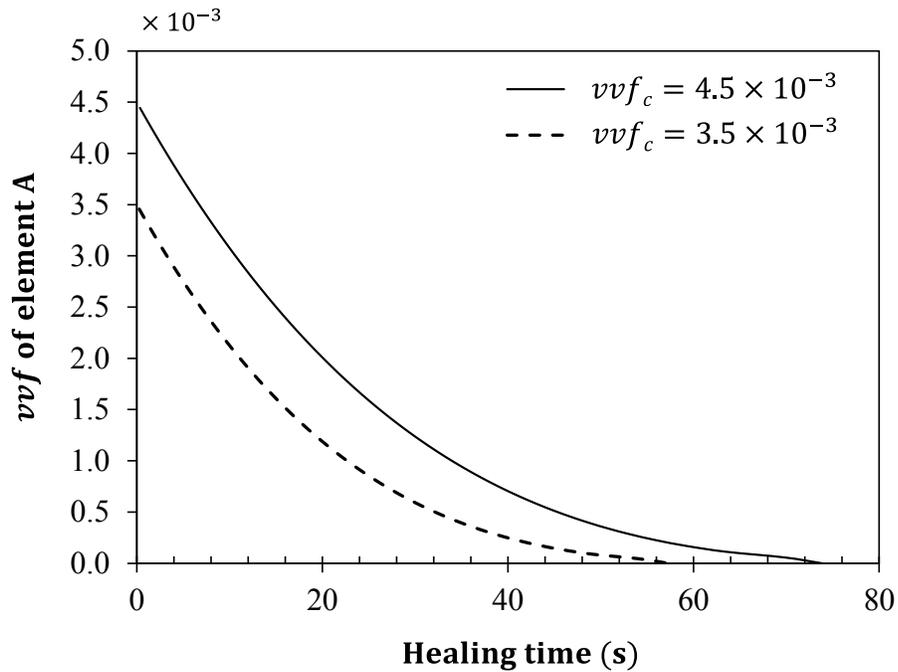
Figure 5.17 *- Comparison of vvf versus healing time for when vvf$_c$=0.0045 and when*

vvf$_c$=0.0035.

5.7 Conclusions

An Abaqus/UMAT subroutine has been developed for the analysis of the material porosity elimination process including two stages of void closure and healing. The model uses GT under reverse loading to predict void closure, where as a result of compression the void volume fraction (vvf) of the material reduces to vvf_c. Further application of the Gurson model will not bond the surfaces of the pores and therefore vvf is not practically zero and thus a healing model is required. Pilling's healing model has been implemented into the UMAT to predict the healing time here but it can be adapted to any other diffusion bonding/healing model.

The simulations were performed on a specimen with porous material properties (GT model) and on the same specimen with the current healing UMAT model. The

comparison of the results showed increased strength in the latter model as a result of porosity elimination.

The healing model is significantly affected by vvf_c. Larger values of vvf_c, imply coarser void surfaces and hence longer healing times. Therefore, for two different vvf_c values, the difference in closure times is insignificant due to plastic deformation whereas the healing times vary noticeably. This is especially important during rolling processes where a correct healing time needs to be achieved by adjusting the rolling speed.

5.8 References

Engineering Toolbox (2013). Available from World Wide Web :
    http://www.engineeringtoolbox.com/friction-coefficients-d_778.html

Farrugia, D. (2012). TATA Steel technical presentation, TATA Steel.

Foster, A. (2007). Birmingham, University of Birmingham. Ph.D thesis.

Kaye, M., Puncreobutr, C., Lee, P. D., Balint, D. S., Connolley, T., Farrugia, D., and
    Lin, J. (2013) Acta Materialia, 10.1016/j.actamat.2013.08.065 (Accepted
    forpublication).

Pilling, J. (1988). "The kinetics of isostatic diffusion bonding in superplastic
    materials." Materials Science and Engineering **100**: 137-144.

Pilling, J., D. W. Liversey, J. B. Hawkyard and N. Ridely (1984). "Solid state
    bonding in superplastic Ti-6AI-4V." Metal Science **18**(3): 117-122.

Pilling, J. and N. Ridely (1987). "Solid state bonding of superplastic AA 7475."
    Materials Science and Technology **3**(5): 353-359.

Abaqus document, *Writing User Subroutines with ABAQUS* [Online]. Available:
    http://imechanica.org/files/Writing%20User%20Subroutines%20with%20AB
    AQUS.pdf.

Chapter 6

Model application to the rolling process

In this chapter the developed UMAT will be applied to model the void elimination of a Free Cutting Steel (FCS) work-piece during rolling. The effect of parameters such as the Roll Gap geometry/shape Factor (RGF) and initial porosity distribution on the void elimination process will be investigated.

6.1 Rolling process

In rolling the cross-sectional area of a work-piece is reduced or changed in shape by the compressive forces exerted by the rotating rolls. In relation to temperature, rolling is classified into three groups: cold rolling which occurs at temperatures less than 0.3 of the melting temperature, T_m, warm rolling which involves temperatures between $0.3T_m$ - $0.7T_m$ and hot rolling which occurs above the recrystallization temperature (above $0.7T_m$).

Hot rolling is often used for as-cast steels. The work-piece starts as an ingot which is a metal stock cast into a size and shape convenient to store and transport. The ingots have different size and weight ranging from small blocks weighing a few kilograms to large ingots weighing several tons. Prior to entering the roll-reduction sequence the ingot is heated in a furnace for a few hours to reach a uniform temperature profile. After removing the scale (oxide forming at high temperatures) from the work-piece surface, it enters a "cogging/blooming mill" where it reduces into a bloom or slab (see Appendix F, part i for information on products of the rolling mill). The largest thickness reduction in the work-piece occurs in a "roughing mill/train". There are two different types of roughing processes. One consists of several rolling stands (typically four to five set of rollers) which reduce the thickness of the work-piece. The other type consists of one single reversing stand through which the work-piece is moved back and forth. The thickness of the work-piece is further reduced in a "finishing train" consisting of four to seven stands, where the desired surface finish is obtained. Figure 6.1 illustrates the hot rolling process of steel.

The product of rolling will be a high quality material with reduced porosity, enhanced ductility and strength and more uniform and finer grain size. Figure 6.2 shows the changes in the grain structure of cast metal during hot rolling (Kalpakjian and Schmid 2001)
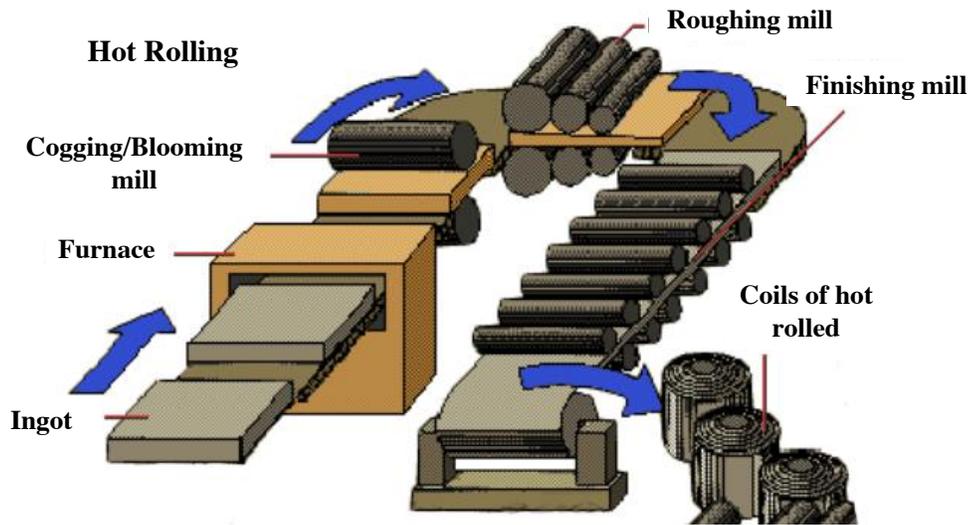
Figure 6.1- Rolling process (Udomphol 2007).
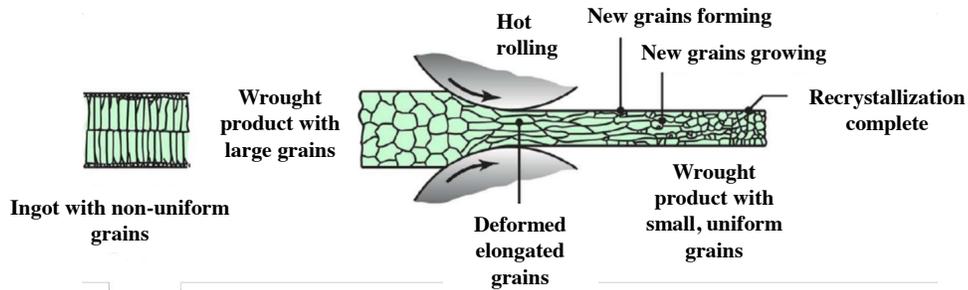
***Figure 6.2**- Change in the grain size during the hot rolling process (Kalpakjian and Schmid 2001).*

Depending on the desired output product, different rolling processes are used. These include flat rolling, shape rolling, tube rolling, plug rolling, ring rolling, etc. Flat rolling involves slabs, plates, sheets, strips and foils. This study focuses on flat rolling process only.

Since rolling is a widely used deformation process it has different versions. The sequence of the stands in the rolling process and also the arrangement of the rolls in the mill stand can differ for any rolling system (Appendix F, part ii summarises different rolling mill configurations).

6.1.1 Parameters in the rolling process

Figure 6.3 illustrates the geometries in flat rolling, where the starting and ending materials have rectangular cross sections. The gap between the two rollers is less than the thickness of the starting material and therefore the thickness of the metal is reduced as it passes through the rollers. The torque produced by the rolling mill depends on the mill's diameter. Smaller diameters will result in less torque. The gripping of the work-piece between the rollers is highly affected by the friction at the interface between the material and the rollers. The maximum thickness reduction possible in a single pass is restricted by the friction between the rolls. For a certain friction present, if the gap between the rollers is smaller than a certain value relative to the initial thickness, the work-piece may hit the rollers and slide back with no gripping taking place. To keep the volume rate of flow constant throughout the roll gap, the speed at which the metal moves during rolling should vary. Therefore, the velocity increases as the thickness decreases.
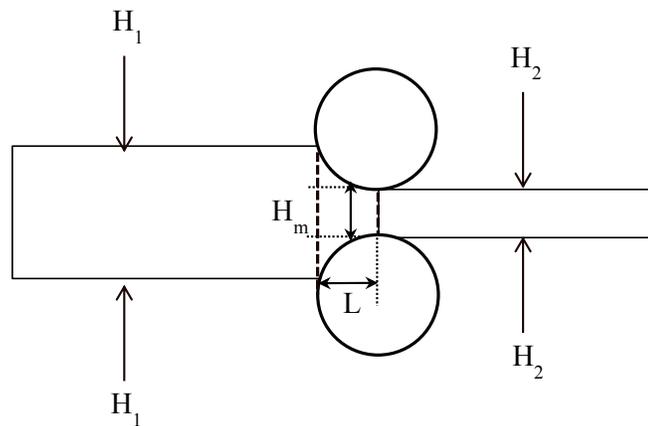
Figure 6.3- Schematic illustration of geometries in rolling.

The force experienced by a work-piece during rolling is influenced by different parameters. Initial temperature of the work-piece significantly affects the rolling force. Increasing the initial temperature (up to a certain point) reduces the rolling force. The rolling force is also affected by the rotation speed of the rollers. Increasing

the roll's rotation speed, the material flow stress increases due to the increase in the average strain rate in the deformation zone. It must be noted that increased strain rate can adversely affect material properties and therefore, very high roll speed reduces the product quality. One important parameter affecting the rolling force is the Roll Gap geometry/shape Factor. The stress regime of the work-piece between the rollers is highly affected by this parameter. The Roll Gap geometry/shape Factor (RGF) is defined as,

$$RGF = \frac{L}{H_m}$$
6.1

Where L and H_m are defined as:

$$L \sim \left(R_r(H_1 - H_2)\right)^{0.5}$$
6.2

$$H_m = \frac{H_1 + H_2}{2}$$
6.3

where R_r is the rollers radius, L is the projected arc of contact between the roll and the work-piece. H_1 and H_2 are the work-piece thickness before and after passing through the rollers, respectively.

The relationship between the RGF and the pressure distribution in a rolling work-piece will be studied in the following sections.

6.1.2 Healing during multi-pass rolling process

The damage healing approach introduced in Chapter 5 will be applied to a rolling case in the following sections. Prior to this, it is important to know that the rolling occurs in a multi-pass process. As explained at the beginning of section 6.1, most of the reduction in the work-piece thickness occurs in the roughing stand. There are two different types of roughing processes. One consists of several rolling stands as shown in Figure 6.1. The other type consists of one single reversing stand through which the

work-piece is moved back and forth. Therefore, if the healing model is applied to the multi-pass rolling, the final closure and healing states of the current roll will be inherited as the starting conditions of the next roll pass.

6.1.3 The FE model

In order to investigate the porosity elimination process during hot rolling, the developed UMAT subroutine (see Chapter 5) has been applied to a case of hot flat product rolling. The FE model represents only one stand in the roughing stage.

A three-dimensional model of two rolls and a billet was provided by TATA steel, a schematic of which is shown in Figure 6.4. The billet has a cross sectional area of 324 cm^2 (18×18 cm) and length of 49.649 cm. Since the billet has a symmetric geometry, only a quarter of the billet has been modelled to save computational time. The rolls have radius of 34.75 cm and $H_m/L = 3.5$.

In Figure 6.4b, plane 'acge' is symmetric relative to the x axis ($U_x = UR_y = UR_z = 0$) and plane 'cdhg' is symmetric relative to the y axis ($U_y = UR_x = UR_z = 0$). The billet has a velocity of 4.49 cms^{-1} in z direction. The velocity of the rollers in x, y, z direction is zero. The angular velocity of the rollers are defined as -0.1288 rad/s and 0.1288 rad/s. The rolls are 3D analytical rigid bodies and are in contact (surface-to-surface contact) with the billet. A 'tangential behaviour' for the contact surfaces was defined using the penalty method. The friction coefficient between the roll and the steel work-piece was taken to be 0.3 (the reduction in the product thickness only occurs if the shear frictional stress is greater than a minimum value). Unless otherwise stated, the default values in ABAQUS/Standard for surface to surface contact were used for all other parameters.

To study the porosity elimination of the work-piece, only a section of the billet has been given porous material properties. The developed UMAT has been applied to the porous section and a finer mesh has been used. Figure 6.5 represents the block

partitioning and the applied mesh. The work-piece consists of a total of 3904 elements with 1464 elements in the porous region. A mesh with 8-node linear brick, reduced integration elements (C3D8R in ABAQUS terminology) was used.

(a)

(b)
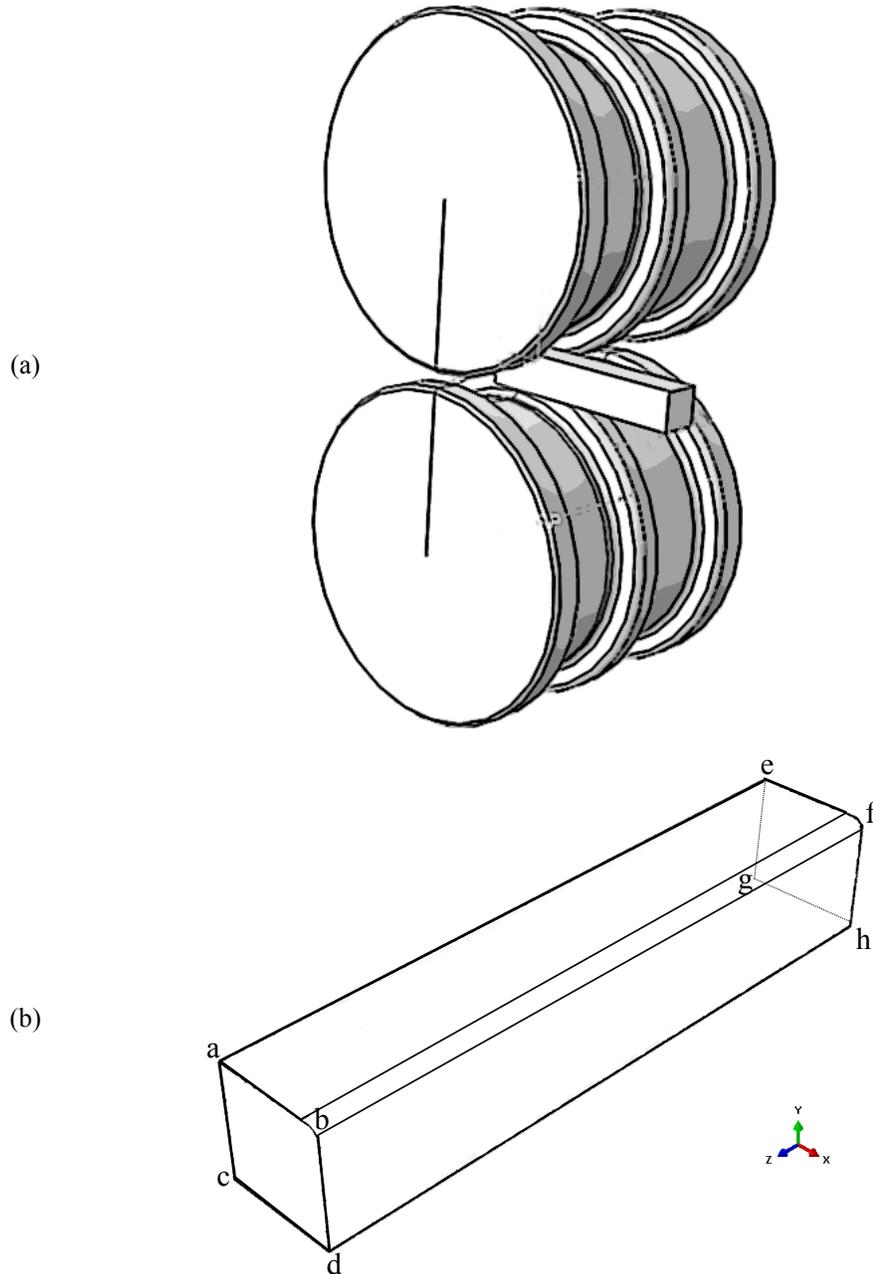
Figure 6.4- *(a) Finite element model of the work-piece between rotating rollers, (b) Finite element model of the billet.*

The non-porous section of the work-piece was modelled using a von Mises yield surface equivalent to the GT model with $vvf = 0$. The initial void volume fraction of the porous section was taken to be $f_o = 0.006$ and vvf_c of 0.003 was assumed. The aforementioned material properties for free cutting steel (FCS) at 900 °C and $\dot{\varepsilon} = 0.1$ (Foster, 2007) (see Appendix A) were used. The calibrated fitting parameters, $q_1 = 2.0$, $q_2 = 1.25$ and $q_3 = 4.0$ were obtained for an average triaxiality, $T = -0.5$ (see Chapter 4, Table 4.4).

Figure 6.6 shows the pressure distribution and evolution throughout the work-piece as it is compressed through the rollers. The compression may be sufficient at some positions in the work-piece and therefore closure and healing may occur in some parts. However at some positions the amount of compression and plastic deformation may not be enough for a closure state to be reached and healing may never occur. In Figure 6.7 some of the elements have reached the closure state and started to heal. These elements are shown in red. As the work-piece is rolled more elements experience compression and the number of these element (shown in red) increase.
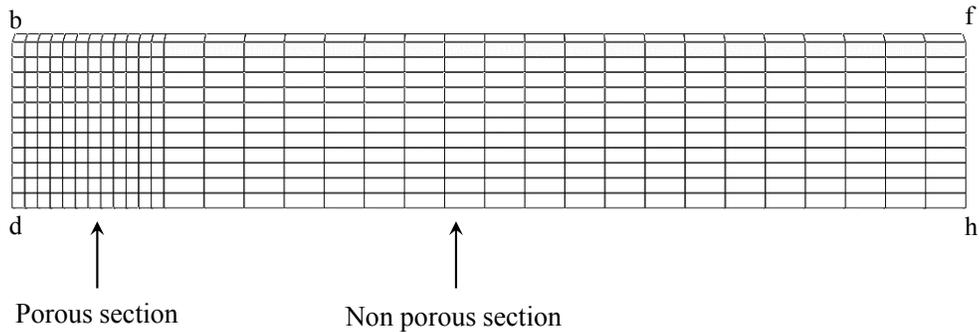
Figure 6.5- Illustration of the block partitioning and the applied mesh.

Pressure [MPa]

	+1.508e+02
	+1.405e+02
	+1.229e+02
	+1.053e+02
	+8.763e+01
	+7.000e+01
	+5.236e+01
	+3.473e+01
	+1.709e+01
	-5.467e-01
	-1.818e+01
	-3.582e+01
	-5.345e+01
	-7.109e+01

t = 2.3 s

t = 3.3 s

t = 4.2 s

t = 5.7 s
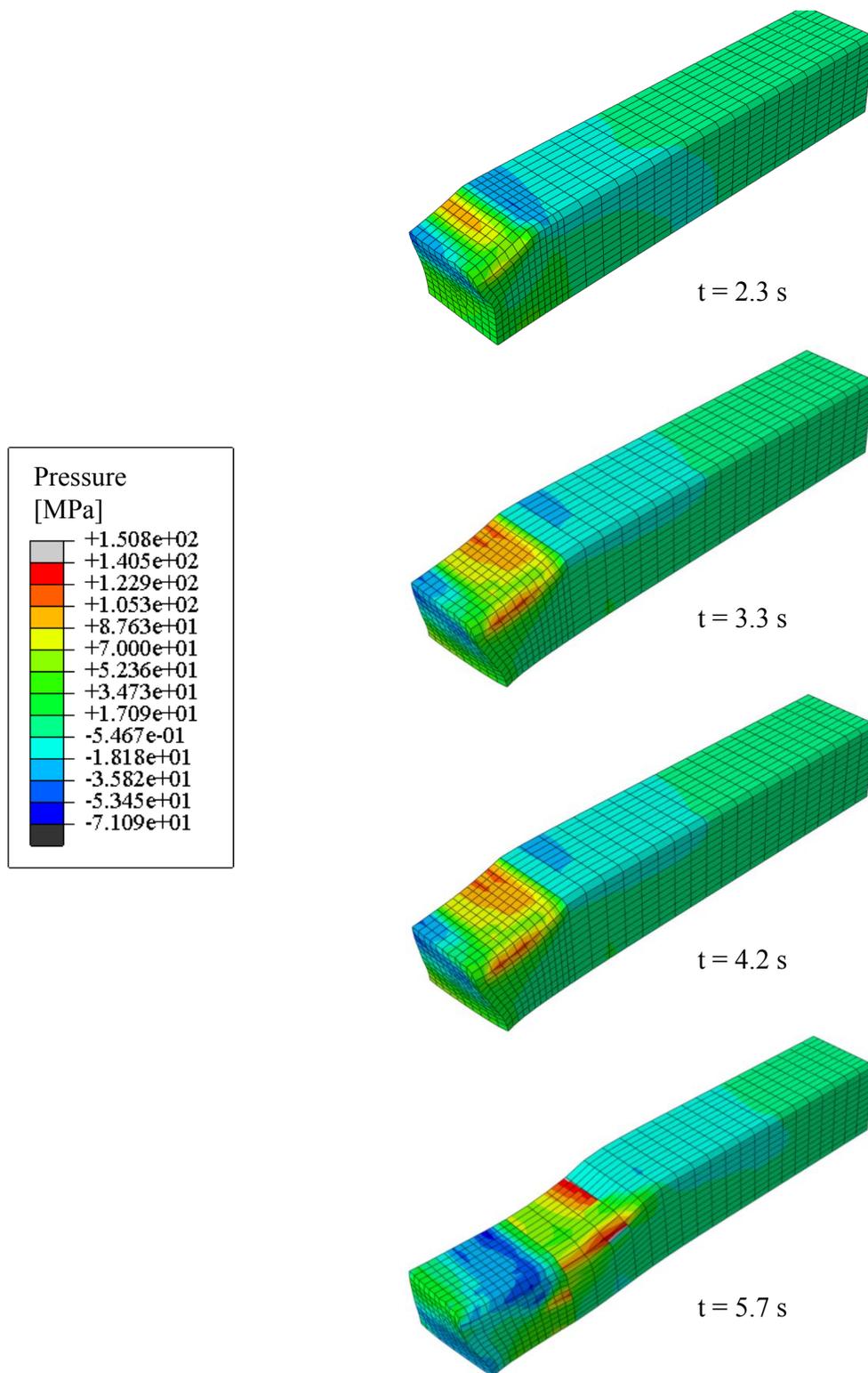
Figure 6.6*- Pressure distribution along the work-piece (positive values represent compression).*

154

t = 2.3 s

t = 3.3 s

t = 4.2 s

t = 5.7 s
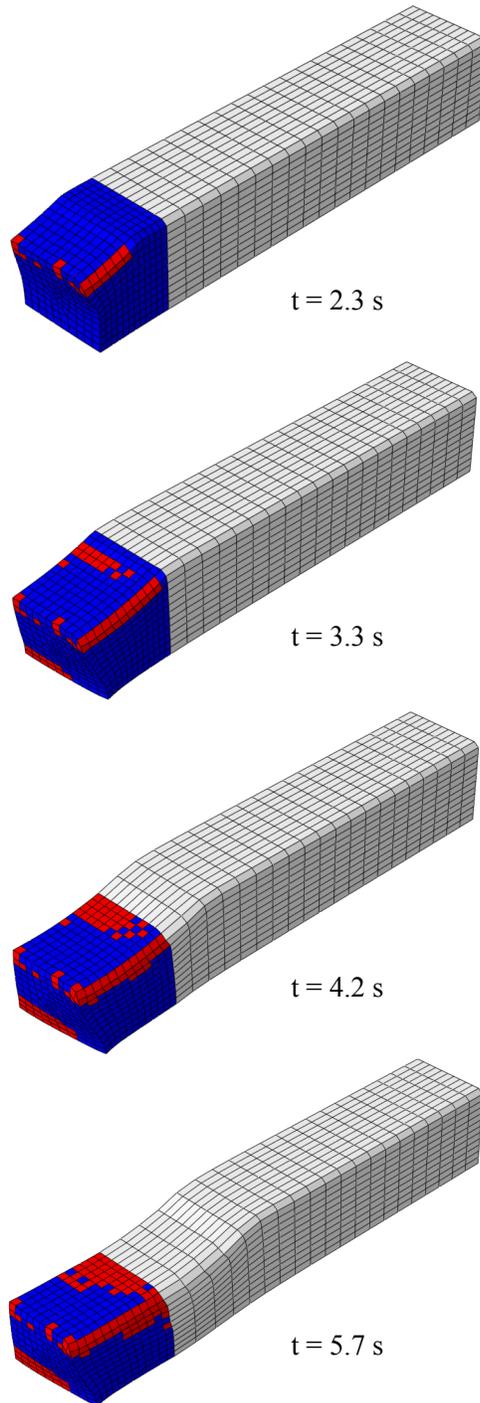
Figure 6.7*-Void elimination process during rolling. Blue represents elements which have not reached the closure state. Red represents elements which have reached the closure state and have started to heal.*

155

It should be noted that in Figure 6.7, although some elements (represented in red) have reached the closure state and have started to heal, more time is required for these elements to heal completely. Since the stress state in the work-piece is non-uniform, the healing times are different for each element. To have an approximate estimation of the predicted healing time for the studied case, three random elements were selected from the model as shown in Figure 6.8. The change in *vvf* with time during healing for these elements is shown in Figure 6.9. This figure shows predicted healing times of 19.1 s, 23.6 s and 20.4 s for element A, B and C, respectively. The average healing time for all closed elements (red elements in Figure 6.7) is 18.1s.
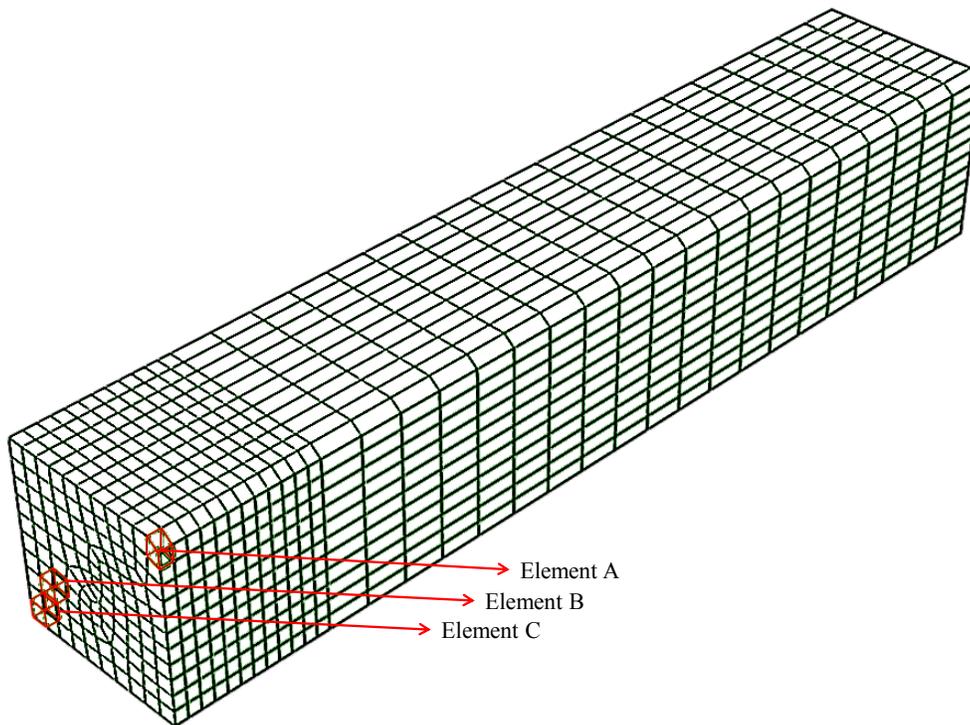
Figure 6.8- Elements selected for investigation of healing time.
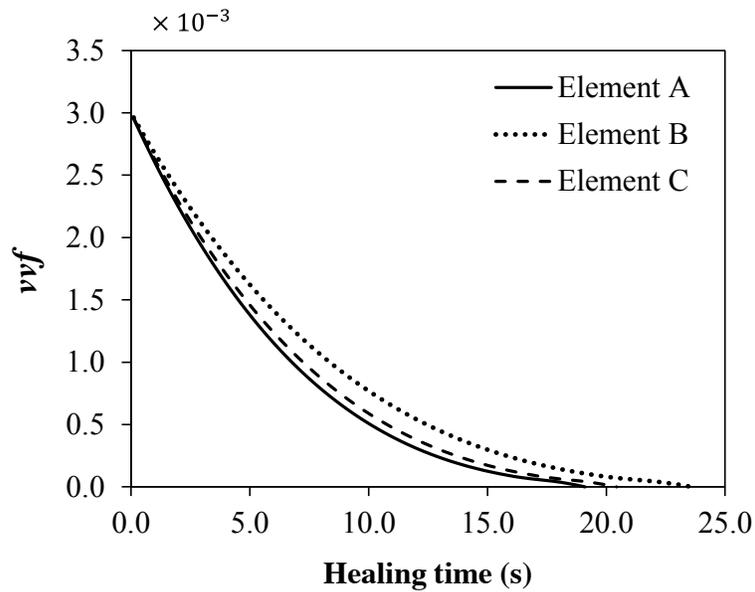
Figure 6.9- *vvf versus time for element A, B and C (see Figure 6.8)*

In the studied case considering the work-piece dimensions and the rotation speed of the rolls, the time for the entire work-piece to pass through the rolls is about 11 seconds. It is clear that since the work-piece is moving through the rolls, different locations of the work-piece experience the compression for less than 11 seconds (e.g. about 1 to 2 seconds). Comparing this to the predicted healing times, the healing cannot be achieved within a single roll pass. It is clear that the predicted healing times are associated with a specific condition (H_m/L, f_o, vvf_c, temperature, etc.). If these conditions change, the healing time will be different. For example, for smaller values of H_m/L, the healing time will be shorter. In that case, considering the fact that rolling (roughing stage in the studied case) involves multiple passes (see section 6.1.2), healing might be possible.

In cases where the applied conditions (e.g H_m/L, f_o, vvf_c, temperature, etc.) result in longer healing times (e.g. 2-3 minutes), reducing the roll rotation speed and increasing the number of roll stands would favour the healing process, but this would reduce output and add cost. Further, this would cause heat loss in the work-piece. Apart from performing the rolling process in furnace rooms, it may not be possible to heal under these conditions.

6.1.4 The effect of roll gap geometry factor on porosity

closure

As mentioned in section 6.1.1, the stress regime of the work-piece between the rolls is highly affected by Roll Gap geometry/shape Factor. For $H_m/L > 4.8$ (where $H_m/L=RGF^{-1}$) all principal stresses are tensile, therefore representing a regime of tensile triaxiality. For $1.8 < H_m/L < 4.8$, the vertical principal stress is compressive, but all the other stresses are tensile. For $1 < H_m/L < 1.8$, the hydrostatic stress becomes compressive therefore porosity closure will be promoted at a greater rate as H_m/L decreases (Farrugia, 2012). Here the porosity closure has been investigated for a range of H_m/L values, with the understanding that greater closure will enable greater healing; the trend with H_m/L can be understood without simulating healing. The FE model explained in section 6.1.3 was used for this purpose (*vvf* of 0.02 was assumed). To compare the closure extent for different H_m/L cases, closure percentage as defined in equation 6.4 has been obtained for H_m/L values of 2.5, 3.5, 4.5, 6.5 and 8.

$$Closure\ percentage = \frac{number\ of\ elements\ closed}{total\ number\ of\ elements} \times 100 \qquad 6.4$$
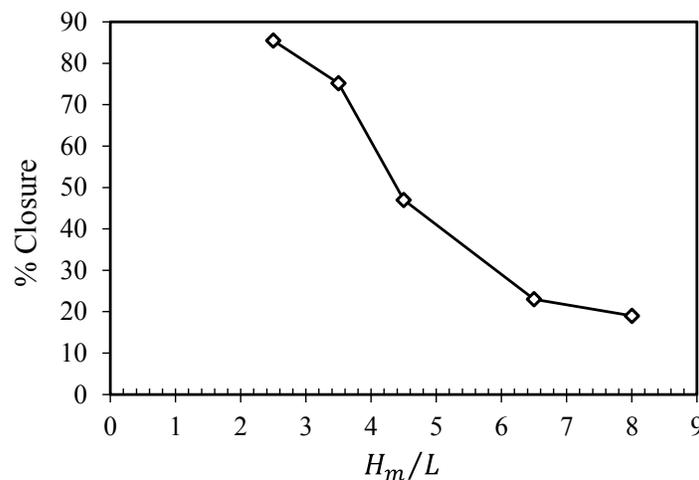
***Figure 6.10**- Closure percentage obtained during rolling for different H_m/L values.*

As explained before for $H_m/L > 4.8$ all principal stresses are tensile, therefore representing a regime of tensile triaxiality. Looking at Figure 6.10, the closure percentage (% Closure) is low for H_m/L values of 6.5 and 8. For $1.8 < H_m/L < 4.8$, the vertical principal stress becomes compressive, therefore porosity closure will be promoted. Figure 6.10 shows higher closure percentage for H_m/L values of 2.5 and 3.5. It is worth mentioning that the value of vvf_c is an assumption here (vvf_c=0.015); in real cases the value of vvf_c might be smaller and therefore the closure percentage values could be different.

6.1.5 The effect of initial porosity size on porosity closure

The porosity elimination in rolling is highly affected by the amount of initial porosity present in the work-piece, or in other words, the initial void volume fraction (f_o). Higher values of f_o, require larger compression (larger reduction in work-piece thickness) and also longer times. Figure 6.11 compares the percentage of closure of a billet with initial porosity values of $f_o = 0.02$ and $f_o = 0.03$ rolled under the same conditions. As it is evident from Figure 6.11 for initial void volume fraction of $f_o = 0.02$, the amount of compression has been sufficient to produce a relatively high closure percentage of around 91%, whereas for a slightly higher initial porosity of $f_o = 0.03$, the closure percentage is relatively smaller (~19 %).
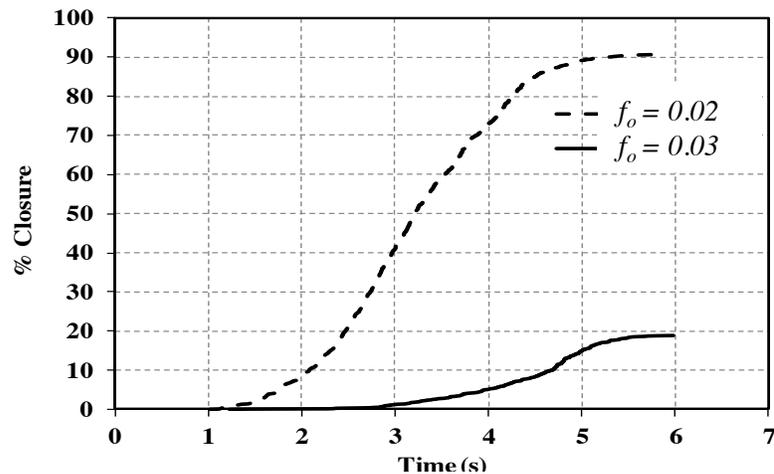
Figure 6.11*-Closure percentage versus time for two different values of initial porosity f_0.*

6.2 Modelling of non-uniform initial porosity distribution

While porous ductile material models are usually based on assumptions of uniform void distributions, the cases of non-uniform porosity is common in real case scenarios. Attempts have been made to model non-uniform porosity distribution. Studies such as that of Oho and Hutchinson (1984) have investigated the effects of an inhomogeneous distributions of voids on material behaviour. To model the non-uniform distribution of the voids Ohno and Hutchinson considered an infinite block of material containing an infinite planar band with an excess void level. The initial void volume fraction outside the band was specified as f_1, while the initial value inside the band was denoted by f_1^b.

Ohno and Hutchinson's model uses Gurson theory to characterize the material, and the initial void volume fraction within the band is taken to be axisymmetric with

$$f_1^b = f_1 + \Delta \exp\left\{-\frac{1}{2}(\frac{r}{s\lambda})^2\right\} \qquad\qquad 6.5$$

where λ is the initial thickness of the band and is also identified as the initial average void spacing. Other conventions are shown in Figure 6.12. The initial void volume fraction inside the band for $r \gg s\lambda$ is also taken as f_1.

Therefore, the non-uniformity of the initial void distribution is a disk-shaped cluster of thickness λ and approximate radius $s\lambda$ whose excess in volume fraction at its centre over the uniform background level is Δ.

The main aim of Ohno and Hutchinson's work was to define the effect of cluster size, S on the ductility. They considered the interaction between the band and the outside material and governed constitutive equations to determine the critical condition for localisation. The implementation of their work, however, to model

spatial distributions of voids in a work-piece in a finite element approach is not feasible.
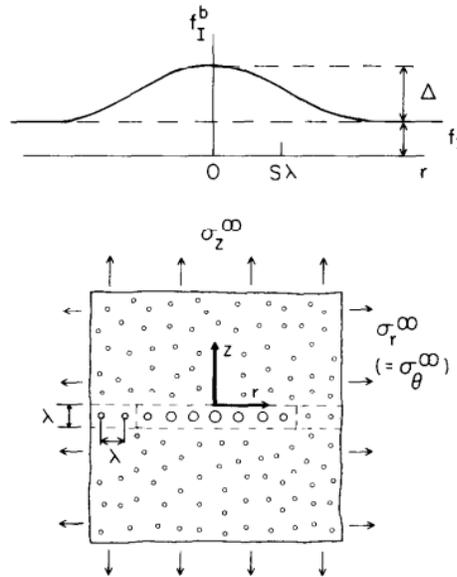
***Figure 6.12**- Conventions used in Ohno and Hutchinson's approach (1984).*

A more applicable approach is that of Becker. He studied the effect of a non-uniform distribution of porosity on flow localization and failure in a porous material (Becker 1987). To model void distribution Becker used a sub-modelling approach by dividing the total area into sub-regions or cells with different void volume fractions. Each cell had a specific void density and was regarded as a continuum. The porosity distribution was mimicked from a porous specimen. The specimen surface was divided into frames, where each frame was adjacent to the previous one. The void area fraction of each frame was measured with an Optomax optical image analysis system. Each frame contained approximately 40 voids with the area fraction being calculated by dividing the void area by the total frame area.

Zhang et al. (1999) have also criticised the assumption of uniform void size and void spacing in porous material damage models, stating that this assumption ignores the effect of void interaction and therefore overestimates the material resistance to fracture. They investigated the effect of void size and spacing and void interaction by

applying different axisymmetric cell models containing one void or two voids with different geometries. Figure 6.13 illustrates the geometry of a cylindrical cell containing two voids used by Zhang et al. Comparison of cell models having identical void spacing and that of different void spacing showed that the void growth was faster for the cell with void spacing smaller in the cross section direction than that in the tension loading direction.

The approach of Zhang et al. is suitable for cases of cell model studies, however implementation of their work to a damage model which is to be used for large scales requires changing or deriving of new constitutive equations in the model to account for void spacing and void size.

This study will follow Becker's approach to model porosity distribution in a cast work-piece.
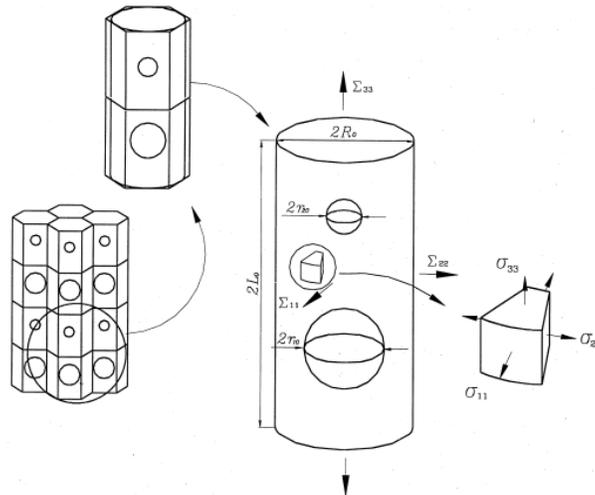
Figure 6.13- The geometry of cylindrical cell containing two voids (Zhang, Bai et al. 1999)

6.2.1 Finite element modelling of initial porosity distribution

As it was shown in Chapter 5, in steel casts the porosity is usually localized and non-uniformly distributed relative to the entire part. In steel cast products the porosity is more concentrated in the centre. It is known that central porosity is basically due to

the volumetric contraction of steel during solidification. Figure 6.14 shows a chain of voids found in the core of a cast steel bloom (see Figure 5.1 in Chapter 5 for more details). This localised type of porosity is highly undesirable and causes inhomogeneties in the material properties.

As shown in Chapter 5, moving from the bloom centre towards the surface, the porosity reduces in size and concentration (see Figure 5.2). Small patches of micro-porosity are likely to be seen in the region between the centre and surface. However, the surface is very unlikely to have porosity and is usually assumed to be non-porous (Farrugia 2012).

Figure 6.14- A chain of voids found in the core of a cast steel bloom.

So far an even distribution of initial porosity, f_0, has been considered in the developed void elimination UMAT. To model non-uniform initial porosity spread, instead of assuming a constant value of f_0, it will be defined as a function which changes with coordinates. In order to obtain any initial porosity distribution, this function has to be replaced with the function defining the distribution of interest.

Here two different configurations have been considered. The first attempt considers a linear porosity spread, increasing from surface to the centre of the work piece. The second attempt models a Gaussian or normal distribution of porosity.

To obtain a linear porosity distribution where the porosity increases linearly from surface to the centre of the work-piece, the configuration in Figure 6.15 is used, where (x_0, y_0) represents the very centre of the work-piece and (x_1, y_1) represents

the corner surface of the work-piece. Equations for planes (1) and (2) can therefore be obtained. The surface describing the linear porosity distribution can therefore be defined as:

$$f = \begin{cases} -f_o \dfrac{y - y_1}{y_1 - y_o}, & y \geq y_l \\ -f_o \dfrac{x - x_1}{x_1 - x_o}, & y < y_l \end{cases}$$
6.6

where y_l is the diagonal of the work-piece shown as dashed line in Figure 6.15.

$$y_l = \frac{y_1 - y_0}{x_1 - x_0} x + b, \quad b = y_0 - \frac{y_1 - y_0}{x_1 - x_0} x_0$$
6.7
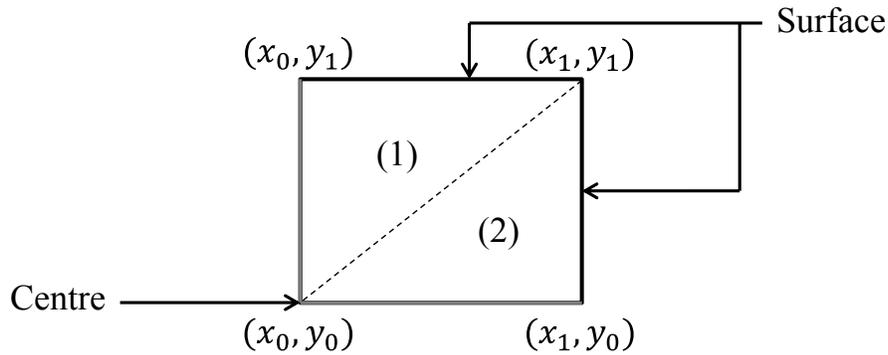
Figure 6.15 *- The cross section of the work- piece and the coordinates.*

The obtained surface has been plotted in Figure 6.16, where maximum and minimum porosity are shown in red and blue respectively.

As a second attempt, a two-dimensional Gaussian function has been used to spread the porosity.

$$f(x,y) = A\exp\left(-\left(\frac{(x - x_o)^2}{2\sigma_x^2} + \frac{(y - y_o)^2}{2\sigma_y^2}\right)\right)$$
6.8

Here the coefficient A is the amplitude, (x_0, y_0) is the centre coordinate and σ_x, σ_y are the x and y spreads of the function. Figure 6.17 was created using $A = f_0$, $\sigma_x = \sigma_y = 50$ and (x_0, y_0) as the coordinates of the work-piece centre.
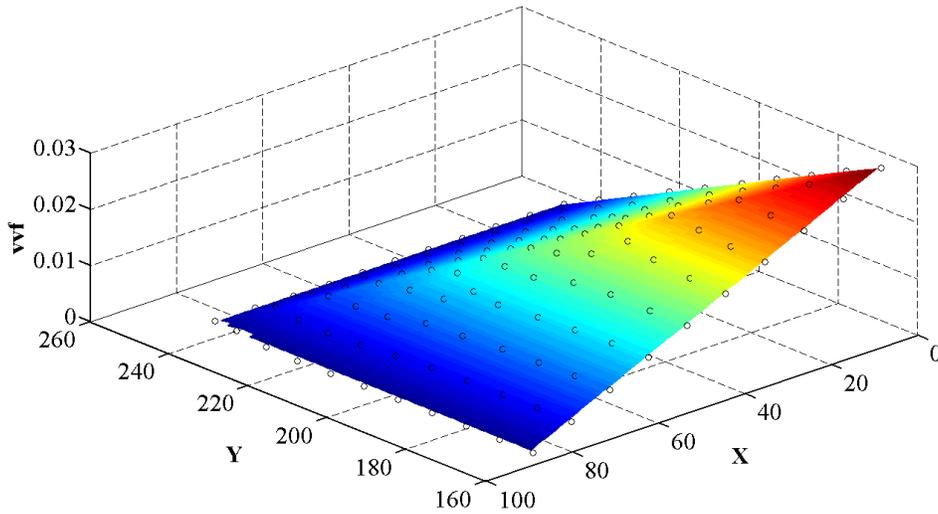
Figure 6.16-*Linear distribution of porosity in the work-piece.*
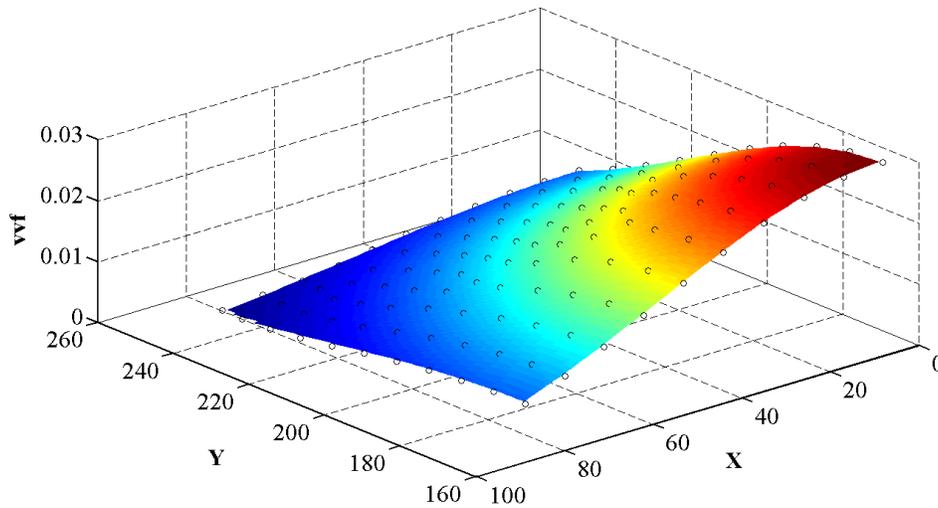
Figure 6.17- Normal distribution of porosity in the work-piece.

The initial porosity with three different distributions of even, linear and Gaussian for a section of the work-piece has been illustrated in Figure 6.18(a), (b) and (c), respectively. The Python codes developed for modelling non-uniform initial porosity are provided in Appendix G.

165
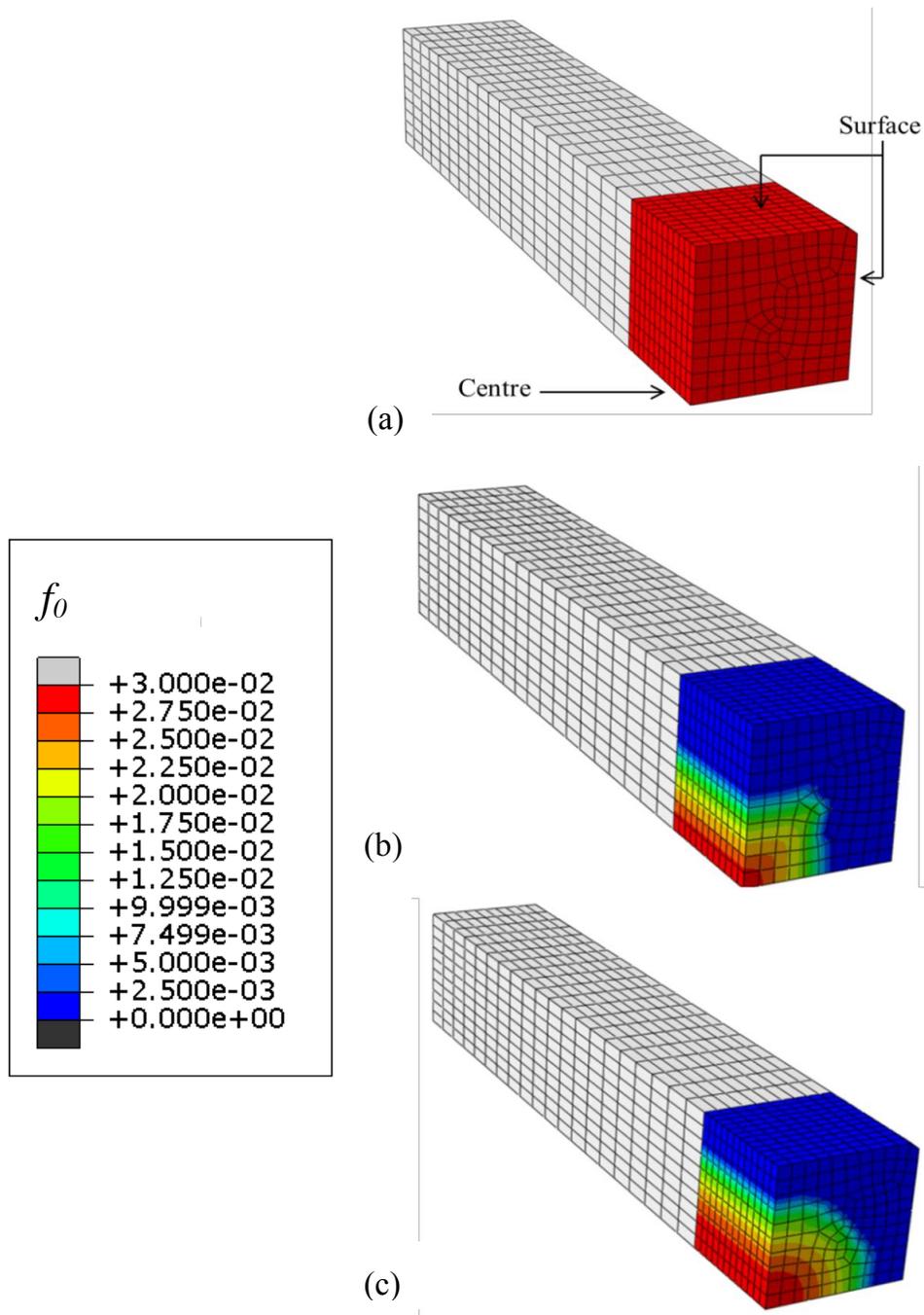
Figure 6.18- The initial porosity with three different distributions of (a) Even, (b) Linear and (c) Gaussian (normal).

166
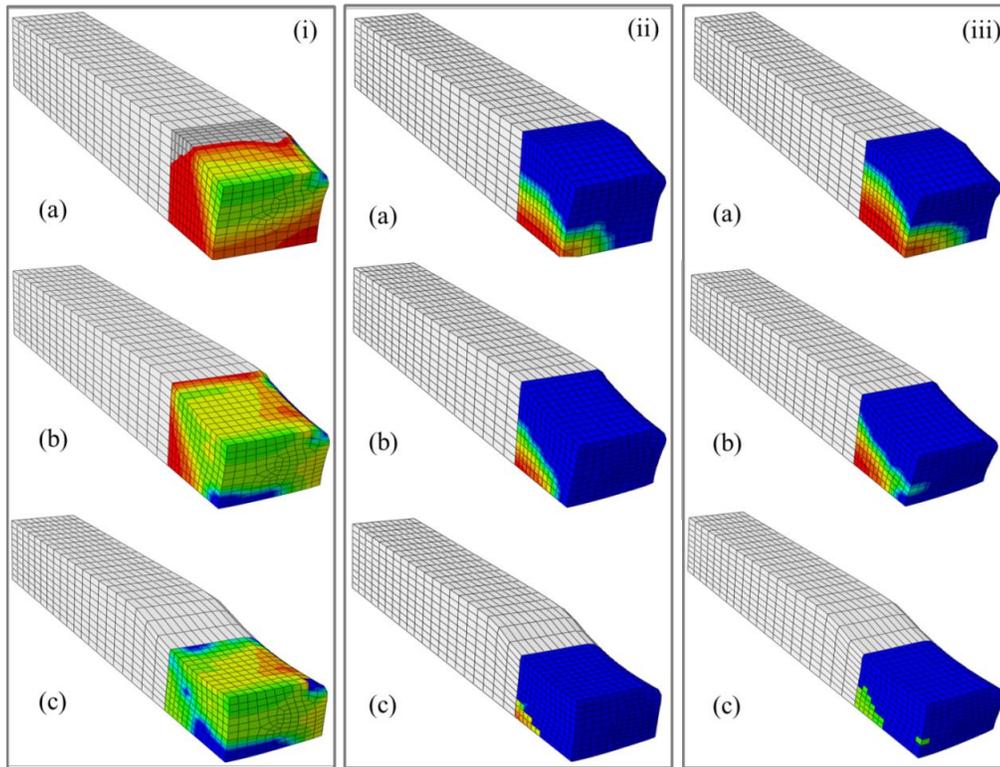
***Figure 6.19**- Comparison of the change in void volume fraction during rolling for (i) evenly, (ii) linear and (iii) normal porosity distributions at three steps: (a) sometime after the roller has touched the section, (b) the roller in middle of the section, and (c) the roller passed through the whole section.*
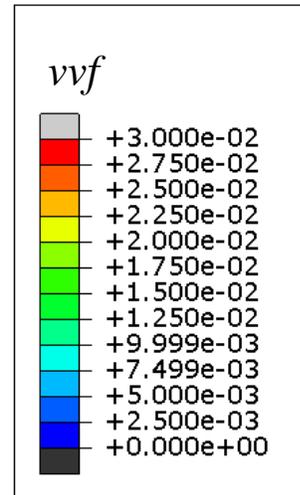
Figure 6.19 compares the change in void volume fraction as the thickness of the work-piece is reduced during rolling, for three different examples of initial porosity distributions. In all three cases the deformation process has been shown in three steps of (a) sometime after the roller has touched the section, (b) the roller in middle of the

167

section, and (c) the roller passed through the whole section. Figure 6.19 (i) illustrates the change in vvf for a case of evenly distributed initial porosity. Figure 6.19 (ii) and (iii) show the porosity spread for linear and normal distributions, respectively. Comparison of the three initial porosity distributions shown in Figure 6.19 indicates that a much lower amount of porosity remains in the work-piece after deformation for linear and normal initial porosity distributions compared to the case of evenly distributed initial porosity.

Figure 6.20 compares the healing percentage versus time for the three considered distributions of initial porosity. It can be seen that for the even distribution of initial porosity the closure percentage achieved is the lowest (18.7%) compared to other cases. This is clearly due to the total initial void volume fraction being larger for the even distribution compared to linear and normal distributions.
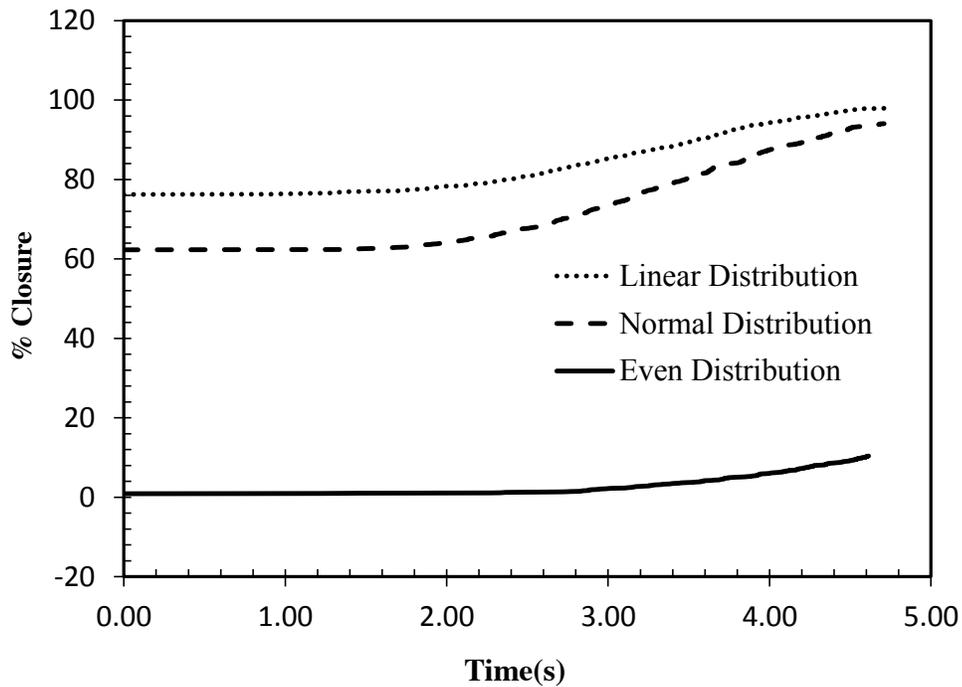
Figure 6.20- *The comparison of closure percentage during rolling for three different initial porosity distributions*.

6.3 Conclusions

The developed UMAT has been applied to model the void elimination of a steel work-piece during rolling. The effect of Roll Gap geometry/shape Factor (RGF=L/H_m) on porosity elimination during rolling was investigated. For $H_m/L > 4.8$ all principal stresses are tensile, representing a regime of tensile triaxiality which leads to lower healing percentages compare to cases of $1.8 < H_m/L < 4.8$, when the vertical principal stress becomes compressive and porosity closure occurs.

The effect of initial void volume fraction on porosity elimination in rolling was also investigated. It was observed that for larger initial void volume fractions the percentage was lower. This was due to the amount of compression for larger values of f_0 not being sufficient at some locations in the work piece to reach the closure state.

Finally, non-uniform initial porosity distributions in a work-piece were modelled. Instead of defining a constant value for f_0 (which creates an even distribution in the work-piece), f_0 was defined as a function which changed with coordinates. Two cases of linear and normal distribution were investigated. To model any distribution the user has to only replace the function with that of interest.

6.4 References

Becker, R. (1987). "The effect of porosity distribution on ductile fracture." <u>Journal of the Mechanics and Physics of Solids</u> **35**(5): 577-599.

Farrugia, D. (2012). TATA Steel technical presentation, TATA Steel.

Kalpakjian, S. and S. R. Schmid (2001). <u>Manufacturing Engineering Technology</u>. New York, Pretince-Hall inc.

Ohno, N. and Hutchinson.J.W. (1984). "Plastic flow localisation due to non-uniform void distribution." <u>Journal of the Mechanics and Physics of Solids</u> **32**(1): 63-85.

Udomphol, T.(2007). "Rolling of Metals." 2013, from <u>http://eng.sut.ac.th/metal/images/stories/pdf/03_Rolling%20of%20metals.pdf</u>

Zhang, K. S., J. B. Bai and D. Francois (1999). "Ductile fracture of materials with high void volume fraction." <u>International Journal of Solids and Structures</u> **36**(23): 3407-3425.

Chapter 7

Experimental work

Devising an appropriate healing process which does not cause discontinuity in the microstructure and mechanical properties at the bonding sites requires an accurate choice of thermo-mechanical processing parameters.

An experimental technique has been developed to estimate the load and the time required for porosity healing in Free Cutting Steel (FCS) at 1000°C. Different analysis techniques have been used to examine the extent of healing and the optimum bond parameters have been obtained. Experimental results are used to calibrate a healing model previously introduced in Chapter 5.

7.1 The aim and objectives of the experimental program

Healing as a process in which material defects such as voids and cracks reduce in size, resulting in a stiffer and stronger material with enhanced mechanical properties, has been an objective in the processing of many materials. Applications of void healing can be seen in polymers and composites (Blaiszik et al., 2010, Wool and O'connor, 1981), biomaterials (Fratzl and Weinkamer, 2008, Vermolen et al., 2008) and also the recovery of concrete (Li and Yang, 2008, Reinhardt and Jooss, 2003). Healing in metal processing has been mostly studied in terms of sinter powder metallurgy where mass transfer of metal occurs at high temperatures in the range of 0.7 to 0.9 of the melting temperature, producing either solid state or liquid phase bonding across powder interfaces (Lumley and Polmear, 2007).

Although healing and diffusion bonding has been widely studied for different alloys, the conditions required to create self-bonding of porosity in FCS has never been inspected and is therefore the main focus of this chapter. It is also worth mentioning that in the majority of studies done on bonding, the examination of the bond strength has been ignored. In most cases SEM and optical microscopy have been used to observe the trace of the bonding line. This study however, offers a better method of checking if a completely strong bond has been made.

To improve the quality of jointed interfaces, it is essential to consider the optimal conditions for bonding. The main factors affecting the quality of the bonds include temperature, pressure and time. The aim of this study is therefore to develop experimental techniques to estimate the load, temperature and time required for healing of FCS as-cast grades. The relationship of the bond parameters and tensile strength of the joints has been discussed. Image analysis techniques have been used to examine the extent of healing and the optimum bond parameters have been

obtained. The experimental results are used to calibrate the void elimination model outlined in Chapter 5.

7.2 The experimental method

A number of cylindrical samples manufactured from non-porous as-cast FCS billets provided by TATA Steel were cut in halves and were compressed under various loading ranges. The holding time was varied in a systematic approach to ensure an accurate estimation of the required load and healing time. The extent of healing was examined by performing a series of tensile tests and inspecting the flow curves of the specimens.

7.2.1 Test Materials

The material was provided by TATA steel in the as-cast state as a number of work-piece cuts. The composition tested was reduced leaded free-cutting steel enhanced with bismuth and tellurium, Telby+ (Table 7.1)

Table 7.1 - Chemical composition of leaded free cutting steel (LFCS) enhanced with Telby+.

C	Mn	P	S	Si	Other(Pb)
0.07	1.00	0.05	0.30	0.01	< 0.4

Test material was supplied as the final 400mm length of a 355×750mm continuous cast work-piece. Defects or porous features are more likely to be found in the centre of the work-piece than the edge. A $50 \times 50 \times 400$mm sample was taken from the work-piece edge as shown in Figure 7.1.
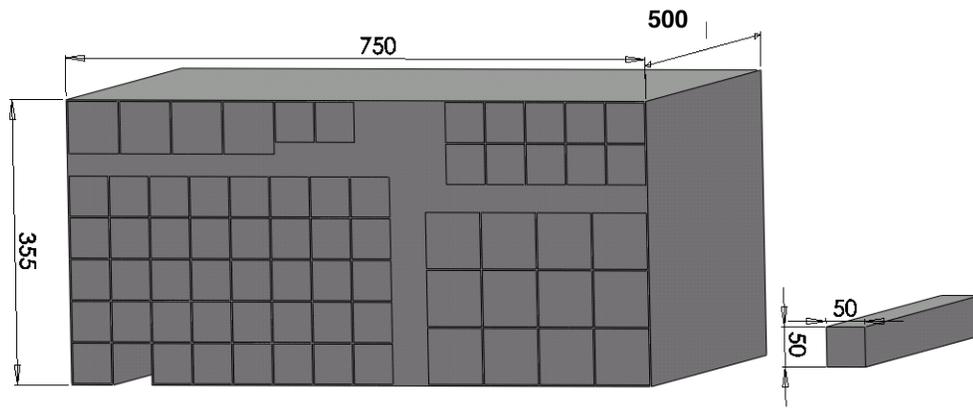
Figure 7.1*- Cast work-piece sectioning plan as defined by TATA*.

7.2.2 Equipment

The tests were conducted in a Gleeble 3800 material test station (Figure 7.2). The Gleeble 3800 is a fully integrated digital closed loop control thermal and mechanical testing system. These machines typically have a high speed heating system, a servo hydraulic system and a computer control and data acquisition system.

The Gleeble uses a direct resistance heating mechanism, in which a low current frequency causes a uniform current density throughout the volume of the specimen. This indicates that the specimen is heated at the same time in the whole volume. Power is controlled by adjusting the phase difference between voltage and current. Electrical current is transferred to the specimen by means of a static and a dynamic jaw. The jaws are water cooled assuring a high heat extraction at each side of the specimen. Therefore only a small part at the centre is held at the prescribed temperature.

Specimen temperature is monitored by thermocouples mounted to the specimen surface at middle of its length. The thermocouple follows the programme cycle to control the electrical current fed into the test-piece enabling the system to simulate

174

different time and temperature conditions. Here K-type thermocouples were used which provides accurate temperature measurement up to 1250ºC.

Figure 7.2*- Gleeble 3800 testing system*.

Figure 7.3*- The extensometer, C-Gauge*.

A jaw-to-jaw transducer called an L-gauge, can be used to measure the length change of the specimen from one jaw to the other jaw; however, this measurement includes the compliance of the jaws and is not very accurate. Therefore the diameter change of a specimen is measured at the specimen centre by an extensometer, C-gauge (Figure 7.3). The reading of the C-gauge is used to determine the stress and strain rate.

Figure 7.4 shows details of the specimen set up within the test chamber and is suitable for both tension and compression tests. Copper grips are used to clamp specimens for testing. Copper grips have very high thermal conductivity and are good for hot ductility studies and welding and casting simulations. Another option is stainless steel hot grips, which due to lower thermal conductivity are more suitable for isothermal testing. Two thermocouples have been mounted on the specimen to monitor the temperature on either side of the middle cut as shown in Figure 7.5. The temperature difference between the two thermocouples was about 30°C (thermocouple 1 showed a temperature of 970°C and thermocouple 2 showed 1000°C) which is insignificant relative to the maximum temperature of 1000°C and indicates a well distributed temperature along the gauge length.
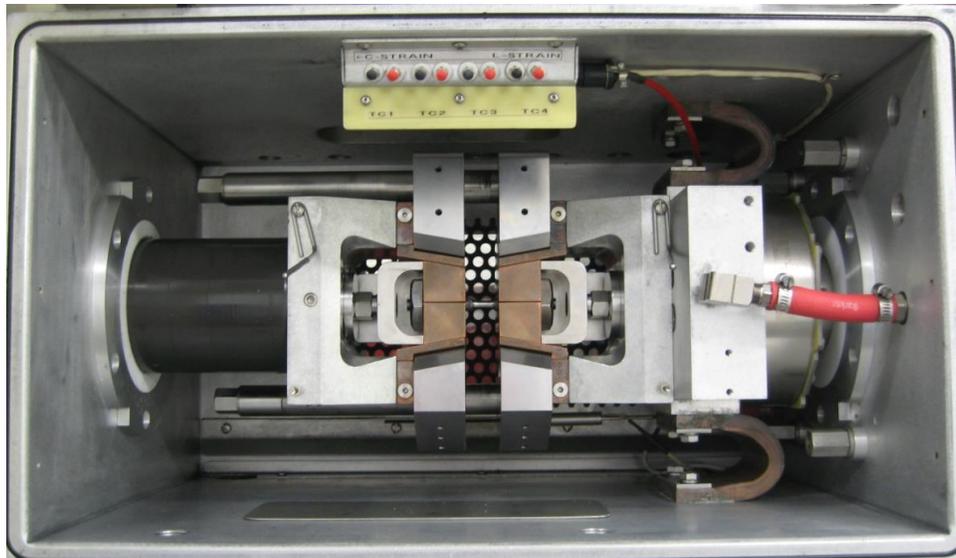
Figure 7.4- *The specimen set up within the test chamber*.

Thermocouple 1 — Thermocouple 2

Hot zone

Bonding interface
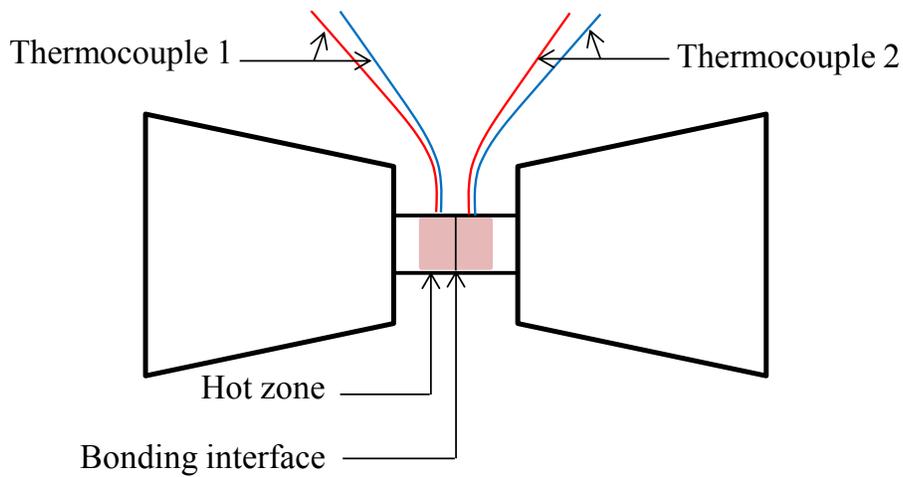
Figure 7.5 *- The thermocouple set up on the specimen.*

7.2.3 The specimens

The specimens were machined into cylindrical shapes with threaded ends (Figure 7.6) and were ground at the cut surface with P2500 abrasive paper followed by cleaning in ethanol to produce a smooth surface finish (with roughness no greater than 9.1μm). The surface preparation was done a few hours prior to diffusion bonding and specimens were kept in a desiccator jar to prevent oxidation. Each of the halved specimens had a length of 55.5mm and a diameter of 10mm; the adopted specimens' shape and key dimensions are shown in Figure 7.7. The test procedure comprised compressing the two halves of each specimen, which were put back together such that their axes were aligned, under various combinations of load and time.
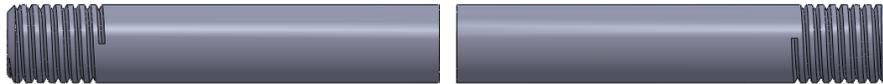
Figure 7.6 *- The cylindrical specimen cut into halves.*
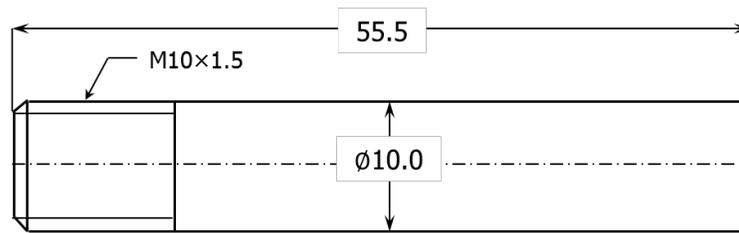
Figure 7.7- *Test specimen dimensions (units in mm).*

7.2.4 Test procedure

The Gleeble direct resistance heating mechanism requires sufficient initial contact between the two halves of the specimen. To achieve this, the specimen was preloaded. It was found that an applied load of -0.5 kN corresponding to a pressure of 6.4 MPa for a 10 mm diameter, maintained for a minimum of 30 s produced the contact conditions necessary to achieve a temperature of 1000°C by direct resistance heating. Following the preloading step the load was increased to one of three values, and the specimens were held under that condition for varying time periods. Table 7.2 provides a summary of the test load and time combinations studied for 1000°C. A schematic diagram of the time, temperature and pressure variation is also illustrated in Figure 7.8.

Table 7.2- *The load-time combinations tested.*

Test No	Load (kN)	Pressure (MPa)	Time
1	1.0	12.7 ($\approx 0.4\sigma_y$)	2.5 min
2	1.0	12.7 ($\approx 0.4\sigma_y$)	4 min
3	1.0	12.7 ($\approx 0.4\sigma_y$)	5 min
4	1.0	12.7 ($\approx 0.4\sigma_y$)	10 min
5	3.0	38.2 ($\approx 1.3\sigma_y$)	20 s
6	3.0	38.2 ($\approx 1.3\sigma_y$)	30 s
7	3.0	38.2 ($\approx 1.3\sigma_y$)	1 min
8	3.0	38.2 ($\approx 1.3\sigma_y$)	2 min
9	5.0	63.7 ($\approx 2.1\sigma_y$)	10 s
10	5.0	63.7 ($\approx 2.1\sigma_y$)	20 s
11	5.0	63.7 ($\approx 2.1\sigma_y$)	30 s
12	5.0	63.7 ($\approx 2.1\sigma_y$)	1 min
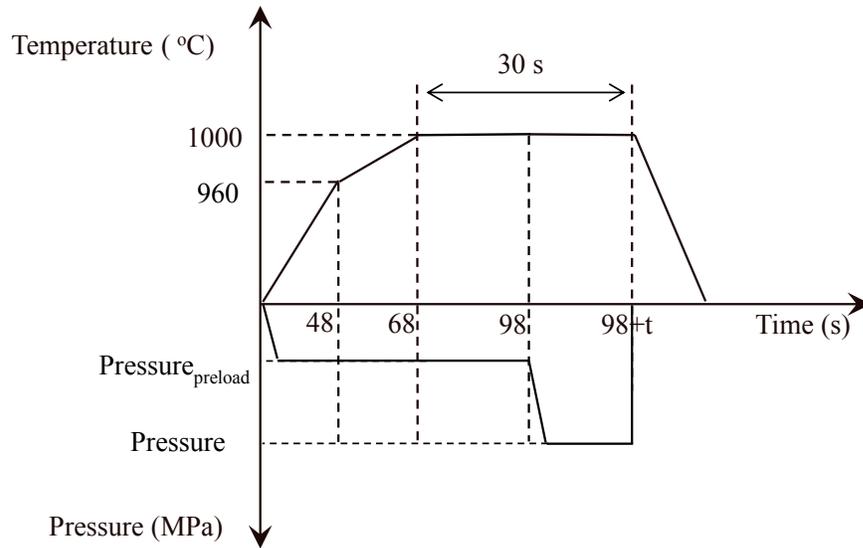
Figure 7.8*- Schematic of the time, temperature and pressure variation during the test (positive pressure is downwards).*

7.2.5 Tensile tests

Tensile tests on the bonded specimens were subsequently carried out to measure the strength of the bonded region. The tensile test specimens were necked in the middle; the adopted geometry and the key dimensions of the specimens are shown in Figure 7.9. The tests were performed at room temperature and a uniform strain rate of $0.2 \ s^{-1}$ was employed. A C-gauge was utilised to measure the change in diameter and the diametral strain was calculated.
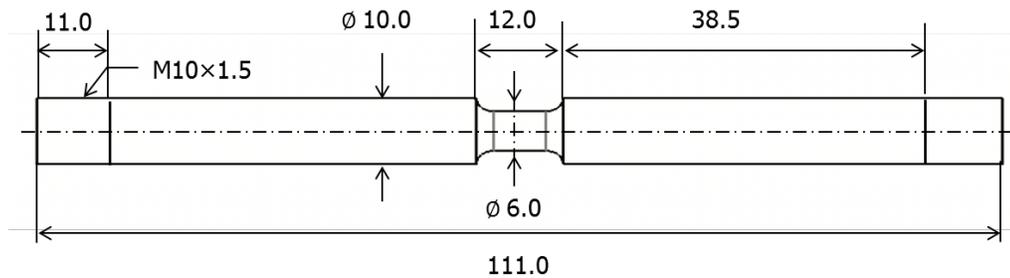
Figure 7.9*-Uniaxial test piece geometry and dimensions (units in mm).*

7.2.6 Experimental results

The flow curves obtained for bonds formed at 1000°C under compressive loads of 12.73 MPa, 38.2 MPa and 63.7 MPa are shown in Figure 7.10 to 7.12 respectively. It is apparent from Figure 7.10 that under a compressive load of 12.73 MPa, holding times of 2.5 min and 4 min are not sufficient for a strong bond to form; for these holding times the bonds broke under considerably lower stress and strain values. This is also evident from Figure 7.13, where it can be observed that the samples held for 2.5 min and 4 min (Figure 7.13(a), and 7.13(b), respectively) fractured at the bond line. This is supported by the presence of flat regions in the cross-sections of the bond interfaces of the 2.5 min and 4 min samples, as shown in Figure 7.14. Although the 5 min sample fractured in the middle of the specimen (see 7.13(c)), the fracture surface shown in Figure 7.14(c) clearly indicates that a strong bond was formed. The 10 min sample (see Figure 7.13 (d)) fractured away from the bond line, indicating the bond was in fact stronger than the material itself; this is supported by the fracture surface shown in Figure 7.14 (d).

The stress strain response of the specimens with 2.5 min and 4 min holding times for the 12.7 MPa pressure (see Figure 7.10) also shows that the bonds formed for these holding times fracture differently from those formed at 5 min and 10 min holding times. In contrast to holding times of 5 min and 10 min where the final stage of fracture is represented by a sudden drop in stress, the 2.5 min and 4 min holding time curves exhibit a final stage of fracture consisting of a gradual, almost linear, decrease of the stress followed by an abrupt reduction to zero. The initial gradual stress drop in these cases is consistent with a fracture sequence consisting of initial separation of the weakly bonded part of the cross-section, followed by fracture of the strongly bonded part, as shown in Figure 7.14.
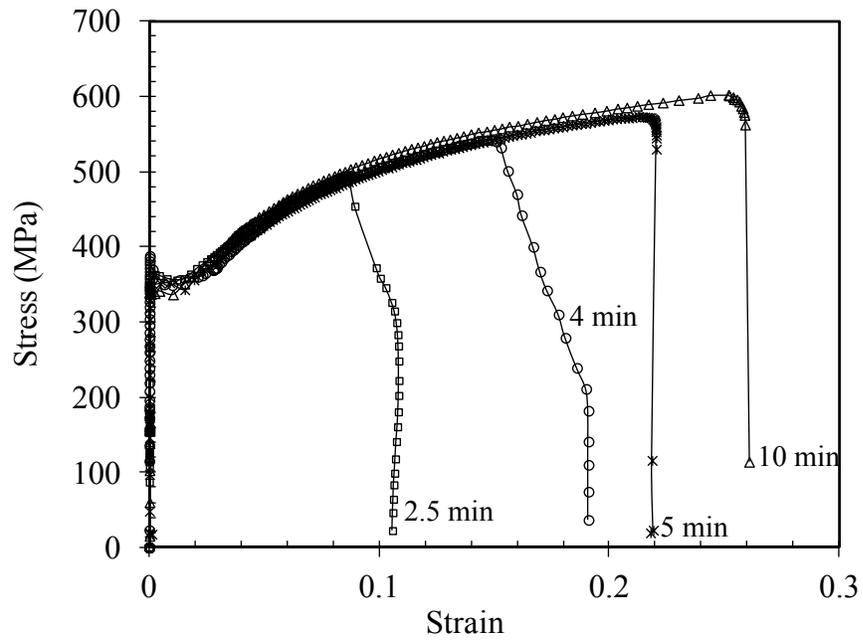
Figure 7.10*- Flow curves obtained for bonds formed at 1000ºC and 12.73 MPa.*
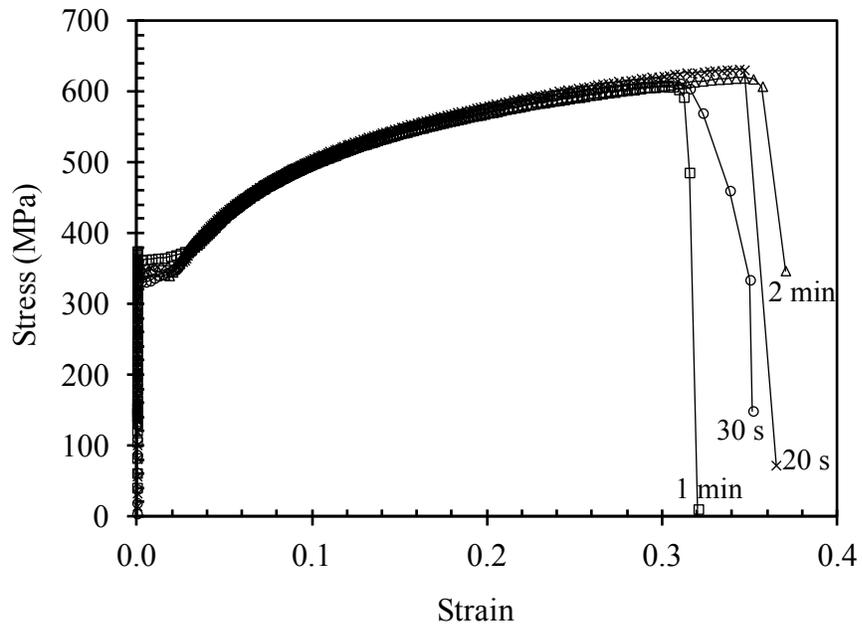
Figure 7.11*- Flow curves obtained for bonds formed at 1000ºC and 38.2 MPa.*

181
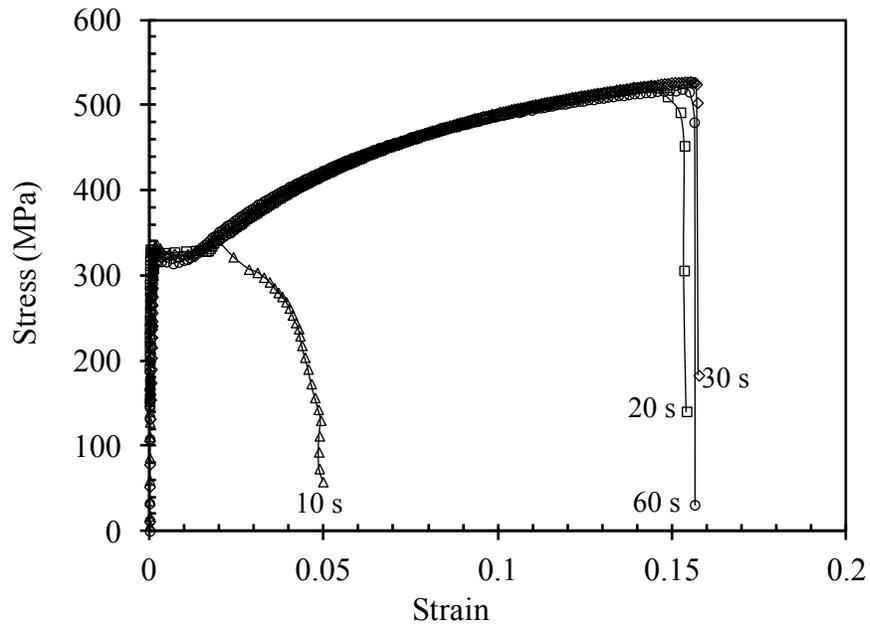
Figure 7.12- Flow curves obtained for bonds formed at 1000^oC and 63.7 MPa.
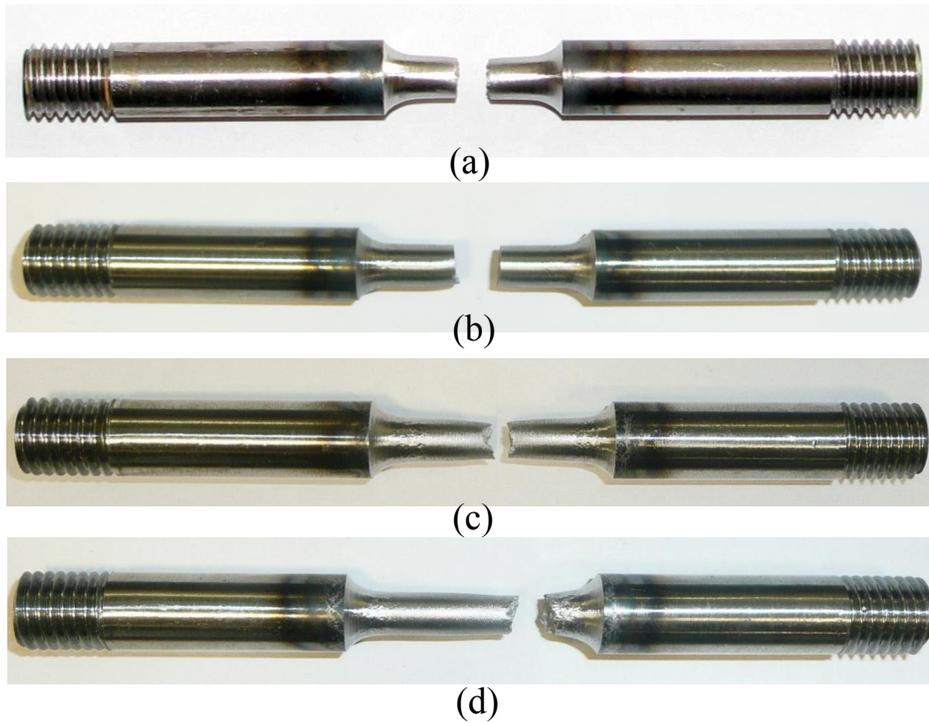
Figure 7.13- Position of necking in samples subjected to 12.7 MPa pressure at $1000°C$ for (a) 2.5 min, (b) 4 min, (c) 5 min, (d) 10 min.
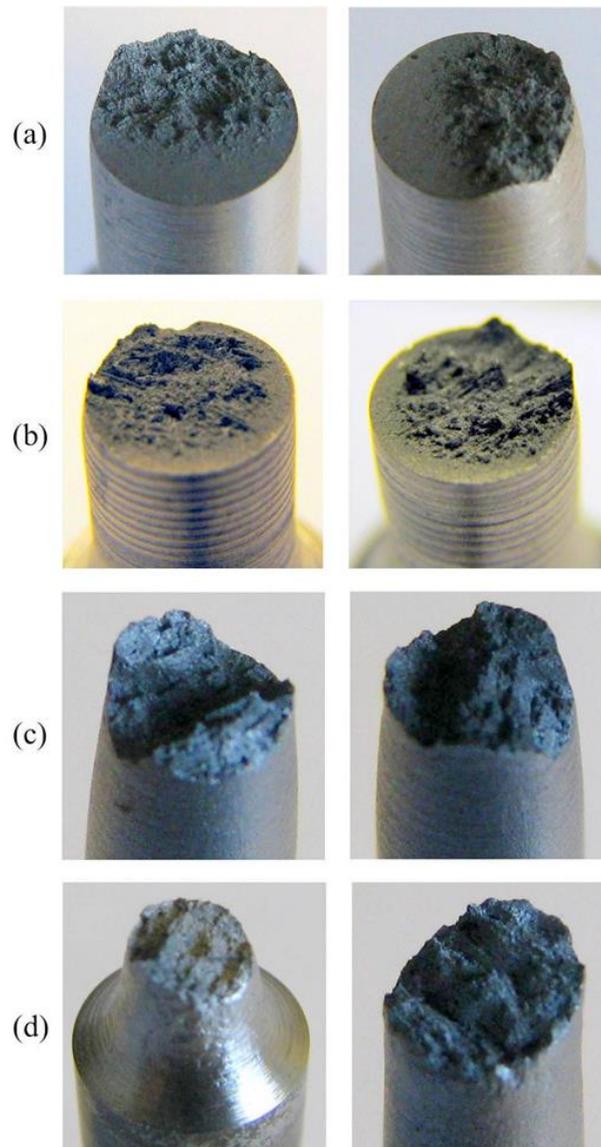
Figure 7.14*- Cross-sections of bonded samples after fracture. Bonds were formed under 12.7 MPa, 1000°C and holding times of (a) 2.5 min, (b) 4.5 min, (c) 5 min, (d) 10 min.*

The failure strains for the 5 min and 10 min samples for the 12.7 MPa pressure are also similar, at least to within the level of scatter expected from testing different samples under the same conditions (this is explored further in Figure 7.15), giving further indication that holding times greater than approximately 5 min are sufficient to form a strong bond at this pressure and temperature.

Figure 7.11 shows that increasing the compressive holding pressure from 12.7 MPa ($\approx 0.42\sigma_y$) to 38.2 MPa ($\approx 1.27\sigma_y$), where σ_y is the yield strength at $1000^{\circ}C$, significantly decreased the holding time required for a strong bond to form. For a pressure of 38.2 MPa, a holding time of 20 s is sufficient to create a strong bond. This can be explained in terms of the plastic deformation that has occurred for the latter case.

As shown in Figure 7.12, increasing the holding pressure further to 63.8 MPa did not significantly alter the required holding time; however, comparing Figure 7.10 and 7.12, a relative decrease in the peak strength can be observed for the higher pressure. The peak strengths of the bonded specimens are listed in Table 7.3, for the ranges of pressures and holding times tested. Table 7.3 shows that for compressive holding pressures below the yield stress, the peak strength of the material is lower than that for a compressive holding pressure approximately equal to the yield stress. It is also evident that increasing the holding pressure to a value well beyond the yield stress caused a decrease in the peak strength of the material. This decrease in peak strength is attributed to the significant amount of barrelling that occurs in these tests under such large compressive pressures, creating damage in the material as a result and therefore softening the material.

Table 7.3- Maximum strengths of bonded specimens.

Load	Holding Time	Max Strength (MPa)
-1 kN ($\approx 0.42\sigma_y$)	5 min	573.55
	10 min	602.19
-3 kN ($\approx 1.27\sigma_y$)	20 s	631.00
	30 s	614.02
	1 min	607.33
	2 min	619.53
-5 kN ($\approx 2.12\sigma_y$)	20 s	522.46
	30 s	527.88
	1 min	518.90

Although a strong bond formed for the 12.7 MPa pressure and 5 min holding time, Figure 7.10 indicates a higher failure strain for the bond formed under 10 min holding time. This can also be seen in Figure 7.11, where for different holding times the material exhibits a range of failure strains with no discernible trend. One possible explanation is the difference in soaking time experienced by the samples. To investigate this, several contiguous cylindrical samples were machined from the same batch of billet surface material and tested using the same procedure as that shown in Figure 7.8. The specimens were heated to 1000$^{\text{o}}$C and a compressive load of 12.7 MPa was applied for holding times of 2.5 min, 5 min and 10 min. The specimens were then machined into dog-bone shapes as in Figure 7.9 and tested in tension to investigate the resulting stress-strain behaviour of each specimen.

Figure 7.15 compares the flow curves obtained from these tensile tests. The highest failure strain is associated with a holding time of 2.5 min, the second highest ductility is exhibited by the 10 min holding time and the 5 min holding time has the minimum ductility. Therefore a relationship between soaking time and ductility

cannot be established. It is more likely that the range of failure strains shown in Figure 7.15 is a result of the natural variability associated with testing different samples taken from the same batch of material.
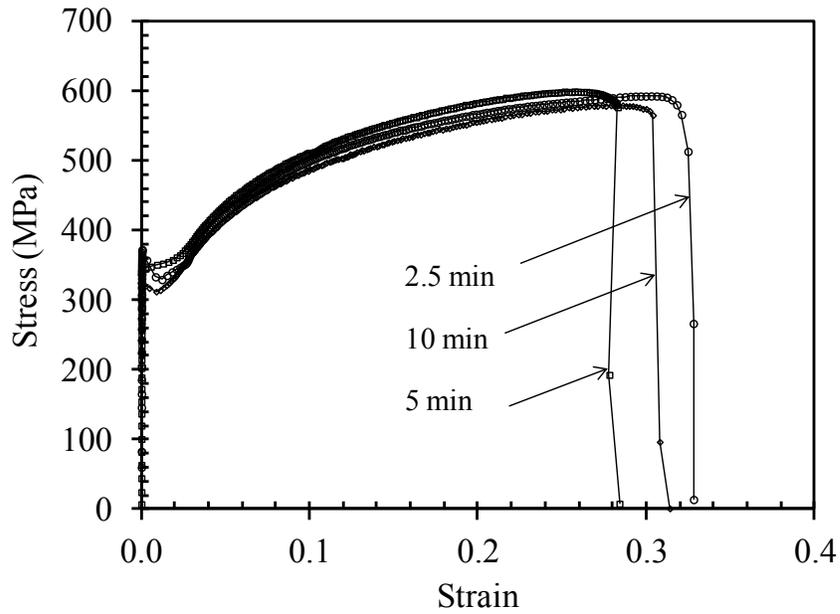
***Figure 7.15**- Flow curves obtained from tensile testing of samples soaked at 1000ºC for different times.*

7.3 SEM images of the bonds

In order to further assess the bonded interfaces, bonded samples were cut longitudinally and images were taken from the bonding zones using a Hitachi S3400N Scanning Electron Microscope (SEM). In order to be able to view the bonding interface the bonded samples were shortened from ends and then were EDM cut through the cross sections. This has been shown in Figure 7.16.
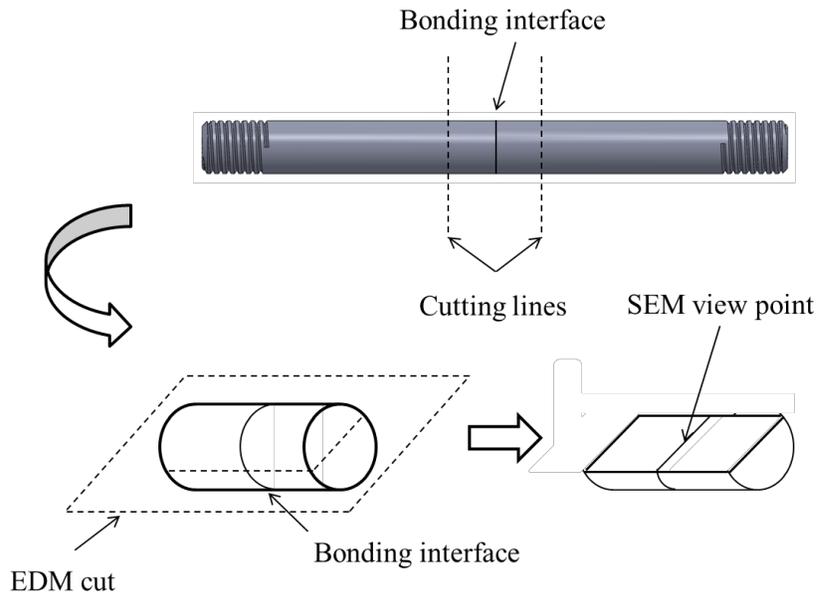
***Figure 7.16**- Preparation of SEM view point for inspection of the bonding interface.*

Figure 7.17 to 7.19 show the bonds formed at 1000°C, under a compressive pressure of 12.37 MPa for holding times of 4 min, 5 min and 10 min respectively. Figure 7.17 shows a crack line at the bond interface. As shown previously for the tensile testing of the 12.37 MPa bonded samples, for a holding time of 4min the bonds were broken under considerably lower stress and strain values, indicating an incomplete bond. The crack line appearing at the bond interface in Figure 7.17 is consistent with this finding.

Increasing the holding time to 5min resulted in better healing of the joint line; as shown in Figure 7.18, the crack was reduced to predominantly a series of separated voids. Increasing the holding time to 10min (Figure 7.19), no trace remained of the bond line confirming a fully bonded sample.
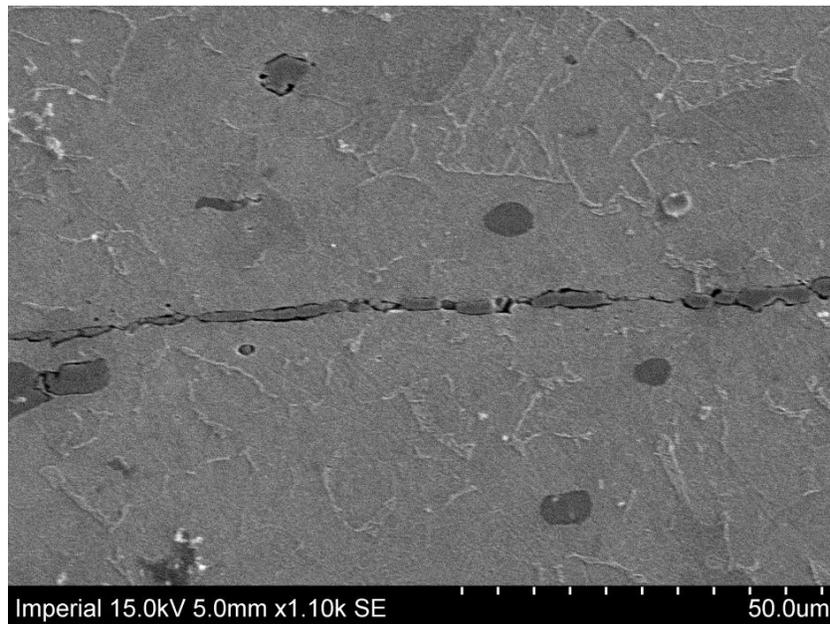
Figure 7.17*- SEM image of the bond line. Bonds were formed under P= 12.37MPa, 1000°C and holding time of 4 min.*
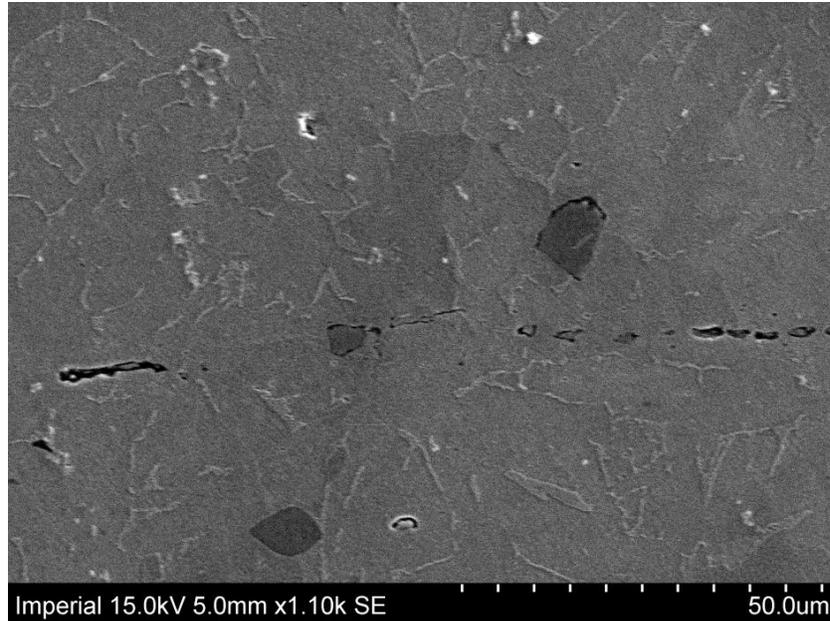
Figure 7.18*- SEM image of the bond line. Bonds were formed under P= 12.37MPa, 1000°C and holding time of 5 min.*

Imperial 15.0kV 5.0mm x1.10k SE   50.0um
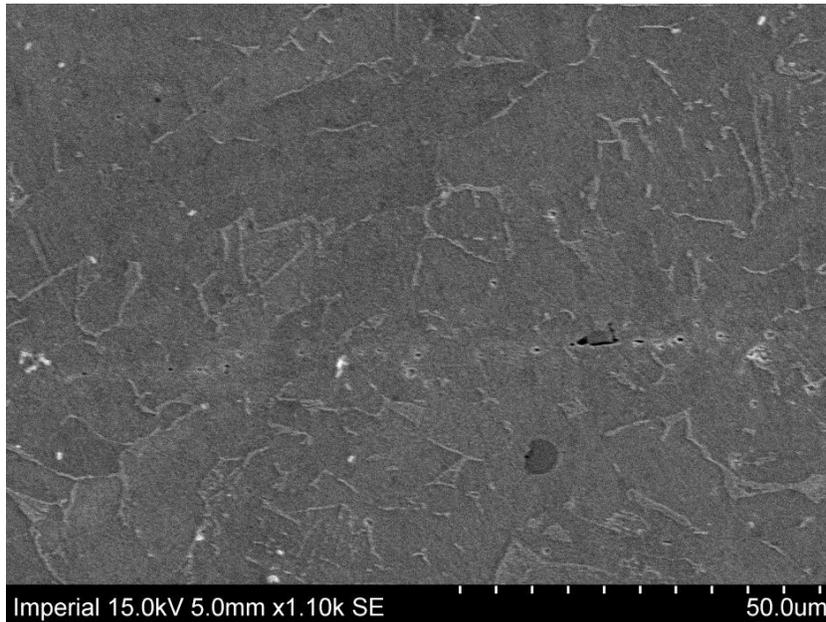
Figure 7.19*- SEM image of the bond line. Bonds were formed under P= 12.37MPa, 1000°C
and holding time of 10 min.*

Similar images were taken of the joint interface for bonds formed at 1000°C, under a
compressive pressure of 63.7 MPa. Figure 7.20 to 7.22 show the bonds formed at
these conditions for holding times of 10s, 20s and 30s respectively. As shown
previously from the tensile testing of the bonded samples, for a holding time of 10s
the bonds were broken under considerably low stress and strain values indicating an
incomplete bond. The crack line appearing at the bond interface in Figure 7.20
confirms this situation. Increasing the holding time to 20s has resulted in better
healing of the bond line. As shown in Figure 7.21 the crack has been reduced to a
series of distanced voids. With increasing the holding time to 30s (Figure 7.22), no
track of the bonding joint can be seen indicating a fully bonded sample.

The SEM images have also been taken of the bonds formed at 1000°C under a
compressive pressure of 38.2 MPa for several holding times. The results have been
shown in Figure 7.23 to 7.25 for holding times of 10s, 20s and 30s respectively.
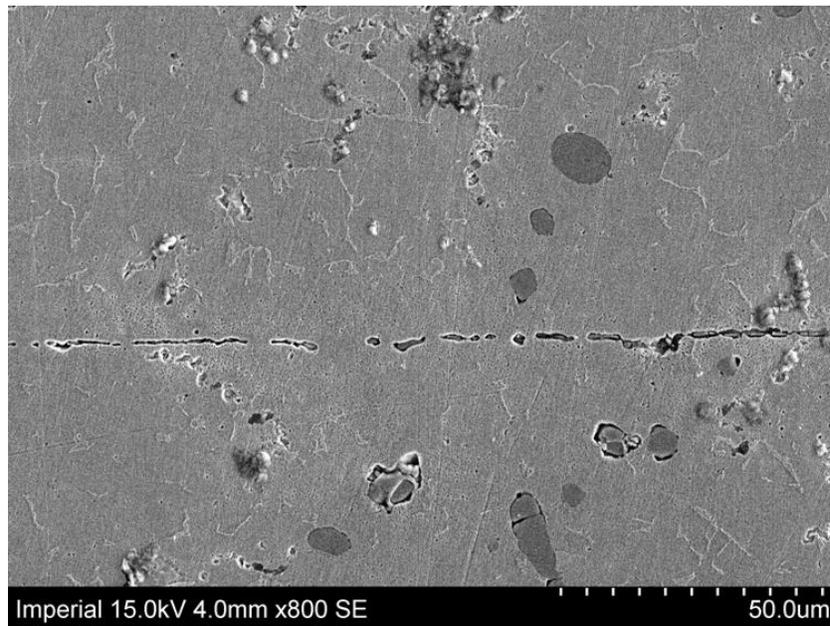
Figure 7.20- *SEM image of the bond line. Bonds were formed under P= 63.7MPa, 1000°C and holding time of 10 s.*

Figure 7.21- *SEM image of the bond line. Bonds were formed under P= 63.7MPa, 1000°C and holding time of 20 s.*

Figure 7.22- *SEM image of the bond line. Bonds were formed under P= 63.7MPa, 1000°C and holding time of 30 s.*
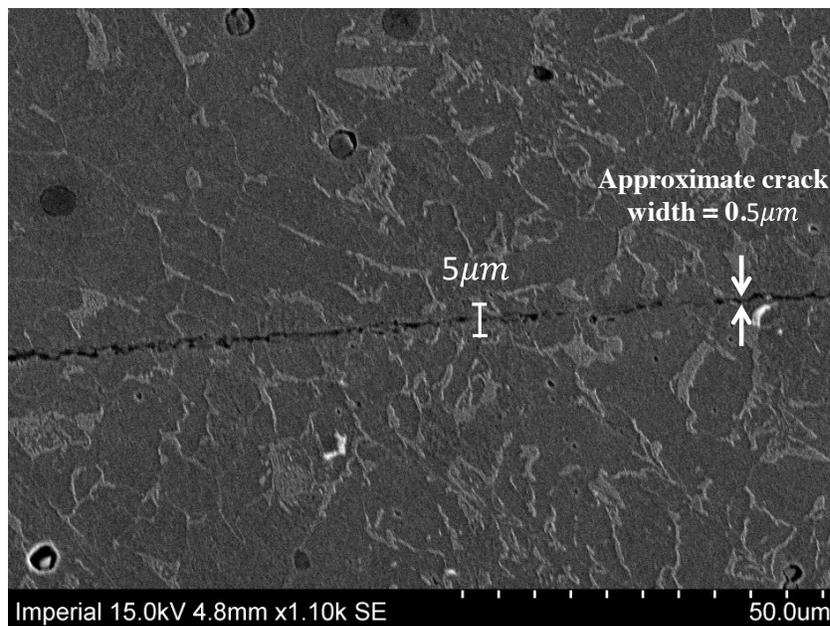
Figure 7.23- *SEM image of the bond line. Bonds were formed under P= 38.2MPa 1000°C and holding time of 10 s.*
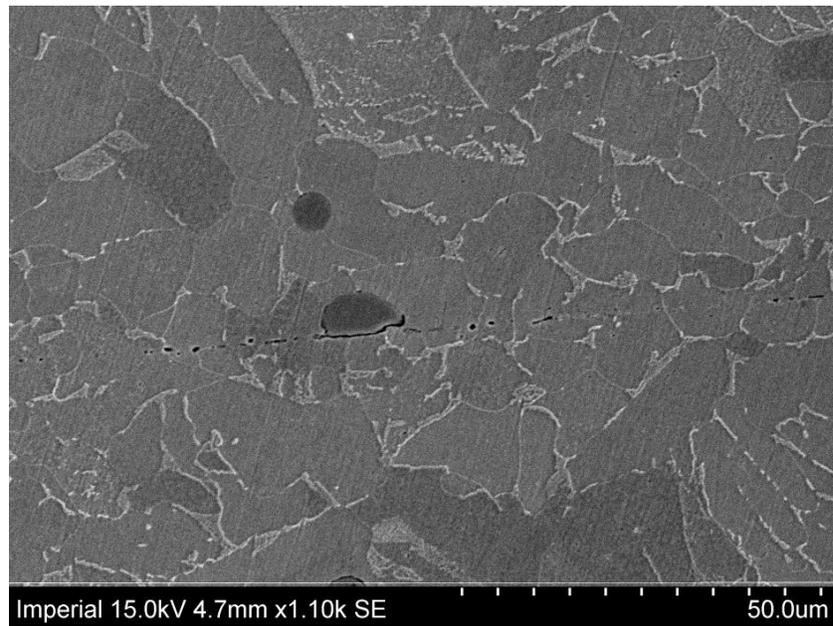
Figure 7.24- *SEM image of the bond line. Bonds were formed under P= 38.2MPa 1000ºC and holding time of 20 s.*
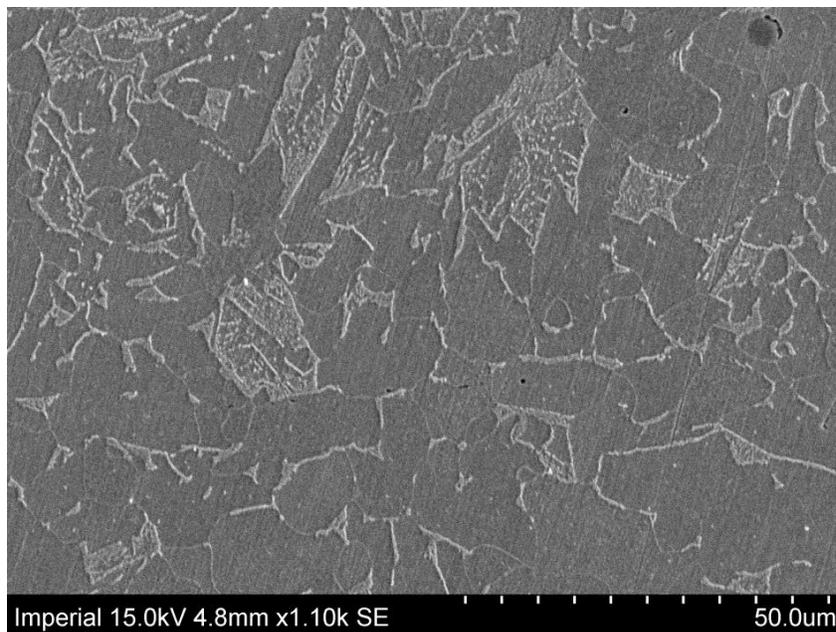
Figure 7.25- *SEM image of the bond line. Bonds were formed under P= 38.2MPa 1000ºC and holding time of 30 s.*

7.4 Model calibration

This section presents a method for calibration of the void elimination model, using the experimental results obtained in section 7.2.6. It is worth mentioning that this calibration process is associated with the healing stage of the void elimination process only (calibration of Pilling model) and does not involve the closure stage.

7.4.1 The concept of *vvf$_c$*

As previously explained, the performed experiments included the diffusion bonding of two cylindrical pieces together at high temperature (1000^oC) under a range of compressive holding loads. Assuming the cylindrical halves are both non-porous, under compressive loading, asperities on the cross sections of the cylindrical halves come into contact. Upon initial contact a series of approximately cylindrical voids form along the bonding surface, the scale of which depends on the roughness. This has been illustrated in Figure 7.26.
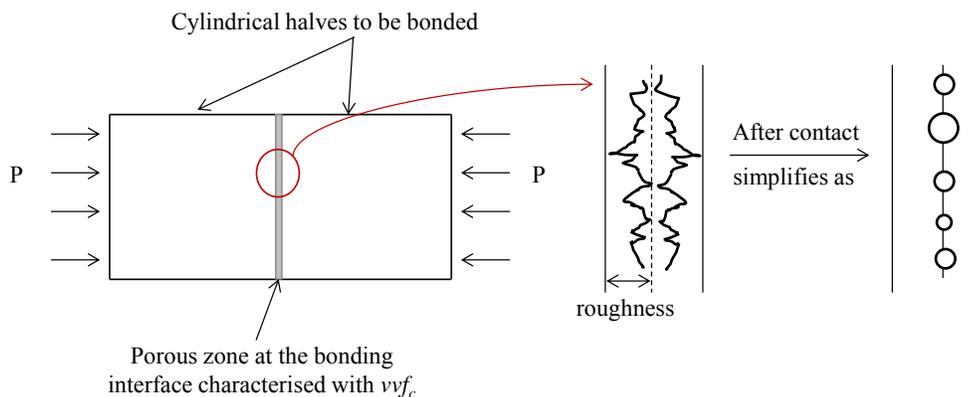
Figure 7.26*- Illustration of the bonding interface and the defined porous zone in the experiments.*

The void volume fraction of the porous zone is characterised by *vvf$_c$*, the void volume fraction at the start of healing:

$$vvf_c = \frac{volume\ of\ the\ voids}{volume\ of\ the\ porous\ zone} \qquad 7.1$$

The porous zone formed upon the contact of asperities on the cross sections is assumed to be approximately a cylinder. The radius of this cylinder is equal to the specimen radius and its height is twice the surface roughness (see Figure 7.26). Therefore, the volume of the porous zone is defined as:

$$volume\ of\ the\ porous\ zone\ =\ \pi\ (specimen\ radius)^2.(2 \times roughness) \qquad 7.2$$

Figure 7.23 shows a SEM image of the bonding line formed under a compressive load of 38.2 MPa at 1000°C and a holding time of 10 s. The total volume of the voids along the bonding line in Figure 7.23 is also assumed to be approximately that of a disc, with a radius equal to the specimen radius and a height equal to the approximate crack width as shown on Figure 7.23. Therefore, the volume of the voids is defined as:

$$volume\ of\ the\ voids = \pi\ (specimen\ radius)^2.crack\ width \qquad 7.3$$

The void volume fraction of the porous zone is therefore defined as:

$$vvf_c\ =\ \frac{crack\ width}{2 \times roughness} \qquad 7.4$$

It was estimated from Figure 7.23 that the width of the crack at the bonding line is approximately 0.5 μm. This will result in a void volume fraction of approximately 0.027 for the bond shown in Figure 7.23 (roughness = 9.1 μm).

It is worth mentioning that the measurement of vvf_c should be done during the phase of load-temperature application. The SEM image shown in Figure 7.23 is associated with a specimen which has been taken out of the Gleeble and therefore is at room temperature and under no compressive load. The measured crack width from this image is therefore larger compared to when the specimen is under compression. Therefore, the void volume fraction calculated using the crack width measured from

the SEM image in Figure 7.23 is larger and is not representative of the actual vvf_c. In addition, the void volume fraction calculated from the SEM image overestimates the total volume of the voids on the bonding interface by approximating it into a disc with a radius equal to the specimen radius and a height equal to the gap width (crack width). This approximation ignores the contact of the asperities and the area already bonded by plastic deformation.

Ideally, vvf_c should be measured while load and temperature are applied. Since porosity is usually concealed within a material, this would require a technique such as in-situ 3D X-ray microscopy. However, currently, it is not possible to distinguish pores in free cutting steel from MnS inclusions, which are much larger than the remnant cylindrical porosity (Kaye et al., 2013) as discussed in Chapter 5. For these reasons, vvf_c is treated as a fitting parameter to be obtained by matching model predictions to experimental data. In the following sections a method will be proposed for finding the vvf_c, where healing times obtained experimentally from section 7.2.6 will be compared against the healing times calculated from the Pilling model for different vvf_c values (considering the applied experimental conditions). Since void volume fraction vvf_c is representative of the amount of porosity in the material, it is clear that a change in the value of vvf_c will affect the predicted healing times. Therefore, the value of vvf_c for which a close agreement between experimental and model predictions of healing time is observed will be selected as the calibrated value. This will be explained in more detail in section 7.4.4. Prior to this, a brief description of the numerical aspects of healing time calculation is given in section 7.4.2 and 7.4.3.

7.4.2 The sensitivity of the healing time to the integration procedure

To obtain the healing time from Pilling model, equation 7.5 which is the inverse of Pilling's diffusion bonding rate equation (see Chapter 3), was integrated using the trapezoidal rule.

$$g = \frac{dt}{df_h} = \left[-2\dot{\varepsilon}_r(1 - f_h) \right.$$

$$\left. - N_i \left(\frac{2}{r_o^2 h_o f_h} \frac{D_{gb}\delta\Omega\sigma_z}{KT} \frac{1 - f_h}{\ln(1/f_h) - (1 - f_h)/2}\right)\right]^{-1}$$

$$\qquad 7.5$$

The parameters in equation 7.5 have been previously explained in Chapter 3.

The trapezoidal integration method approximates an integral from a to b by approximating the area under a curve using trapezoids. This is shown mathematically in equation 7.6, where N is the number of intervals.

$$\int_{a=vvf_c}^{b=0} t\, df_h \approx \frac{\Delta f_h}{2}\left[g(f_{h_0}) + 2g(f_{h_1}) + 2g(f_{h_2}) + \cdots + 2g(f_{h_{N-1}})\right.$$

$$\left. + g(f_{h_N})\right] \qquad 7.6$$

$$\Delta f_h = \frac{b - a}{N}, t_i = a + i\Delta f_h$$

It is clear from Equation 7.5 that dt/df_h is an undefined function at $f_h = 0$, therefore, to calculate the total healing time, b should be set to a very small value e.g. 1E-18, instead of absolute zero. The effect of this substitution on the accuracy of healing time will be very insignificant. This is because: $\lim_{f_h \to 0} \frac{dt}{df_h} = 0$ (see Appendix H), indicating that healing time versus f_h reaches a plateau when f_h tends to zero.

The accuracy of the numerical integration process can be affected by the magnitude of the subdivisions (Δf_h) used. The sensitivity of the calculated healing times (using this integration method) to the number of subdivisions has been studied. Figure 7.27 compares the calculated healing times versus f_h for three values of N. It is clear from this figure that smaller values of N will result in slightly smaller healing time predictions.
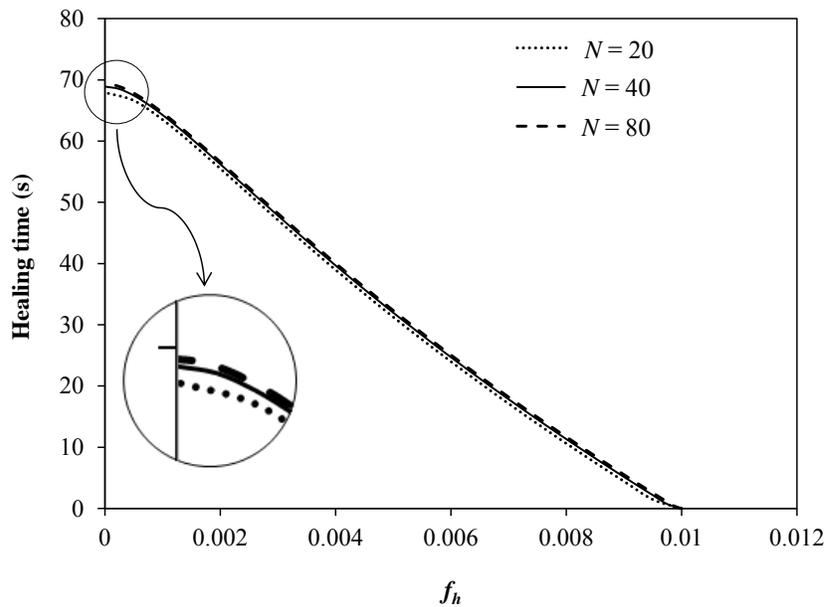
Figure 7.27- *Sensitivity of the numerically calculated healing time to the number of integration subdivisions, N. The healing times were calculated for 1000°C and load of 38.2 MPa.*

For further investigation, two larger values of N were studied ($N = 180$ and $N = 320$). The calculated healing times and the percentage of healing time difference (%Δ) relative to the healing time corresponding to the largest N ($N = 320$) are shown in Figure 7.28 for all studied values of N. Although larger values of N will result in better accuracy, it is clear from Figure 7.28 that the %Δ is insignificant. It is worth mentioning that high values of N will result in longer computational times. Therefore, the value of N can be chosen so that it satisfies accuracy and computational efficiency requirements. This study uses $N = 80$ for calculation of healing time.
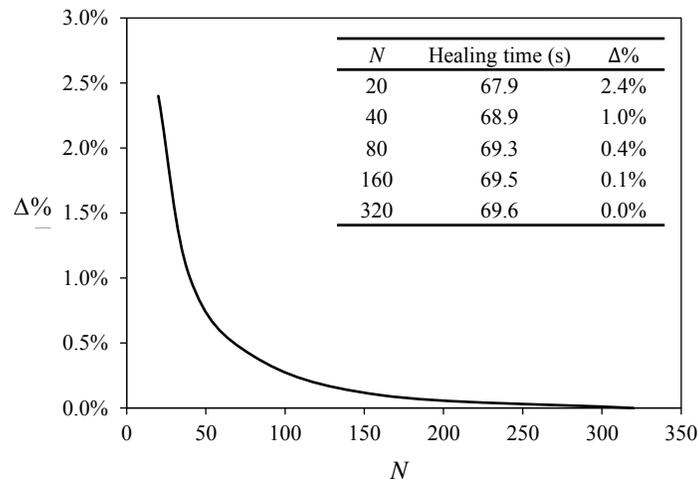
Figure 7.28 *- Percentage healing time difference (%Δ) relative to the healing time corresponding to the largest N (N= 320) for vvf$_c$ = 0.01 and load of 38.2 MPa.*

7.4.3 Methods of healing time calculation

To calculate the healing time from the Pilling model for the considered experimental conditions, two approaches were used. Initially, the experiment was simulated using ABAQUS/Standard version 6.10-2. A model of a quarter cylinder with symmetry boundary conditions on the flat faces was created, representing one of the two cylindrical halves in the experiment. Plane 'cdef' is symmetric relative to the *x* axis. Plane 'abcf' is symmetric relative to the *y* axis and plane 'bcd' is symmetric relative to the *z* axis. The load was applied as a uniaxial pressure to plane 'aef', increasing linearly with time as ramp (pressures of 12.7MPa, 38.2MPa and 63.7MPa were tried). The cylinder was partitioned to include a thin section (with thickness of 0.009mm) on the cut end, shown in blue in Figure 7.29. This thin section represents the bonding interface; the developed UMAT was applied to this section only (with elastic-plastic properties of free cutting steel at 1000°C and $\dot{\varepsilon}$ (see Appendix A) elsewhere equivalent to the GT material with *vvf* = 0). A mesh with 8-node linear brick elements (C3D8 in ABAQUS terminology) was used.

As previously described in Chapter 3, the Pilling diffusion bonding rate equation is a function of hydrostatic pressure. The hydrostatic pressure distribution at the bonding

interface was obtained from the created ABAQUS model. This pressure distribution for three uniaxial loads has been shown in Figure 7.30. Figure 7.30a shows a uniform hydrostatic pressure distribution for a uniaxial load of 12.7 MPa. Figure 7.30b and 7.29c show the hydrostatic pressure distribution for a uniaxial load of 38.2 MPa and 63.7 MPa, respectively where the distribution is almost uniform with maximum and minimum values of 13.1 MPa and 12.9 MPa for the uniaxial load of 38.2 MPa and maximum and minimum values of 21.7 MPa and 21.5 MPa for the uniaxial load of 63.7 MPa. It is worth mentioning that the ABAQUS UMAT calculates the healing time for each element based on the hydrostatic pressure of that element. For non-uniform pressure distribution, the healing times will differ for each element (elements which experience higher pressure have smaller healing times). Therefore, maximum and minimum values of hydrostatic pressure correspond to smallest and largest healing times, respectively. The smallest and largest healing times and the corresponding hydrostatic pressures for the considered case have been summarised in Table 7.4. It is clear from this table that the difference in the predicted healing times for maximum and minimum values of hydrostatic pressure is insignificant and therefore a uniform hydrostatic pressure distribution can be assumed and a single healing time can be calculated for all elements.
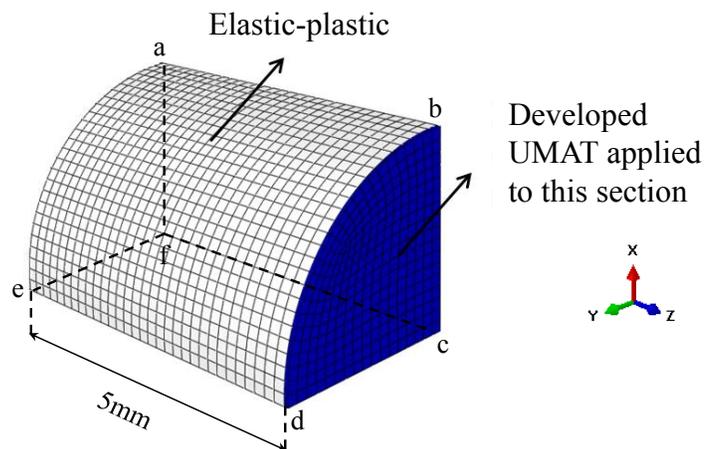
Figure 7.29*- A quarter cylinder, representing one of the two cylindrical halves with a thin porous section on the cut end, representing the bonding surface.*
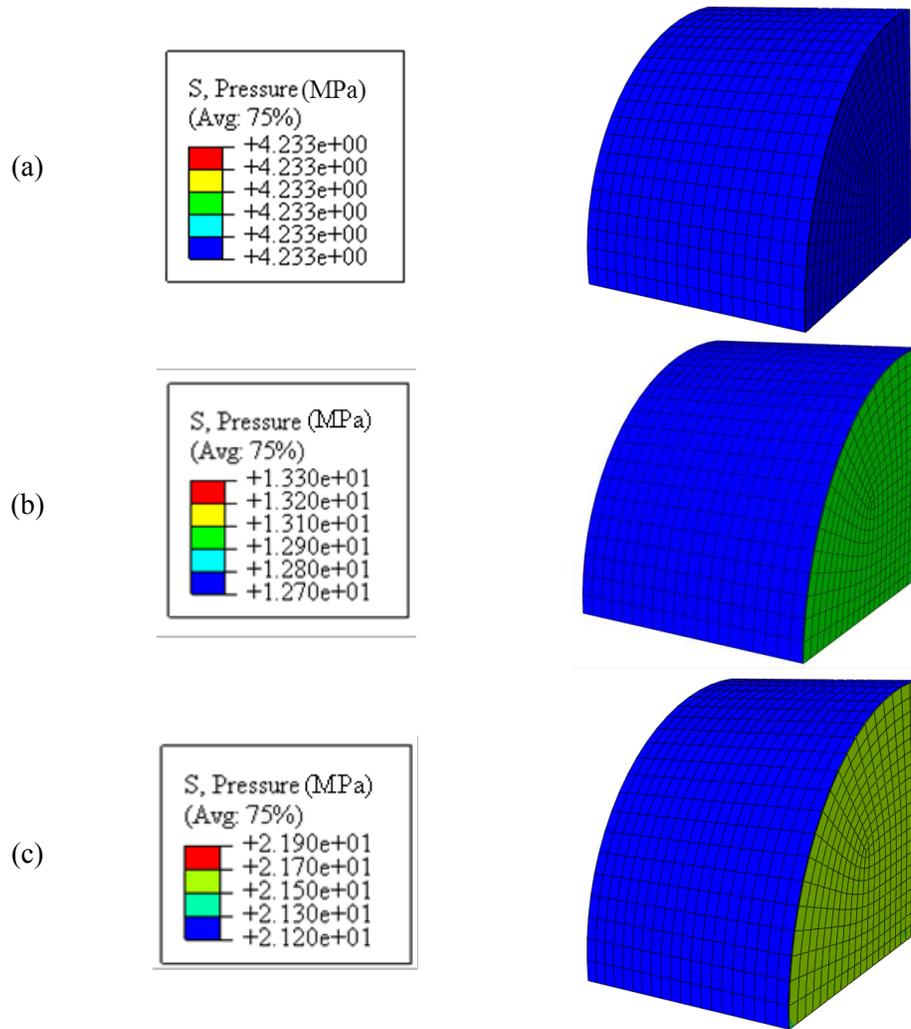
Figure 7.30- Hydrostatic pressure distribution for uniaxial loads of (a) 12.7 MPa, (b) 38.2 MPa and (c) 63.7 MPa.

Table 7.4- Calculated healing times using ABAQUS model for $vvf_c = 0.01$.

Uniaxial Load (MPa)	Max and Min values of hydrostatic pressure (MPa)	Min and Max Healing times(s)
12.7	4.23	178.4
38.2	13.1	67.2
	12.9	68.2
63.7	21.7	39.4
	21.5	39.8

As the second approach and to be computationally efficient, the healing times can also be calculated numerically directly from Pilling model for the same experimental boundary conditions. To do so equation 7.5 was integrated numerically using trapezoidal method in Microsoft Excel 2010. A temperature of 1000^oC (the temperature at which experiments were performed) was used and the hydrostatic pressure was assumed to have a uniform distribution on the specimen cross section at the bonding interface. The magnitude of the hydrostatic pressure was taken to be a third of the applied uniaxial load. For $vvf_c = 0.01$, the healing times calculated using this approach are compared against the maximum and minimum healing times obtained from ABAQUS (Table 7.4). This comparison is shown in Figure 7.31, where a very good match is observed between the results obtained from both approaches. Since the calibration process involves investigating a range of parameters (different vvf_c and loads), the numerical approach has been preferred for calculation of the healing time to save computational time.
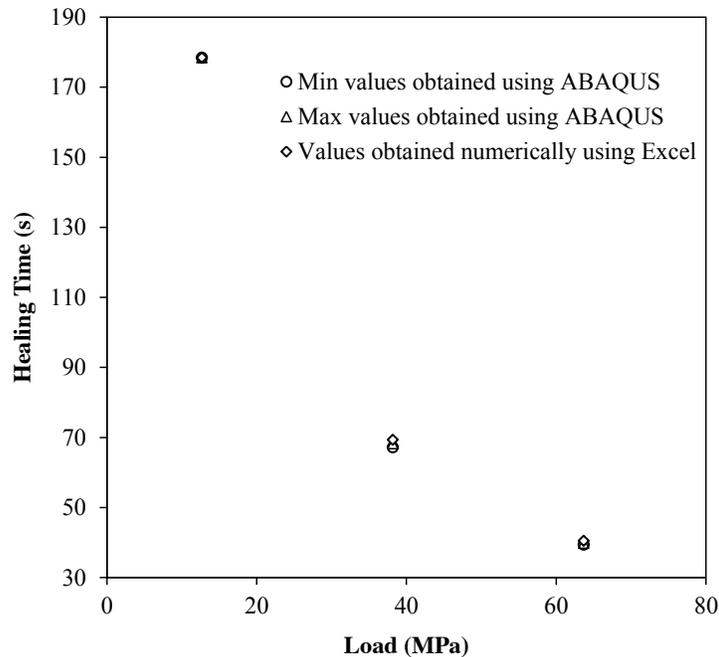
Figure 7.31- Comparison of the healing times calculated numerically using Excel against the maximum and minimum healing times obtained from ABAQUS.

7.4.4 Calibration procedure

To identify the representative vvf_c, the healing time was calculated from the Pilling model for different values of vvf_c at 1000°C and under three loading conditions (63.7 MPa, 38.2 MPa and 12.7 MPa). Figure 7.32 shows the calculated healing times under different loading conditions for a range of vvf_c.

As previously shown in Figure 7.12, the healing time obtained experimentally for a holding load of 63.7 MPa is between 10 and 20 s. The results in Figure 7.30 show that for $vvf_c = 0.003$, the model predicts a healing time of 13 s, which is in accord with the experimental data. The calibration factor vvf_c is therefore taken to be 0.003 for this condition. As previously shown in Figure 7.11, the healing time obtained experimentally for 38.2 MPa was approximately 20 s. The calibrated model ($vvf_c =$ 0.003) predicts a healing time of 21 s for this pressure, which is in close agreement with the experiments.

For a pressure of 12.7 MPa, the calibrated model ($vvf_c = 0.003$) predicts a healing time of 50.7 s. However, the healing time previously obtained from the experiments for this loading condition was greater than 300 s, indicating the calibration is not valid for this loading condition. This establishes a range of pressure, approximately 35 to 65 MPa ($\approx \sigma_y$ to $2\sigma_y$), for which the calibrated value of $vvf_c = 0.003$ is valid at 1000°C. Conditions involving pressures outside this range, or significantly different temperatures, would require additional calibration.

It should be noted that the obtained vvf_c here is almost 10 times smaller than the value (0.027) calculated in section 7.4.1 from the SEM image shown in Figure 7.23. As previously discussed in section 7.4.1, Figure 7.23 shows the SEM image of the bonding interface for a specimen at room temperature and under no compression. It is therefore reasonable for the calculated vvf_c from the SEM image in Figure 7.23 to be larger than the obtained value here. It is difficult to further assess the accuracy of the calibrated vvf_c value with experimental data available.
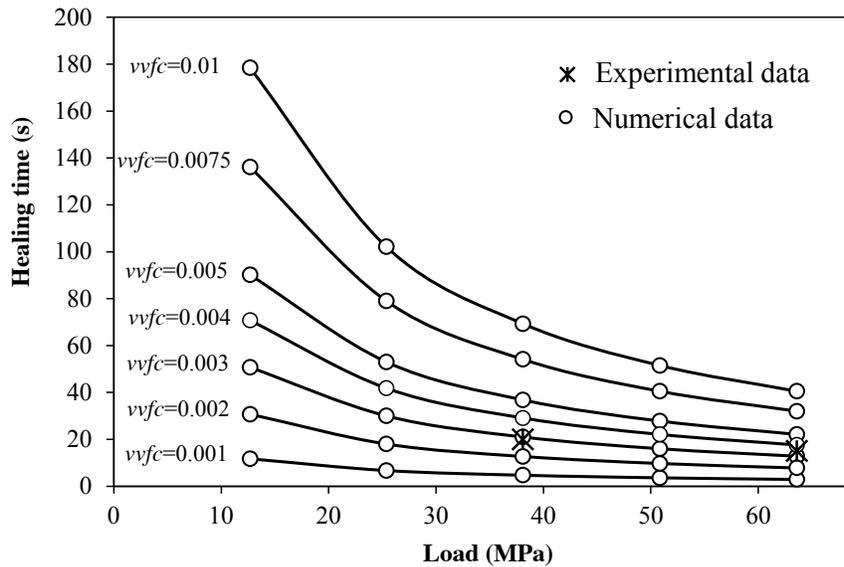
Figure 7.32 *- Comparison of the healing times obtained experimentally with healing times calculated from the Pilling model at different loads and for different values of vvf$_c$.*

7.4.5 Sensitivity to roughness

Any parameter which influences the healing time prediction can directly affect the calibration of the *vvf$_c$*. This section investigates the influence of variation of the absolute scale of the surface roughness on the bonding time. To study the effect of surface roughness, a range of *vvf$_c$* values were considered and the corresponding healing times were calculated numerically (using Microsoft Excel) for several surface roughnesses. Figure 7.33a to 7.33c show the effect of surface roughness on healing time at different loads for three *vvf$_c$* values of 0.001, 0.005 and 0.01. It is evident from this figure that surface roughness significantly affects the predicted healing time. For rougher surfaces the required healing time is longer. This is especially more pronounced at lower loads.

(a) *vvf$_c$* = 0.001

(b) *vvf$_c$* = 0.005

(c) *vvf$_c$* = 0.01
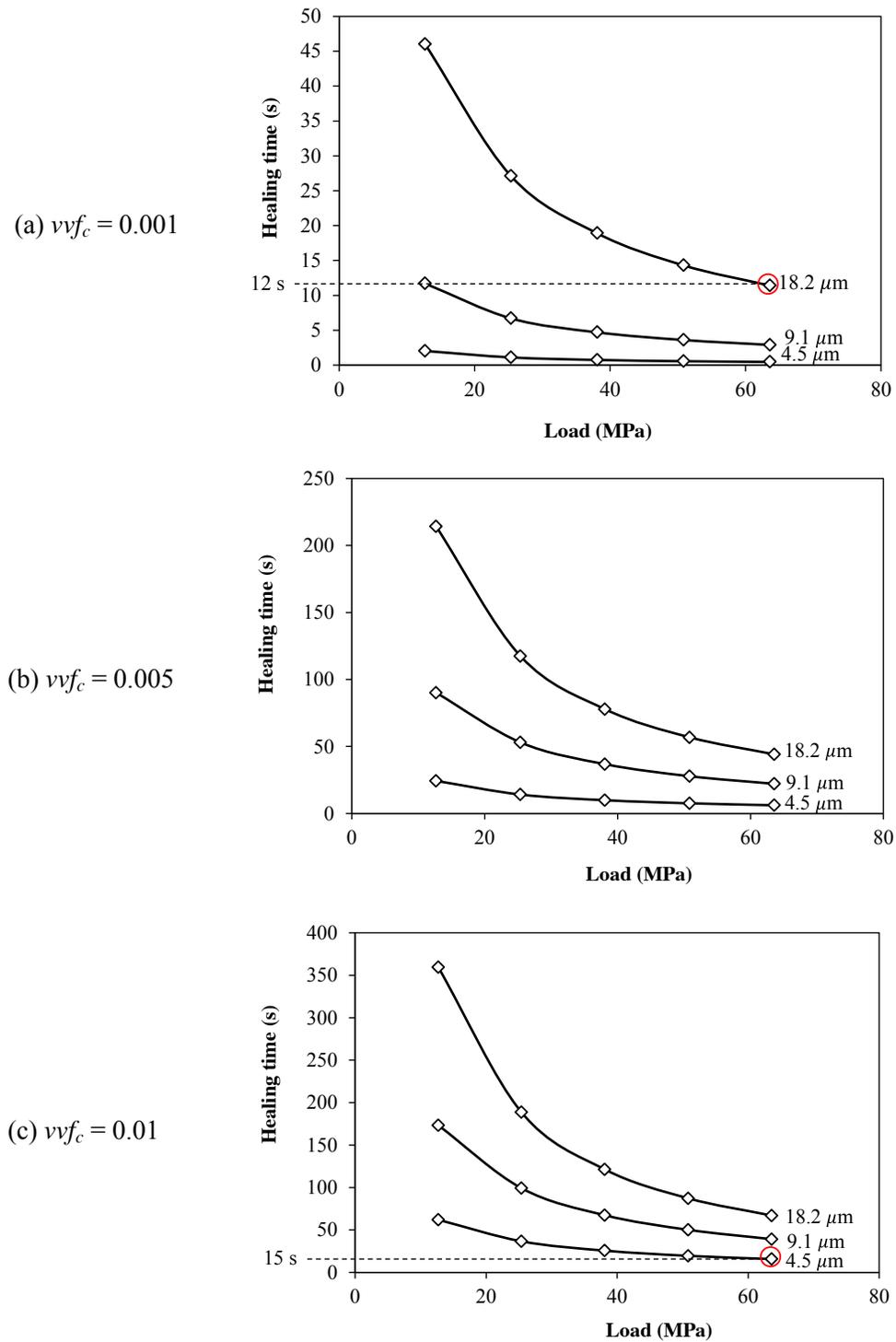
Figure 7.33- The effect of surface roughness on healing time at different loads for three vvf$_c$ values of (a) 0.001, (b) 0.005 and (c) 0.01.

For better understanding of the effect of surface roughness on the calibration of vvf_c, an example has been outlined on Figure 7.33. A point has been marked on Figure 7.33a representing a healing time of approximately 12 s for the roughness of 18.2 μm and vvf_c = 0.001 at a load of 63.7 MPa and temperature of 1000°C. Another point has been marked on Figure 7.33c for the same load and temperature conditions, showing a healing time of approximately 15 s for the roughness of 4.5 μm and vvf_c = 0.01. Let's assume the healing time obtained experimentally for this load-temperature condition is around 12 to 15 s. Therefore, depending on the surface roughness, the value of vvf_c could be taken as 0.001 or 0.01. This shows the significant effect of surface roughness on the calibration of vvf_c.

7.4.6 Application of the calibration method for TATA steel

The process of hot rolling of steel is usually performed at a temperature range of 900-1200°C. Semi-finished products such as blooms, billets and slabs are first heated in a furnace until they are red hot. The first stage of rolling involves passing the large semi-finished product backwards and forwards through a roughing stand. A stand consists of a set of steel rolls on which pressure is applied to compress the hot steel passing through them. After leaving the roughing stand the products are continuously passed through a series of finishing stands which compress the steel and cause further thickness reduction. Depending on the thickness reduction and the final shape required, different stands are used and therefore the applied compressive load varies.

The proposed calibration method in this study investigates only one temperature (1000°C) and three compressive loads of -1 kN, -3 kN and -5 kN. These values were selected to cover a wide range of loading conditions reproducing stresses below, approximately equal to and above the yield stress of the material. Application of compressive loads higher than -5 kN was not experimentally possible due to

significant barrelling and collapse of the specimen. It should be noted that in the performed experiments in this chapter, the compressive load is uniaxial and is not exactly identical to that in the rolling process. In addition, the specimens have cylindrical shape and are different to semi-finished products in shape and size. These are the imitations of the calibration process and may provide an approximate estimation for the rolling process. In order to increase the accuracy of the calibration procedure for rolling purposes, it is recommended that the proposed calibration strategy be applied to real rolling cases. The following steps are suggested to achieve this objective:

1) The average pore surface roughness should be measured for a work-piece (billet, bloom, etc.). This is required for calculation of the healing time.

2) The rolling should be performed for several load, temperature and time combinations. The load parameter can be controlled by using different roll stands and setups (single, double, etc.) resulting in different thickness reductions. The time parameter can be controlled by performing the rolling process in furnace rooms and using very slow roll speed.

3) The extent of healing of the rolled products can be investigated using SEM and X-ray tomography. In addition, tensile tests can be performed for different sections of the rolled product to study the strength and ductility and to ensure complete healing.

4) Following step 2 and 3, combinations of load, temperature and time leading to healing can be obtained. These are considered as experimental data.

5) The healing times for each load-temperature combination in step 4 can be predicted using the Pilling model for different vvf_c values for the specified pore surface roughness obtained in step 1.

6) Finally the predicted healing times in step 5 can be compared with the experimental data obtained in step 4. The vvf_c for which a good agreement

between predicted and experimentally obtained healing times is seen can be taken as the calibrated vvf_c.

7.5 Limitations of the approach in rolling applications

As mentioned in Chapter 2, diffusion bonding is a slow process. If the applied load and temperature are low, the time required for diffusion bonding to occur could be very long. As shown in chapter 6, for a single roll pass, when $H_m/L = 3.5$, $f_o = 0.006$, $T = 900^oC$ and roll speed $= 0.1288$ rad/s (see Chapter 6, section 6.1.3 for further detail), the average healing time predicted by the model was 18.1s. For the applied roll speed of 0.1288 rad/s, different locations of the work-piece experience the compression for about 1 to 2 seconds. Therefore, it is clear that the healing cannot be achieved within a single roll pass. Considering the fact that rolling (roughing stage in the studied case) involves multiple passes (see section 6.1.2), healing might be possible if a high number of passes is used. However, the number of passes is limited by the desired thickness reduction.

In cases where the applied conditions (e.g H_m/L, f_o, vvf_c, temperature, etc.) result in longer healing times (e.g. 2-3 minutes), reducing the roll rotation speed and increasing the number of roll stands would favour the healing process, but this would reduce output and add cost. Furthermore, this would cause heat loss in the work-piece. Apart from performing the rolling process in furnace rooms, it may not be possible to heal under these conditions.

7.6 Conclusions

An experimental technique has been developed to identify the conditions (temperature, pressure, time) required for healing in Free Cutting Steel (FCS).

Different combinations of load and time were tested and optimum conditions have been obtained for $1000^{o}C$.

Different analysis techniques have been used to examine the extent of healing. The position of fracture on the specimen and also the cross section of the fracture surface have been inspected. Tensile tests on the bonded specimens have been carried out to measure the strength of the bonded region.

The effect of soaking time on the failure strain has been studied for fully bonded samples. No particular trend in failure strain with soaking time was observed. The variation in failures strain is attributed to natural specimen variability.

It was observed that maximum strength is achieved for holding pressures approximately equal to the yield stress, σ_y. This is attributed to lower pressures being insufficient for strain hardening to occur, and larger pressures causing significant barrelling and subsequent damage in the material.

The experimental results have been used to calibrate a previously developed void elimination model, with $vvf_c = 0.003$ determined to be valid for the pressure range 35 to 65 MPa ($\approx \sigma_y$ to $2\sigma_y$) at $1000^{o}C$.

The obtained results may only provide an approximate estimation for the rolling process. In order to increase the accuracy of the calibration procedure for rolling purposes, it is recommended that the proposed calibration strategy be applied to real rolling cases. A procedure has been suggested to achieve this objective.

7.7 References

Blaiszik, B. J., Kramer, S. L. B., Olugebefola, S. C., Moore, J. S., Sottos, N. R. & White, S. R. 2010. Self-healing polymers and composites. *Annual Review of Materials Research*, 40, 179–211.

Fratzl, P. & Weinkamer, R. 2008. Hierarchical structure and repair of bone: deformation, remodelling, healing. *Self Healing Materials*. Springer Netherlands.

Kaye, M., Puncreobutr, C., Lee, P. D., Balint, D. S., Connolley, T., Farrugia, D., and Lin, J. (2013) Acta Materialia, 10.1016/j.actamat.2013.08.065 (Accepted for publication).

Li, V. C. & Yang, E. 2008. Self healing in concrete materials. *Self Healing Materials*. Springer Netherlands.

Lumley, R. N. & Polmear, I. J. 2007. Advances in self healing of metals. *In:* ZWAAG, S. (ed.) *Self Healing Materials: An Alternative Approach to 20 Centuries of Materials Science*. Dordrecht: Springer Series in Materials Science.

Reinhardt, H. W. & Jooss, M. 2003. Permeability and self-healing of cracked concrete as a function of temperature and crack width. *Cement and Concrete Research*, 33, 981-985.

Vermolen, F., Rossum, M., Perez, E. & Adam, J. 2008. Modeling of self healing of skin tissue *Self Healing Materials*. Springer Netherlands.

Wool, R. P. & O'connor, K. M. 1981. A theory crack healing in polymers. *Journal of Applied Physics*, 52, 5953-5963.

Chapter 8

Conclusions and future work

8.1 Overall conclusions

This research has focused on developing a realistic damage healing model that can predict the extent of porosity eliminated under temperature, time and load conditions used for hot rolling applications. This has required a combination of high quality experimental data and observations as well as robust physically-based models. The project's conclusions have been summarised in detail below:

- The dependency of void closure on stress triaxiality was studied and closure load versus triaxiality plots were obtained from an RVE model (a cylinder of material with a central spherical hole constructed using ABAQUS). To investigate the closure of a void at a specific position in a steel billet, the stress state at that point can be identified and triaxiality obtained by FE modelling of the rolling process. The conditions for plastic closure of a void in that position could therefore be predicted from the obtained closure load - triaxiality plots. The effects of temperature and strain rate were investigated and the results showed that for high temperatures and low strain rates, less pressure would be required for void closure.

- The applicability of a reverse Gurson model for void closure was examined. Finite element analyses were performed on a representative material volume (RMV) containing a spherical void and an equivalent cell of Gurson-Tevergaard (GT) material. The numerical results of the voided RMV and the GT cell were compared and an optimization method was used to adjust the Gurson model's parameters. The effect of triaxiality on the void closure process was also investigated and the results showed that the Gurson model was in better agreement with the voided cell model for higher triaxialities. This could be explained by the fact that Gurson assumes a spherical void shape which is more correct at high triaxialities.

- An Abaqus/UMAT subroutine was developed for the analysis of the material porosity elimination process including two stages of void closure and healing. The model used GT under reverse loading to predict void closure. Pilling's healing model was implemented into the UMAT to predict the healing time. However the healing stage can be adapted to any other diffusion bonding/healing model. The developed UMAT was compared against the GT model and the developed UMAT and the results showed an increase in material strength as a result of porosity elimination. It was also shown that the healing model was significantly affected by vvf_c. Larger values of vvf_c, implied coarser void surfaces and therefore resulted in longer healing times. It was shown that for two different vvf_c values, the difference in closure times was insignificant due to plastic deformation whereas the healing times varied noticeably.

- The developed UMAT was applied to simulate the void elimination of a steel bloom during rolling. The effect of Roll Gape shape Factor (RGF) on porosity elimination during rolling was investigated. It was shown that for $H_m/L > 4.8$ the closure percentages were lower compared to cases of $1.8 < H_m/L < 4$. This was due to all principal stresses being tensile (representing a regime of tensile triaxiality) for cases of $H_m/L > 4.8$ compared to cases of $1.8 < H_m/L < 4$ where the vertical principal stress becomes compressive and porosity closure occurs.

- The effect of initial void volume fraction on porosity elimination in rolling was also investigated. It was observed that for larger initial void volume fractions the closure percentage was lower. This was due to the amount of compression for larger values of vvf_o not being sufficient at some locations in the work piece to reach the closure state.

- Non-uniform initial porosity distributions in a bloom were modelled. Instead of defining a constant value for vvf_o (which creates an even distribution in the

bloom), *vvf$_o$* was defined as a function of spatial coordinates. Two cases of linear and normal distribution were investigated. To model any distribution the user has to only replace the distribution function. This makes the model approach very versatile and adjustable to other applications with different distributions.

- An experimental technique was developed to estimate the conditions required for healing of Free Cutting Steel (FCS). Different combinations of load and time were tested and optimum conditions were obtained. The relationship between the bond parameters and tensile strength of the joints was inspected. It was seen that for compressive holding loads below the yield point the maximum strength of the material was lower than that for a compressive holding load around the yield point. It was also seen that increasing the holding load to a value well beyond the yield point causes a decrease in the maximum strength of the material. This was explained in terms of the large amount of barrelling that occurred under large compressive loads, damaging the sample and therefore softening the material. The experimental results have been used to calibrate a void elimination model previously developed by the author.

8.2 Suggestions for future work

- The experiments performed in this project have investigated the healing times under three loading conditions. To improve the accuracy and reliability of the healing time predictions of the developed model, the experiments can be performed for several more load-holding time combinations. The effect of temperature is also suggested to be investigated by performing the tests at different temperatures within the range of temperatures used in the rolling application. However these tests will require a large number of trial and error

attempts. The recalibration of the model can therefore be done based on the obtained results.

- Surface roughness has a significant influence in the healing process. To investigate this effect and include it in the developed model, specimens of different surface roughness can be tested under the same procedure explained in Chapter 7 and the healing times can be obtained.

- It was observed in the previous chapter that the healing was better at the centre of the bond interface compared to the edges. This was due to a more compressive triaxiality at the specimen centre. It is therefore worth investigating the effect of triaxiality on the healing process. To investigate the effect of void shape on the healing process, healing of artificially made voids of different shapes can be performed. To eliminate other influencing factors and inspect the shape effect only, other parameters such as temperature, holding pressure, void surface roughness, triaxiality and void size should be kept constant for each void shape. This requires the manufacturing of samples out of non-porous material containing voids of the same volume but different shapes.

- In real cases porosity is embedded within the material. To inspect the elimination of natural porosity in FCS material, two experimental techniques are suggested here:

  • The first approach involves small cylinders of porous material which need to be scanned using x-ray to measure the amount of initial porosity. These samples need to be compressed under different temperature-load-holding-time combinations and then scanned using x-ray to investigate the extent of healing. The healing can be accomplished in a Gleeble system as in those tests explained in Chapter 7. The complication with this approach involves the sample size limitation, as the maximum penetrable thickness

for steel by x-ray is 4-5 mm. Such a small size specimen requires complicated machining and special test set ups. In Gleeble, isothermal flow stress compression anvils as shown in Figure 8.1 provide a uniform temperature distribution throughout the compression specimen during deformation tests. These anvils require a minimum contact with the specimen in order to produce a uniform temperature. For small specimens, sufficient contact cannot be made between the specimen and the anvils. Different solutions can be suggested for the problem of size limitation and complicacy of Gleeble testing for such small specimens. One solution could be the manufacturing of three-piece specimens as shown in Figure 8.2a. In this figure, part B is a porous cylindrical specimen cut from the centre of a bloom. To increase the contact area, parts A1 and A2 can be used, where part B is positioned between part A1 and part A2 and the whole set up is placed in the Gleeble so that the cross sections of part A1 and A2 are in contact with the anvils. However since parts A1, A2 and B are unattached, a mould (part C) is required to hold the pieces together when positioning the sample in the Gleeble (Figure 8.2b). This specimen set up is advantageous over dog-bone shape specimens since it requires simpler machining. Also using the specimen set up in Figure 8.2, Parts A1 and A2 can be cut from non-porous sections (to ensure good contact), whereas with dog-bone specimens such requirement may not be guaranteed.

- The porosity distribution increases from surface to the centre of a cast. In most cases the surface of the work-piece can be almost free of pores whereas the centre part usually has the highest porosity distribution. To investigate the healing of the voids concentrated at the centre, it is necessary to design a sample that somehow represents this porosity

distribution. To do so a porous section can be injected into a block of non-porous steel (very surface part of a billet) as shown in Figure 8.3.
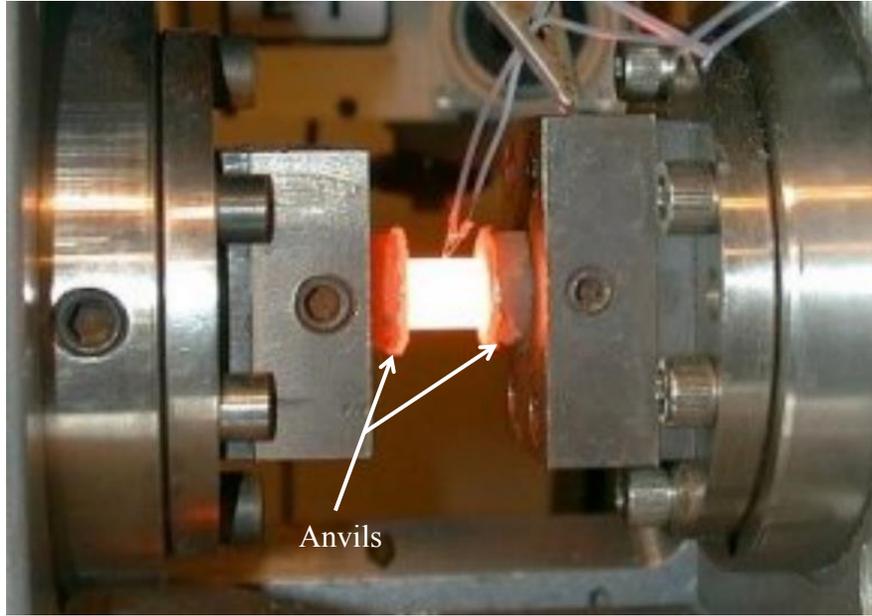
Figure 8.1*-Anvils in Gleeble compression set up.*
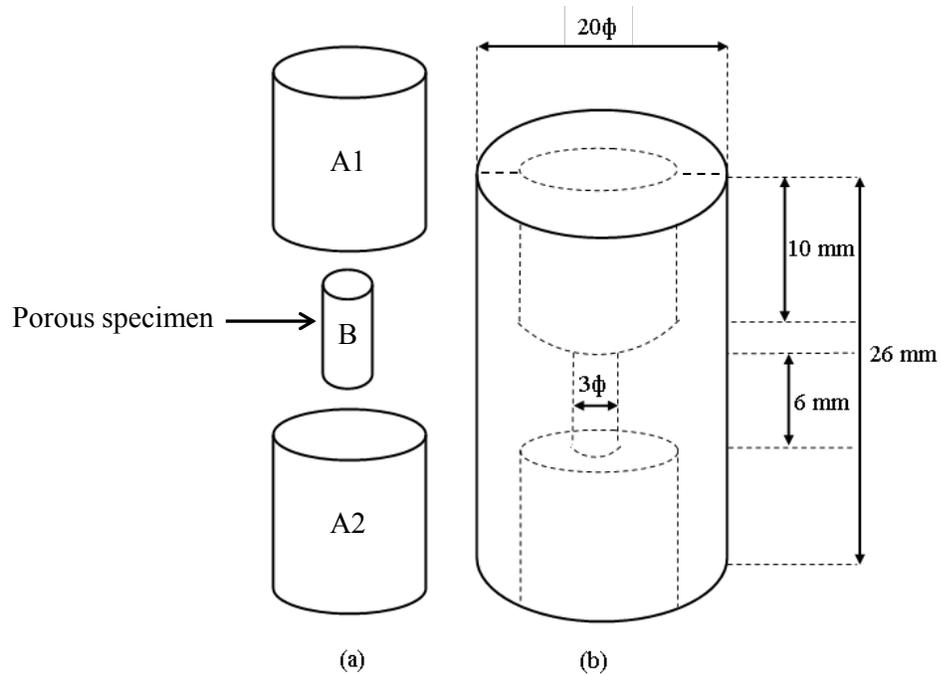
Figure 8.2*- (a) The three piece sample, (b) the mould.*
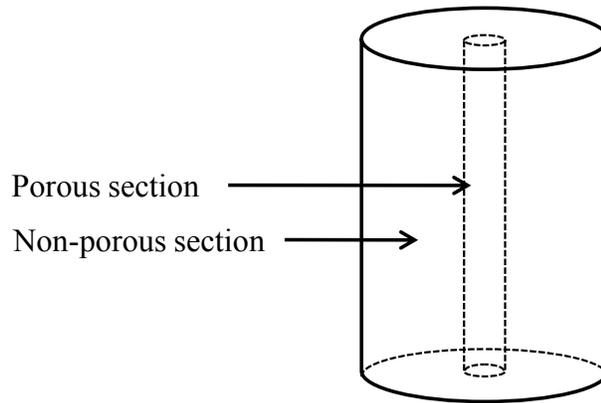
Figure 8.3 *- Sample with injected porous section in its centre.*

8.3 Recommendations for TATA steel

As discussed in Chapter 2, the application of diffusion bonding is advantageous over welding or adhesive bonding because of the fact that no melting or additional materials are involved, thus there is no discontinuity in the microstructure and in properties at the bonding sites. Upon correct execution of the process, the bond produced will have indistinguishable properties from those of the bulk metal and the bond line will have continuous metallurgical structure. However, diffusion bonding is not as fast as welding or adhesive bonding.

As explained in Chapter 6, the time required to obtain satisfactory bonding significantly depends on parameters such as temperature, H_m/L, vvf_c and f_o. Higher temperatures and lower values of H_m/L, vvf_c and f_o reduce the healing time. In an ideal case, if the highest possible temperature and the lowest possible values for H_m/L, vvf_c and f_o are considered and the highest number of stands is used in the multi-pass rolling process, diffusion bonding is possible to occur. However, such ideal cases are not always practical.

To ensure diffusion bonding occurs during rolling, the process has to be slowed down. Reducing the roll rotation speed and increasing the number of roll stands will

favour the diffusion bonding process. However, this will cause heat loss in the work-piece during rolling. To prevent this, the rolling process can be performed in furnace rooms but this would reduce output and add cost. Apart from performing the rolling process in furnace rooms, it may not be possible to heal under these conditions.

Appendices

Appendix A: The free cutting steel material data:

Test 1

Temperature:	900
Strain Rate:	0.1
Testpiece:	22 A2 205

True Strain	True Stress	
0	0	
0.002496	21.41887	
0.005286	43.5493	
0.012334	60.71616	
0.026136	66.93404	
0.047868	74.59221	
0.077528	82.57222	Peak Stress
0.109244	89.36411	106.2008
0.14096	95.30389	
0.159755	98.06371	Strain at fail
0.189415	102.2495	0.364
0.216139	104.6874	
0.247855	106.2008	Stress @
0.266649	104.3505	e=0.1
0.29631	97.02045	87.38449
0.323033	79.82758	
0.354749	46.17051	
0.364	0	

Test 2

Temperature:	900
Strain Rate:	1
Testpiece:	22 A1 075

True Strain	True Stress	
0	0	
0.007859	38.17105	
0.015717	57.91264	
0.02947	80.0588	
0.051082	97.16647	
0.074658	111.2701	
0.098234	122.1737	Peak Stress
0.12181	130.6187	158.5324
0.145386	138.3378	
0.168966	144.7028	Strain at fail
0.192546	151.0056	0.352
0.216122	156.2313	
0.229875	158.2159	Stress @
0.240681	158.5324	e=0.1
0.255416	157.849	122.8063
0.267204	152.2697	
0.286851	133.088	
0.310427	97.74193	
0.334003	62.2972	
0.352	0	

Test 3

Temperature:	900
Strain Rate:	10
Testpiece:	22 A3 335

True Strain	True Stress	
0	0	
0.004852	35.98829	
0.011091	49.26787	
0.018023	61.94055	
0.027728	77.00529	
0.042978	93.84223	
0.06516	110.4572	Peak Stress
0.090115	121.0442	196.4634
0.11507	135.5402	
0.140025	148.2853	Strain at fail
0.16498	158.7006	0.337
0.189935	169.9888	
0.21489	180.138	Stress @
0.239845	187.2141	e=0.1
0.2648	194.5237	126.7861
0.279358	196.4634	
0.302233	187.5019	
0.323001	106.2959	
0.337	0	

Test 5

Temperature:	1100
Strain Rate:	0.1
Testpiece:	22 C2 335

True Strain	True Stress	
0	0	
0.00846	14.16344	
0.017377	30.52683	
0.034296	40.34204	
0.057847	45.34272	
0.084713	49.66176	
0.113236	53.43797	Peak Stress
0.156851	58.69955	74.6923
0.188862	62.03769	
0.228646	65.94387	Strain at fail
0.291296	71.33976	1.03
0.342742	74.33302	
0.390414	74.6923	Stress @
0.460038	72.90534	e=0.1
0.497078	72.01081	51.68558
0.560185	71.6712	
0.631523	71.16756	
0.722067	69.48183	
0.812611	64.97928	
0.903155	56.20633	
0.9937	40	
1.03	0	

Test 6

Temperature:	1100
Strain Rate:	1
Testpiece:	22 B1 205

True Strain	True Stress	
0	0	
0.014779	26.98867	
0.062401	53.48913	
0.154368	70.72355	
0.211847	78.35842	
0.263575	84.75933	
0.30586	89.56209	Peak Stress
0.343425	93.36945	101.3711
0.378627	96.63628	
0.412651	98.71306	Strain at fail
0.463095	101.3711	0.849
0.497588	101.3047	
0.540291	100.1485	Stress @
0.607618	93.4897	e=0.1
0.699577	78.51231	60.53516
0.798113	58.55171	
0.849	0	
0.849	0	

Test 7

Temperature:	1100
Strain Rate:	10
Testpiece:	22 B3 205

True Strain	True Stress	
0	0	
0.004202	28.53892	
0.008404	44.24134	
0.029414	69.35941	
0.053725	76.02508	
0.093043	85.19875	
0.129661	92.31358	Peak Stress
0.172911	100.6297	114.5785
0.216161	107.5062	
0.241973	110.5413	Strain at fail
0.283843	114.5785	0.504
0.300422	114.0362	
0.324211	113.3807	Stress @
0.347622	108.9145	e=0.1
0.361129	107.6005	86.55043
0.381838	101.1247	
0.406845	89.35069	
0.426259	79.0532	
0.437064	72.24474	
0.46208	59.26335	
0.480688	45.50109	
0.504	0	

Appendix B: All codes used for the calibration of *q* parameters in

Chapter 4

i) Python code to create the voided cell model

```
#------------------------------------------------------------------
#This python code creates and runs (in ABAQUS) the voided cell (VC)
#model explained in section 4.1.2.


#CreateVC.py
#------------------------------------------------------------------

from abaqus import *
from abaqusConstants import *
import sketch
import part
import step
import regionToolset
from material import *
import regionToolset
import section
import displayGroupMdbToolset as dgm
import assembly
import step
import interaction
import load
import mesh
import job
import visualization
import xyPlot
import displayGroupOdbToolset as dgo
import connectorBehavior
#------------------------------------------------------------------

#READ TEXT FILES

mypath='C:/sa809'
text_file = open(mypath+'/readparameter.txt','r')
lines=text_file.readlines()
#------------------------------------------------------------------

#PARAMETERS

E=120000.0
v=0.33
W=2.5
H=5.0
Rdie=6.0
Tstr=lines[0]
T=float(Tstr)
f0str=lines[1]
f0=float(f0str)

my_amp=((0.0, 0.0), (2,0.1),(4,0.2),(6,0.3),(8,0.4),(10,0.5),
```

```
          (12,0.6),(14,0.7),(16,0.8),(18,0.9),(20,1.0))
#---------------------------------------------------------------------

#MAXIMUM LOAD APPLIED TO THE CYLINDER

Py=12400
Px=Py*(3*T+1)/(3*T-2)
#---------------------------------------------------------------------

#CALCULATING THE LOAD APPLIED TO THE DIE

PPy=Py*(W/Rdie)**2.0

r=pow(((3.0*f0*H*pow(W,2.0))/4.0),0.33333333)
#---------------------------------------------------------------------

#GEOMETRY

myPoints=((0,r),(0,H/2),(W,H/2),(W,-H/2),(0,-H/2),(0,-r))
m=(W,0)
middlePoint =(r,0)
#---------------------------------------------------------------------

#MODEL DEFINITION

myModel = mdb.Model(name='CellModel-1')
#---------------------------------------------------------------------

#SKETCH GEOMETRIES

mySketch=myModel.ConstrainedSketch(name='Sketch A', sheetSize=10.0)
mySketch.Line(myPoints[0],myPoints[1])
mySketch.Line(myPoints[1],myPoints[2])
mySketch.Line(myPoints[2],myPoints[3])
mySketch.Line(myPoints[3],myPoints[4])
mySketch.Line(myPoints[4],myPoints[5])
mySketch.Arc3Points(myPoints[0],myPoints[5], middlePoint)
CentreLine=mySketch.ConstructionLine((0,15),(0,0))
mySketch.assignCenterline(CentreLine)
#---------------------------------------------------------------------

#CREATE PART1

MyPart=mdb.models['CellModel-1'].Part(dimensionality=AXISYMMETRIC,
                                       name='Part-1',
                                       type=DEFORMABLE_BODY)
MyPart.BaseShell(sketch=mySketch)
#---------------------------------------------------------------------

#CREATE THE FCS MATERIAL AND THE SECTION

myMaterial=mdb.models['CellModel-1'].Material(name='Material-1')
myMaterial.Elastic(table=((50.0, 0.0), (77.49135123, 0.05),
 (92.68496318, 0.1), (102.9185488, 0.15), (110.8575636, 0.2),
 (117.434901, 0.25), (123.0976329, 0.3), (128.097914, 0.35),
 (132.5932383, 0.4), (136.6891598, 0.45), (140.4601843, 0.5),
 (143.9610364, 0.55), (147.2331995, 0.6), (150.3089364, 0.65),
 (153.2138781, 0.7), (155.9687552, 0.75), (158.5905939, 0.8),
 (161.0935641, 0.85), (163.4895965, 0.9), (165.7888385, 0.95),
```

221

```
 (168.0, 1.0),  (170.1306173, 1.05),  (172.1872589, 1.1),
 (174.1756877, 1.15),  (176.10099, 1.2),  (177.9676796, 1.25),
 (179.7797821, 1.3),  (181.5409042, 1.35),  (183.2542913, 1.4),
 (184.922875, 1.45),  (186.549313, 1.5),  (188.136023, 1.55),
 (189.6852115, 1.6),  (191.1988979, 1.65),  (192.6789354, 1.7),
 (194.1270298, 1.75),  (195.5447542, 1.8),  (196.9335633, 1.85),
 (198.2948051, 1.9),  (199.6297311, 1.95),  (200.9395057, 2.0)))


mdb.models['CellModel-1'].HomogeneousSolidSection(name='Section-1',
                                                material='Material
                                                -1',
                                                thickness=None)
f = MyPart.faces
faces = f.getSequenceFromMask(mask=('[#1 ]', ), )
region = regionToolset.Region(faces=faces)

MyPart.SectionAssignment(region=region,
                         sectionName='Section-1',
                         offset=0.0,
                         offsetType=MIDDLE_SURFACE,
                         offsetField='')
#----------------------------------------------------------------

#INSTANCE THE PART1

myAssembly=mdb.models['CellModel-1'].rootAssembly
myInstance=myAssembly.Instance(dependent=OFF,
                  name='Part-1-1',
                  part=MyPart)
#----------------------------------------------------------------

#DEFINE SETS

e1 = myInstance.edges
edges1 = e1.getSequenceFromMask(mask=('[#10 ]', ), )
myAssembly.Set(edges=edges1, name='TOP')

edges1 = e1.getSequenceFromMask(mask=('[#8 ]', ), )
myAssembly.Set(edges=edges1, name='Right')

edges1 = e1.getSequenceFromMask(mask=('[#4 ]', ), )
myAssembly.Set(edges=edges1, name='Bottom')

edges1 = e1.getSequenceFromMask(mask=('[#22 ]', ), )
myAssembly.Set(edges=edges1, name='Axis')
#----------------------------------------------------------------

#DEFINE SURFACES

side1Edges1 = e1.getSequenceFromMask(mask=('[#1 ]', ), )
myAssembly.Surface(side1Edges=side1Edges1, name='HoleSurface')

side1Edges1 = e1.getSequenceFromMask(mask=('[#10 ]', ), )
myAssembly.Surface(side1Edges=side1Edges1, name='TOP')

side1Edges1 = e1.getSequenceFromMask(mask=('[#8 ]', ), )
myAssembly.Surface(side1Edges=side1Edges1, name='Right')
#----------------------------------------------------------------
```

```
#CUT

myCutPoints=((0,2*r), (W,2*r), (W,-2*r), (0, -2*r), (2*r,H/2),
(2*r,-H/2), (W,0))
originPoint=(0.0,0.0)
interSection1=(2*r,2*r)
interSection2=(2*r,-2*r)
f1 = myInstance.faces
t = myAssembly.MakeSketchTransform(sketchPlane=f1[0],
                                   sketchPlaneSide=SIDE1,
                                   origin=(0.0, 0.0, 0.0))
s = mdb.models['CellModel-1'].ConstrainedSketch(name='__profile__',
                                                sheetSize=200,
                                                gridSpacing=0.5,
                                                transform=t)
g, v, d, c = s.geometry, s.vertices, s.dimensions, s.constraints

s.setPrimaryObject(option=SUPERIMPOSE)

myAssembly.projectReferencesOntoSketch(sketch=s,
                                       filter=COPLANAR_EDGES)

s.Line(point1=myCutPoints[0], point2=myCutPoints[1])
s.HorizontalConstraint(entity=g[9])
s.PerpendicularConstraint(entity1=g[7], entity2=g[9])
s.CoincidentConstraint(entity1=v[7], entity2=g[7])
s.CoincidentConstraint(entity1=v[8], entity2=g[5])
s.Line(point1=myCutPoints[2], point2=myCutPoints[3])
s.HorizontalConstraint(entity=g[10])
s.PerpendicularConstraint(entity1=g[3], entity2=g[10])
s.CoincidentConstraint(entity1=v[9], entity2=g[3])
s.CoincidentConstraint(entity1=v[10], entity2=g[5])
s.Line(point1=myCutPoints[4], point2=myCutPoints[5])
s.VerticalConstraint(entity=g[11])
s.PerpendicularConstraint(entity1=g[6], entity2=g[11])
s.CoincidentConstraint(entity1=v[11], entity2=g[6])
s.CoincidentConstraint(entity1=v[12], entity2=g[4])
s.Line(point1=middlePoint, point2=myCutPoints[6])
s.HorizontalConstraint(entity=g[12])
s.PerpendicularConstraint(entity1=g[2], entity2=g[12])
s.CoincidentConstraint(entity1=v[13], entity2=g[2])
s.EqualDistanceConstraint(entity1=v[0], entity2=v[1],
midpoint=v[13])
s.CoincidentConstraint(entity1=v[14], entity2=g[5])
s.EqualDistanceConstraint(entity1=v[4], entity2=v[5],
midpoint=v[14])
s.Line(point1=originPoint, point2=interSection1)
s.CoincidentConstraint(entity1=v[15], entity2=g[9])
s.Line(point1=originPoint, point2=interSection2)
s.CoincidentConstraint(entity1=v[16], entity2=g[10])
f1 = myAssembly.instances['Part-1-1'].faces
pickedFaces = f1.getSequenceFromMask(mask=('[#1 ]', ), )
myAssembly.PartitionFaceBySketch(faces=pickedFaces, sketch=s)
s.unsetPrimaryObject()
#-----------------------------------------------------------------

#CREATE PART2
```

```
s = mdb.models['CellModel-1'].ConstrainedSketch(name='__profile__',
                                                sheetSize=200.0)
g, v, d, c = s.geometry, s.vertices, s.dimensions, s.constraints
s.sketchOptions.setValues(viewStyle=AXISYM)
s.setPrimaryObject(option=STANDALONE)
s.ConstructionLine(point1=(0.0, -100.0), point2=(0.0, 100.0))
s.FixedConstraint(entity=g[2])
s.Line(point1=(0.0, 0.0), point2=(20.0, 0.0))
s.HorizontalConstraint(entity=g[3])
s.PerpendicularConstraint(entity1=g[2], entity2=g[3])
s.CoincidentConstraint(entity1=v[0], entity2=g[2])
s.ObliqueDimension(vertex1=v[0], vertex2=v[1],
textPoint=(7.38391780853271, 3.68696308135986), value=Rdie)
p = mdb.models['CellModel-1'].Part(name='Part-2',
                                   dimensionality=AXISYMMETRIC,
                                   type=DISCRETE_RIGID_SURFACE)
p = mdb.models['CellModel-1'].parts['Part-2']
p.BaseWire(sketch=s)
v2, e1, d2, n1 = p.vertices, p.edges, p.datums, p.nodes
p.ReferencePoint(point=v2[1])
#-------------------------------------------------------------------

#INSTANCE PART2

myAssembly=mdb.models['CellModel-1'].rootAssembly
p = mdb.models['CellModel-1'].parts['Part-2']
myInstance = myAssembly.Instance(name='Part-2-1', part=p,
dependent=OFF)
myAssembly.translate(instanceList=('Part-2-1', ), vector=(0.0, 2.5,
                                                           0.0))
#-------------------------------------------------------------------

#PART2 SURFACES

s1 = myAssembly.instances['Part-2-1'].edges
side1Edges1 = s1.getSequenceFromMask(mask=('[#1 ]', ), )
myAssembly.Surface(side1Edges=side1Edges1, name='Surf-out')

s1 = myAssembly.instances['Part-2-1'].edges
side2Edges1 = s1.getSequenceFromMask(mask=('[#1 ]', ), )
myAssembly.Surface(side2Edges=side2Edges1, name='Surf-in')
#-------------------------------------------------------------------

#CREATE A NEW STEP

mdb.models['CellModel-1'].StaticStep(name='Apply Load Step',
        previous='Initial', timePeriod=20.0,
        maxNumInc=100000000, initialInc=20.0, minInc=2e-16,
        maxInc=20.0)

mdb.models['CellModel-1'].steps['
                                Apply Load Step'].setValues(nlgeom=ON)
#-------------------------------------------------------------------

#DEFINE CONTACT

mdb.models['CellModel-1'].ContactProperty('IntProp-1')
mdb.models['CellModel-1'].interactionProperties['IntProp-
1'].NormalBehavior(
```

```
                       pressureOverclosure=HARD, allowSeparation=ON,
                       constraintEnforcementMethod=DEFAULT)

a = mdb.models['CellModel-1'].rootAssembly
region=a.surfaces['HoleSurface']
mdb.models['CellModel-1'].SelfContactStd(name='Int-1',
                               createStepName='Initial',
                               surface=region,
                               interactionProperty='IntProp-1',
                               enforcement=SURFACE_TO_SURFACE,
                               thickness=ON)

mdb.models['CellModel-1'].ContactStd(name='Int-2',
                                createStepName='Initial')
mdb.models['CellModel-1'].interactions['Int-2'].
                       includedPairs.setValuesInStep(
                               stepName='Initial',
                               useAllstar=ON)
mdb.models['CellModel-1'].interactions['Int-2'].
                       contactPropertyAssignments.
                       appendInStep(stepName='Initial',
                       assignments=((GLOBAL, SELF, 'IntProp-1'), ))
#----------------------------------------------------------------

#CREATE BOUNDARY CONDITIONS

region = myAssembly.sets['Axis']
mdb.models['CellModel-1'].XsymmBC(name='BC-1',
                               createStepName='Initial',
                               region=region)
region = myAssembly.sets['Bottom']
mdb.models['CellModel-1'].DisplacementBC(name='BC-2',
                                     createStepName='Initial',
                                     region=region, u1=UNSET,
                                     u2=SET, ur3=UNSET,
                                     amplitude=UNSET,
                                     distributionType=UNIFORM,
                                     fieldName='',
                                     localCsys=None)
a = mdb.models['CellModel-1'].rootAssembly
r1 = a.instances['Part-2-1'].referencePoints
refPoints1=(r1[2], )
region = regionToolset.Region(referencePoints=refPoints1)
mdb.models['CellModel-1'].DisplacementBC(name='BC-3',
createStepName='Initial',region=region, u1=SET, u2=UNSET, ur3=SET,
                 amplitude=UNSET,distributionType=UNIFORM,
                 fieldName='', localCsys=None)
#----------------------------------------------------------------
#CREATE LOAD

mdb.models['CellModel-1'].TabularAmplitude(name='Amp-2',
                                     timeSpan=STEP,
                                     smooth=SOLVER_DEFAULT,
                                     data=my_amp)
region = myAssembly.surfaces['Surf-out']
mdb.models['CellModel-1'].Pressure(name='Load-2',
                               createStepName='Apply Load Step',
                               region=region,
```

```
                                           distributionType=UNIFORM,
                                           field='', magnitude=PPy,
                                           amplitude='Amp-2')

mdb.models['CellModel-1'].TabularAmplitude(name='Amp-1',
                                           timeSpan=STEP,
                                           smooth=SOLVER_DEFAULT,
                                           data=my_amp)

region = myAssembly.surfaces['Right']
mdb.models['CellModel-1'].Pressure(name='Load-1',
                                           createStepName='Apply Load
                                           Step',region=region,distributionT
                                           ype=UNIFORM, field='',
                                           magnitude=Px, amplitude='Amp-1')
#-------------------------------------------------------------------

#MESH

e1 = myAssembly.instances['Part-1-1'].edges
pickedEdges = e1.getSequenceFromMask(mask=('[#ffffffff ]', ), )
myAssembly.seedEdgeByNumber(edges=pickedEdges, number=10)
f1 = myAssembly.instances['Part-1-1'].faces
pickedRegions = f1.getSequenceFromMask(mask=('[#3ff ]', ), )
myAssembly.setMeshControls(regions=pickedRegions,
technique=STRUCTURED)
partInstances =(myAssembly.instances['Part-1-1'], )
myAssembly.generateMesh(regions=partInstances)

a = mdb.models['CellModel-1'].rootAssembly
partInstances =(a.instances['Part-2-1'], )
a.seedPartInstance(regions=partInstances, size=0.6,
deviationFactor=0.1)
a.generateMesh(regions=partInstances)
#-------------------------------------------------------------------

#DEFINE NODESETS

n1 = myAssembly.instances['Part-1-1'].nodes
nodes1 = n1.getSequenceFromMask(mask=('[#230 #0 #1ff #80000000 #ff
]', ), )
myAssembly.Set(nodes=nodes1, name='EdgeNodeSet1')

n1 = myAssembly.instances['Part-1-1'].nodes
nodes1 = n1.getSequenceFromMask(mask=('[#4c0 #0:2 #1ff0 #1ff00 ]',
), )
myAssembly.Set(nodes=nodes1, name='HoleSurfaceNodeSet1')


#FIELD OUTPUT REQUIST

mdb.models['CellModel-1'].fieldOutputRequests['F-Output-
1'].setValues
                           (numIntervals=2000,variables=('S',
                           'PE', 'PEEQ', 'PEMAG', 'LE', 'U',
                           'RF', 'CF',
                           'CSTRESS','COORD'))

mdb.models['CellModel-1'].fieldOutputRequests['F-Output1'].setValues
```

```
                                        (numIntervals=2000,variables=('S
                                        ', 'PE', 'PEEQ', 'PEMAG', 'LE',
                                        'U', 'RF', 'CF',
                                        'CSTRESS','COORD'))
#----------------------------------------------------------------

#JOB

mdb.Job(name='VCJob-03-12', model='CellModel-1', description='',
        type=ANALYSIS, atTime=None, waitMinutes=0, waitHours=0,
        queue=None, memory=50, memoryUnits=PERCENTAGE,
        getMemoryFromAnalysis=True, explicitPrecision=SINGLE,
        nodalOutputPrecision=SINGLE, echoPrint=OFF,
        modelPrint=OFF, contactPrint=OFF, historyPrint=OFF,
        userSubroutine='', scratch='',
        parallelizationMethodExplicit=DOMAIN,
        multiprocessingMode=DEFAULT, numDomains=1, numCpus=1)

mdb.jobs['VCJob-03-12'].submit(consistencyChecking=OFF)
```

ii) Python code to create the GT cell model

```
#---------------------------------------------------------------------
#This python code creates and runs (in ABAQUS) the Gurson-Tvergaard
#(GT) model explained in section 4.2.2.

#CreatePartGT.py
#---------------------------------------------------------------------

from abaqus import *
from abaqusConstants import *
import sketch
import part
import step
import regionToolset
from material import *
import regionToolset
import section
import displayGroupMdbToolset as dgm
import assembly
import step
import interaction
import load
import mesh
import job
import visualization
import xyPlot
import displayGroupOdbToolset as dgo
import connectorBehavior
#---------------------------------------------------------------------

#READ q VALUES FROM TEXT FILE

mypath='C:/sa809'
TextFile = open(mypath+'/read_it.txt','r')
f=TextFile.readlines()
q1=float(f[0])
q2=float(f[1])
q3=float(f[2])
TextFile.close()
#---------------------------------------------------------------------

#READ TEXT FILE

text_file = open(mypath+'/readparameter.txt','r')
lines=text_file.readlines()
#---------------------------------------------------------------------

#PARAMETERS

E=120000.0
v=0.33
W=2.5
H=5.0
Rdie=6.0
Tstr=lines[0]
T=float(Tstr)
```

```
f0str=lines[1]
f0=float(f0str)

my_amp=((0.0,0.0),(2,0.1),(4,0.2),(6,0.3),(8,0.4),(10,0.5),(12,0.6),
        (14,0.7),(16,0.8),(18,0.9),(20,1.0))
#-------------------------------------------------------------------

#MAXIMUM LOAD APPLIED TO THE CYLINDER

Py=12400
Px=Py*(3*T+1)/(3*T-2)
#-------------------------------------------------------------------

#CALCULATING THE LOAD APPLIED TO THE DIE

PPy=Py*(W/Rdie)**2.0
r=pow(((3.0*f0*H*pow(W,2.0))/4.0),0.33333333)
#-------------------------------------------------------------------

#GEOMETRY

myPoints=((0,H/2),(W,H/2),(W,-H/2),(0,-H/2))
#-------------------------------------------------------------------

#MODEL DEFINITION

myModel = mdb.Model(name='CellModel')
#-------------------------------------------------------------------

#SKETCH GEOMETRIES

mySketch=myModel.ConstrainedSketch(name='Sketch A', sheetSize=10.0)
CentreLine=mySketch.ConstructionLine((0,15),(0,0))
mySketch.assignCenterline(CentreLine)
mySketch.rectangle(point1=(0.0, H/2), point2=(W, -H/2))
#-------------------------------------------------------------------

#CREATE PART

MyPart=mdb.models['CellModel'].Part(dimensionality=AXISYMMETRIC,
                                    name='Part1',
                                    type=DEFORMABLE_BODY)
MyPart.BaseShell(sketch=mySketch)
#-------------------------------------------------------------------

#CREATE THE FreeCuttingSteel material and section

myMaterial=mdb.models['CellModel'].Material(name='Material1')
myMaterial.Density(table=((7.8e-09, ), ))
myMaterial.Elastic(table=((E, v),))
myMaterial.PorousMetalPlasticity(relativeDensity=1-f0, table=((q1,
q2, q3), ))
myMaterial.Plastic(table=((50.0, 0.0), (77.49135123, 0.05),
  (92.68496318, 0.1), (102.9185488, 0.15), (110.8575636, 0.2),
  (117.434901, 0.25), (123.0976329, 0.3), (128.097914, 0.35),
  (132.5932383, 0.4), (136.6891598, 0.45), (140.4601843, 0.5),
  (143.9610364, 0.55), (147.2331995, 0.6), (150.3089364, 0.65),
  (153.2138781, 0.7), (155.9687552, 0.75), (158.5905939, 0.8),
  (161.0935641, 0.85), (163.4895965, 0.9), (165.7888385, 0.95),
```

```
 (168.0, 1.0), (170.1306173, 1.05), (172.1872589, 1.1),
 (174.1756877, 1.15), (176.10099, 1.2), (177.9676796, 1.25),
 (179.7797821, 1.3), (181.5409042, 1.35), (183.2542913, 1.4),
 (184.922875, 1.45), (186.549313, 1.5), (188.136023, 1.55),
 (189.6852115, 1.6), (191.1988979, 1.65), (192.6789354, 1.7),
 (194.1270298, 1.75), (195.5447542, 1.8), (196.9335633, 1.85),
 (198.2948051, 1.9), (199.6297311, 1.95), (200.9395057, 2.0))


mdb.models['CellModel'].HomogeneousSolidSection(name='Section1',
                                                material='Material
                                                1',
                                                thickness=None)
f = MyPart.faces
faces = f.getSequenceFromMask(mask=('[#1 ]', ), )
region = regionToolset.Region(faces=faces)

MyPart.SectionAssignment(region=region,
                         sectionName='Section1',
                         offset=0.0,
                         offsetType=MIDDLE_SURFACE,
                         offsetField='')
#--------------------------------------------------------------------

#INSTANCE THE PART

myAssembly=mdb.models['CellModel'].rootAssembly
myInstance=myAssembly.Instance(dependent=OFF,
                               name='Part11',
                               part=MyPart)
#--------------------------------------------------------------------

#DEFINE SETS

e1 = myInstance.edges
edges1 = e1.getSequenceFromMask(mask=('[#1 ]', ), )
myAssembly.Set(edges=edges1, name='TOP')

edges1 = e1.getSequenceFromMask(mask=('[#8 ]', ), )
myAssembly.Set(edges=edges1, name='Right')

edges1 = e1.getSequenceFromMask(mask=('[#4 ]', ), )
myAssembly.Set(edges=edges1, name='Bottom')

edges1 = e1.getSequenceFromMask(mask=('[#2 ]', ), )
myAssembly.Set(edges=edges1, name='Axis')
#--------------------------------------------------------------------

#DEFINE SURFACES

side1Edges1 = e1.getSequenceFromMask(mask=('[#1 ]', ), )
myAssembly.Surface(side1Edges=side1Edges1, name='TOP')

side1Edges1 = e1.getSequenceFromMask(mask=('[#8 ]', ), )
myAssembly.Surface(side1Edges=side1Edges1, name='Right')
#--------------------------------------------------------------------

#CREATE PART2

s = mdb.models['CellModel'].ConstrainedSketch(name='__profile__',
```

```
                                                    sheetSize=200.0)
g, v, d, c = s.geometry, s.vertices, s.dimensions, s.constraints
s.sketchOptions.setValues(viewStyle=AXISYM)
s.setPrimaryObject(option=STANDALONE)
s.ConstructionLine(point1=(0.0, -100.0), point2=(0.0, 100.0))
s.FixedConstraint(entity=g[2])
s.Line(point1=(0.0, 0.0), point2=(20.0, 0.0))
s.HorizontalConstraint(entity=g[3])
s.PerpendicularConstraint(entity1=g[2], entity2=g[3])
s.CoincidentConstraint(entity1=v[0], entity2=g[2])
s.ObliqueDimension(vertex1=v[0], vertex2=v[1],
textPoint=(7.38391780853271, 3.68696308135986),    value=6.0)
p = mdb.models['CellModel'].Part(name='Part-2',
                                 dimensionality=AXISYMMETRIC,
                                 type=DISCRETE_RIGID_SURFACE)
p = mdb.models['CellModel'].parts['Part-2']
p.BaseWire(sketch=s)
v2, e1, d2, n1 = p.vertices, p.edges, p.datums, p.nodes
p.ReferencePoint(point=v2[1])
#----------------------------------------------------------------

#INSTANCE PART2

myAssembly=mdb.models['CellModel'].rootAssembly
p = mdb.models['CellModel'].parts['Part-2']
myInstance = myAssembly.Instance(name='Part-2-1', part=p,
                                 dependent=OFF)
myAssembly.translate(instanceList=('Part-2-1', ), vector=(0.0, 2.5,
                                    0.0))
#----------------------------------------------------------------

#PART2 SURFACES

s1 = myAssembly.instances['Part-2-1'].edges
side1Edges1 = s1.getSequenceFromMask(mask=('[#1 ]', ), )
myAssembly.Surface(side1Edges=side1Edges1, name='Surf-out')

s1 = myAssembly.instances['Part-2-1'].edges
side2Edges1 = s1.getSequenceFromMask(mask=('[#1 ]', ), )
myAssembly.Surface(side2Edges=side2Edges1, name='Surf-in')
#----------------------------------------------------------------

#CREATE A NEW STEP

mdb.models['CellModel'].StaticStep(name='Apply Load Step1',
                                   previous='Initial',
                                   timePeriod=24.0,
                                   maxNumInc=100000000,
                                   initialInc=24.0, minInc=2e-16,
                                   maxInc=24.0)


mdb.models['CellModel'].steps['Apply Load
Step1'].setValues(nlgeom=ON)
mdb.models['CellModel'].rootAssembly.regenerate()
#----------------------------------------------------------------

#DEFINE CONTACT
```

```
mdb.models['CellModel'].ContactProperty('IntProp-1')
mdb.models['CellModel'].interactionProperties['IntProp-
1'].NormalBehavior(pressureOverclosure=HARD,
                 allowSeparation=ON,
                 constraintEnforcementMethod=DEFAULT)
mdb.models['CellModel'].ContactStd(name='Int-2',
                                createStepName='Initial')
mdb.models['CellModel'].interactions['Int2'].includedPairs.setValues
InStep(stepName='Initial', useAllstar=ON)mdb.models[
                'CellModel'].interactions['Int2'].
                contactPropertyAssignments.appendInStep(
                stepName='Initial', assignments=((GLOBAL, SELF,
                'IntProp-1'), ))
#----------------------------------------------------------------

#CREATE BOUNDARY CONDITIONS

region = myAssembly.sets['Axis']
mdb.models['CellModel'].XsymmBC(name='BC-1',
createStepName='Initial', region=region)
region = myAssembly.sets['Bottom']
mdb.models['CellModel'].DisplacementBC(name='BC-2',
                                createStepName='Initial',
                                region=region, u1=UNSET,
                                u2=SET,ur3=UNSET,
                                amplitude=UNSET,
                                distributionType=UNIFORM,
                                fieldName='', localCsys=None)

a = mdb.models['CellModel'].rootAssembly
r1 = a.instances['Part-2-1'].referencePoints
refPoints1=(r1[2], )
region = regionToolset.Region(referencePoints=refPoints1)
mdb.models['CellModel'].DisplacementBC(name='BC-3',
                                createStepName='Initial',
                                region=region, u1=SET,
                                u2=UNSET, ur3=SET,
                                amplitude=UNSET,
                                distributionType=UNIFORM,
                                fieldName='', localCsys=None)
#----------------------------------------------------------------

#CREATE LOAD

mdb.models['CellModel-1'].TabularAmplitude(name='Amp-2',
                                timeSpan=STEP,
                                smooth=SOLVER_DEFAULT,
                                data=my_amp)
region = myAssembly.surfaces['Surf-out']
mdb.models['CellModel-1'].Pressure(name='Load-2',
                                createStepName='Apply Load Step',
                                region=region,
                                distributionType=UNIFORM,
                                field='', magnitude=PPy,
                                amplitude='Amp-2')

mdb.models['CellModel-1'].TabularAmplitude(name='Amp-1',
                                timeSpan=STEP,
```

```
                                                        smooth=SOLVER_DEFAULT,
                                                        data=my_amp)

region = myAssembly.surfaces['Right']
mdb.models['CellModel-1'].Pressure(name='Load-1',
                                createStepName='Apply Load
                                Step',region=region,distributionType=U
                                NIFORM, field='', magnitude=Px,
                                amplitude='Amp-1')
#-------------------------------------------------------------------

#MESH

partInstances =(myAssembly.instances['Part11'], )
myAssembly.seedPartInstance(regions=partInstances, size=0.5,
                            deviationFactor=0.1)
myAssebly = mdb.models['CellModel'].rootAssembly
e1 = myAssembly.instances['Part11'].edges
pickedEdges = e1.getSequenceFromMask(mask=('[#2 ]', ), )
myAssembly.seedEdgeByNumber(edges=pickedEdges, number=20)
myAssembly = mdb.models['CellModel'].rootAssembly
e1 = myAssembly.instances['Part11'].edges
pickedEdges = e1.getSequenceFromMask(mask=('[#5 ]', ), )
myAssembly.seedEdgeByNumber(edges=pickedEdges, number=10)
myAssembly = mdb.models['CellModel'].rootAssembly
e1 = myAssembly.instances['Part11'].edges
pickedEdges = e1.getSequenceFromMask(mask=('[#8 ]', ), )
myAssembly.seedEdgeByNumber(edges=pickedEdges, number=20)
myAssembly = mdb.models['CellModel'].rootAssembly
f1 = myAssembly.instances['Part11'].faces
pickedRegions = f1.getSequenceFromMask(mask=('[#1 ]', ), )
myAssembly.setMeshControls(regions=pickedRegions,
                            technique=STRUCTURED)
myAssembly = mdb.models['CellModel'].rootAssembly
partInstances =(myAssembly.instances['Part11'], )
myAssembly.generateMesh(regions=partInstances)

a = mdb.models['CellModel'].rootAssembly
partInstances =(a.instances['Part-2-1'], )
a.seedPartInstance(regions=partInstances, size=0.6,
deviationFactor=0.1)
a.generateMesh(regions=partInstances)
#-------------------------------------------------------------------

#CREATE ELEMENTSET

e1 = myAssembly.instances['Part11'].elements
elements1 = e1.getSequenceFromMask(mask=('[#ffffffff:6 #ff ]', ), )
myAssembly.Set(elements=elements1, name='WHOLE')


#FIELD OUTPUT REQUIST

mdb.models['CellModel'].FieldOutputRequest(name='F-Output-1',
                                        createStepName='Apply
                                        Load
                                        Step1',numIntervals=200,
                                        variables=('S', 'PE',
```

233

```
                                        'PEEQ', 'U', 'RF', 'CF',
                                        'VVF'))
#-----------------------------------------------------------------

#JOB

mdb.Job(name='GTJob-03-12', model='CellModel',
        description='examine the effect of triaxiality',
        type=ANALYSIS,atTime=None, waitMinutes=0, waitHours=0,
        queue=None, memory=50, memoryUnits=PERCENTAGE,
        getMemoryFromAnalysis=True,
        explicitPrecision=DOUBLE, nodalOutputPrecision=FULL,
        echoPrint=OFF, modelPrint=OFF, contactPrint=OFF,
        historyPrint=OFF, userSubroutine='',
        scratch='', parallelizationMethodExplicit=DOMAIN,
        multiprocessingMode=DEFAULT, numDomains=1, numCpus=1)

mdb.jobs['GTJob-03-12'].submit(consistencyChecking=OFF)
```

iii)   Python code to extract the coordinates of the void

```
#--------------------------------------------------------------------
#This python code extracts the XY coordinates of the void (profile
#of the void as explained in section 4.1.3)during deformation. The
#data is written into a ABAQUS report file (xyCoordReport.rpt).


#xyReport.py
#--------------------------------------------------------------------

from abaqus import *
from abaqusConstants import *
import __main__
import section
import regionToolset
import displayGroupMdbToolset as dgm
import part
import material
import assembly
import step
import interaction
import load
import mesh
import job
import sketch
import visualization
import xyPlot
import displayGroupOdbToolset as dgo
import connectorBehavior
#--------------------------------------------------------------------

#CREATING XY DATA

o1=session.openOdb(name='c:/sa809/CM4R.odb')
session.viewports['Viewport: 1'].setValues(displayedObject=o1)
odb = session.odbs['c:/sa809/CM4R.odb']
session.xyDataListFromField(odb=odb, outputPosition=NODAL,
variable=(('COORD', NODAL, ((COMPONENT, 'COOR1'), (COMPONENT,
'COOR2'),)), ), nodeSets=('HOLESURFACENODESET1', ))
#--------------------------------------------------------------------

#WRITING REPORT

a=odb.rootAssembly.nodeSets['HOLESURFACENODESET1'].nodes[0]
l=len(a)
myLabelsx=[]
myLabelsy=[]
myxyData=[]
for i in range(0,l):
    bx=a[i].label
    myLabelsx.append(bx)
    myC1Label='COORD:COOR1 PI: PART-1-1 N: '+str(bx)
    x = session.xyDataObjects[myC1Label]
    myxyData.append(x)
for i in range(0,l):
    by=a[i].label
    myLabelsy.append(by)
```

```
    myC2Label='COORD:COOR2 PI: PART-1-1 N: '+str(by)
    y = session.xyDataObjects[myC2Label]
    myxyData.append(y)
session.writeXYReport(fileName='c:/sa809/xyCoordReport.rpt',
                      appendMode=OFF, xyData=myxyData)
```

iv) <u>Python code to extract the coordinates of the VC cell</u>

```
#---------------------------------------------------------------------
#This python code extracts the XY coordinates of the cell (profile
#of the cell as explained in section 4.1.3) during deformation. The
#data is written into a ABAQUS report file (ReadTotalVolume.rpt).

#TotalVolume.py
#---------------------------------------------------------------------
from abaqus import *
from abaqusConstants import *
import __main__
import section
import regionToolset
import displayGroupMdbToolset as dgm
import part
import material
import assembly
import step
import interaction
import load
import mesh
import job
import sketch
import visualization
import xyPlot
import displayGroupOdbToolset as dgo
import connectorBehavior
#---------------------------------------------------------------------

#CREATING XY DATA
o1=session.openOdb(name='c:/sa809/CM4R.odb')
session.viewports['Viewport: 1'].setValues(displayedObject=o1)
odb = session.odbs['c:/sa809/CM4R.odb']
session.xyDataListFromField(odb=odb,
outputPosition=NODAL,variable=(('COORD', NODAL, ((COMPONENT,
'COOR1'),(COMPONENT, 'COOR2'), )), ), nodeSets=('EDGENODESET1', ))
#---------------------------------------------------------------------

#WRITING REPORT

a=odb.rootAssembly.nodeSets['EDGENODESET1'].nodes[0]
l=len(a)
myLabelsx=[]
myLabelsy=[]
myxyData=[]

for i in range(0,l):
    bx=a[i].label
    myLabelsx.append(bx)
    myC1Label='COORD:COOR1 PI: PART-1-1 N: '+str(bx)
    x = session.xyDataObjects[myC1Label]
```

```
    myxyData.append(x)

for i in range(0,l):
    by=a[i].label
    myLabelsy.append(by)
    myC2Label='COORD:COOR2 PI: PART-1-1 N: '+str(by)
    y = session.xyDataObjects[myC2Label]
    myxyData.append(y)
session.writeXYReport(fileName='c:/sa809/ReadTotalVolume.rpt',
                      appendMode=OFF, xyData=myxyData)
```

v)   Python code to extract *vvf* of the GT cell model

```
#---------------------------------------------------------------
#This python code extracts the vvf of the GT cell model (see section
#4.2) during deformation. The data is written into a text file
#(VvfReport.txt).


#VVFreport.py
#---------------------------------------------------------------

import sys, os
from abaqus import*
from abaqusConstants import*
from caeModules import *
#---------------------------------------------------------------

# POINT AT FILES:

mypath='C:/sa809'

FOLDER = mypath

os.chdir(FOLDER)
for filename in os.listdir('.'):
 filelist = open(mypath+'/GTJob-03-12.odb','a')


odb=session.openOdb(name=mypath+'/GTJob-03-12.odb')

session.viewports['Viewport: 1'].setValues(displayedObject=odb)
session.viewports['Viewport: 1'].view.setValues(nearPlane=43.3078,
                farPlane=76.6444, width=38.4285, height=19.2364,
                cameraPosition=(-22.5167, 5.17008, 48.8967),
                cameraUpVector=(-0.602818, -0.140808, -0.785356))
odb = session.odbs[mypath+'/GTJob-03-12.odb']
#---------------------------------------------------------------

#EXTRACT DATA

session.xyDataListFromField(odb=odb,
outputPosition=INTEGRATION_POINT,
                        variable=(('VVF',INTEGRATION_POINT,),),
                        steps=('Apply Load Step1',),elementSets=
                        ('WHOLE', ))
x1 = session.xyDataObjects['VVF PI: PART11 E: 1 IP: 1']
#---------------------------------------------------------------

#ASSIGN DATA TO VARIABLE

VVF=[]
TIME=[]
NoElem=200
for i in range(1, NoElem+1):
    a= 'VVF PI: PART11 E:',i, 'IP: 1'
    b= 'VVF PI: PART11 E:',i, 'IP: 1'
    c= 'VVF PI: PART11 E:',i, 'IP: 1'
```

238

```python
    d= 'VVF PI: PART11 E:',i, 'IP: 1'
    a1=str(a)
    b1=str(b)
    c1=str(c)
    d1=str(d)
    a11=''
    b11=''
    c11=''
    d11=''
    for j in a1:
        if j!='('and j!=')'and j!=','and j!=','and j!="'":
            a11=a11+j
    for k in b1:
        if k!='('and k!=')'and k!=','and k!="'":
            b11=b11+k
    for h in c1:
        if h!='('and h!=')'and h!=','and h!="'":
            c11=c11+h
    for l in d1:
        if l!='('and l!=')'and l!=','and l!="'":
            d11=d11+l
    ipoint1 = session.xyDataObjects[a11]
    ipoint2 = session.xyDataObjects[b11]
    ipoint3 = session.xyDataObjects[c11]
    ipoint4 = session.xyDataObjects[d11]

    aveipoint=[]
    time=[]
    for j in range(0,len(ipoint1)):
        timetemp=ipoint1[j][0]
        time.append(timetemp)

        tmp=float((float(ipoint1[j][1])+float(ipoint2[j][1])+
        float(ipoint3[j][1])+float(ipoint4[j][1])))/4
        aveipoint.append(tmp)
    VVF.append(aveipoint)
    TIME=time

aveVVF=[]
for j in range(0,len(ipoint1)):
    sum=0
    for i in range(0,NoElem):
        sum=sum+VVF[i][j]
    aveVVF.append(sum/NoElem)


myoutput=''

f=open(mypath+'/VvfReport.txt','w')
for i in range(0,len(TIME)):
    myoutput=str(TIME[i])+','+str(aveVVF[i])+'\n'
    f.write(myoutput)
```

vi) MATLAB code for optimisation

```matlab
%-------------------------------------------------------------------
%This MATLAB code creates a graphical user interface (GUI) explained
%in section 4.2.3.

%Interface.m
%-------------------------------------------------------------------

function varargout = Interface(varargin)
% INTERFACE M-file for Interface.fig

% Begin initialization code
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Interface_OpeningFcn, ...
                   'gui_OutputFcn',  @Interface_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code

% --- Executes just before Interface is made visible.
function Interface_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = Interface_OutputFcn(hObject, eventdata,
handles)

% Get default command line output from handles structure
varargout{1} = handles.output;




function triaxiality_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function triaxiality_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```matlab
function fo_Callback(hObject, eventdata, handles)
function fo_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)

mypath=get(handles.e_path,'string');

triaxiality=str2num(get(handles.triaxiality,'string'));
fo=str2num(get(handles.fo,'string'));
dlmwrite([mypath
'\readparameter.txt'],[triaxiality;fo],'newline','pc','precision','%
.5f');
dos(['"' mypath '\CreateVC' '" ' '"' mypath '"']);
Optimisation
function e_path_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function e_path_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)

%-------------------------------------------------------------------
%This MATLAB code includes the optimization function, where initial
%guess, constraints and tolerances are defined.

%Optimisation.m
%-------------------------------------------------------------------

handles=guidata(Interface);
mypath=get(handles.e_path,'string');
myJobname='sh1'

% Get the XY coordinates of the void and cell of the VC model
dos(['"' mypath '\xyReport' '" ' '"' mypath '"']);
dos(['"' mypath '\TotalVolume' '" ' '"' mypath '"']);

%initial guess
p0=[1 1 1]

%first run
figure
runonce0(p0,mypath);

%Constraints
lb=[1.0 0.75 2.0];
```

```matlab
ub=[2.0 1.25 4.0];
clc
A=[];
b=[];
Aeq=[];
beq=[];

%Pattern Search
options=psoptimset('display','iter','MaxIter',100,'PlotFcn',{@psplot
bestf,@psplotbestx});
[p,F,exitflag,output]=patternsearch(@MyCost,p0,A,b,Aeq,beq,lb,ub,[],
options)

% Genetic Algorithm
%options = gaoptimset('display','iter','MutationFcn',
%                     @mutationadaptfeasible,'PlotFcn',
%                     {@gaplotbestf,@gaplotbestindiv}); %
%                                'StallTimeLimit',100,
%x = ga(@MyCost,3,A,b,Aeq,beq,lb,ub,[],options)

%Last run
figure
runonce(p,myJobname,mypath)
q=p
F
dlmwrite('q-history.txt',[p F],'-append','newline','pc')


%-------------------------------------------------------------------
%This MATLAB code will be run only once at the beginning of the
%calibration process to plot the vvf versus time graphs for GT and
%VC cell models for initial guess of [q1=1,q2=1 and q3=1].

%runonce0.m
%-------------------------------------------------------------------

function F=runonce0(p,mypath);

myfile=[mypath '\read_it.txt'];
dlmwrite(myfile,p','newline','pc','precision','%.4f');

%Run bat files
dos(['"' mypath '\CreatePartGT' '" ' '"' mypath '"']);
dos(['"' mypath '\VVFreport' '" ' '"' mypath '"']);

%read GT data
M=dlmread('VvfReport.txt');
timeGT=M(:,1);
VVFGT=M(:,2);

%read VC data
outerfile='ReadTotalVolume.rpt';
innerfile='xyCoordReport.rpt';
[vv,timeVC]=readabaqusreport(innerfile);
[vt,timeVC]=readabaqusreport(outerfile);
VVFVC=vv./vt;
VVFGTi=interp1(timeGT,VVFGT,timeVC);
VVFGTi=VVFGTi';
```

```
plot(timeGT,VVFGT,'o')
hold all
plot(timeVC,VVFVC,'.');
plot(timeVC,VVFGTi,'x');
grid on;
F=norm(VVFGTi-VVFVC);
F
dlmwrite('VVFGTi0.txt',[timeVC
VVFGTi'],'newline','pc','delimiter','\t');

%-------------------------------------------------------------------
%This MATLAB code reads ReadTotalVolume.rpt and XYCoordReport.rpt
%files and calculates the vvf of the VC cell model.

%readabaqusreport.m
%-------------------------------------------------------------------

function [myv,time]=readabaqusreport(filename)

fid=fopen(filename);

for i=1:3
    L=fgetl(fid);
end

l4=fgetl(fid);

Ci=find(L=='N');
NoOfNodes=length(Ci);

for j=1:NoOfNodes
    mystr=L(Ci(j)+3:Ci(j)+6);
    NodeNo(j)=str2num(mystr);
end

myformat='%f';
for i=1:NoOfNodes
    myformat=[myformat ' %f'];
end

A=fscanf(fid,myformat,[NoOfNodes+1 inf]);
B=A';

time=A(1,:)';

NoOfProfiles=length(time);

ix=2:(length(A(:,1))-1)/2+1;
iy=(length(A(:,1))-1)/2+2:length(A(:,1));
for j=1:NoOfProfiles
    P{j}={A(ix,j) A(iy,j) 0 NodeNo(1:NoOfNodes/2)'};
end

y0=P{1,1}{1,2};
[y0sorted,ip]=sort(y0,'descend');

for i=1:NoOfProfiles
    Ps{1,i}{1,1}=P{1,i}{1,1}(ip);
    Ps{1,i}{1,2}=P{1,i}{1,2}(ip);
```

243

```
    Ps{1,i}{1,3}=P{1,i}{1,3};
    Ps{1,i}{1,4}=P{1,i}{1,4}(ip);
end

for i=1:NoOfProfiles
    x=Ps{1,i}{1,1};
    y=Ps{1,i}{1,2};
    area=pi*x.^2;
    v(1)=0;
    for j=2:length(x)
        v(j)=pi*(x(j)^2+x(j-1)^2+x(j)*x(j-1))/3*abs((y(j)-y(j-1)));
    end
    Ps{1,i}{1,3}=sum(v);
end

for i=1:NoOfProfiles
    x=Ps{1,i}{1,1};
    y=Ps{1,i}{1,2};
    %hold all
    %plot(x,y,'.');hold
    %pause
end
for i=1:NoOfProfiles
    myv(i)=Ps{1,i}{1,3};
end

%-----------------------------------------------------------------
%This MATLAB code defines the cost function (see section 4.2.5,
%equation 4.11)

%MyCost.m
%-----------------------------------------------------------------

function F=MyCost(p);
handles=guidata(interface);

%Writing parametrs file
mypath=get(handles.e_path,'string');
myfile=[mypath '\read_it.txt'];
dlmwrite(myfile,p','newline','pc','precision','%.4f');

%Run bat files
dos(['"' mypath '\CreatePartGT' '" ' '"' mypath '"']);
dos(['"' mypath '\VVFreport' '" ' '"' mypath '"']);

%read GT data
M=dlmread('VvfReport.txt');
timeGT=M(:,1);
VVFGT=M(:,2);

%read VC data
outerfile='ReadTotalVolume.rpt';
innerfile='xyCoordReport.rpt';
[vv,timeVC]=readabaqusreport(innerfile);
[vt,timeVC]=readabaqusreport(outerfile);

%grid on;
VVFVC=vv./vt;
VVFGTi=interp1(timeGT,VVFGT,timeVC);
```

```
VVFGTi=VVFGTi';

F=norm(VVFGTi-VVFVC);
dlmwrite('q-history.txt',[p F],'-append','newline','pc')
```

```
%--------------------------------------------------------------------
%This MATLAB code will be run only once at the end of the
%calibration process to plot the vvf versus time graphs for GT and
%VC cell models for the optimized q parameters.

%runonce.m
%--------------------------------------------------------------------

function F=runonce(p,jobname,mypath);
myfile=[mypath '\' 'read_it.txt'];
dlmwrite(myfile,p','newline','pc','precision','%.4f');

%Run bat files
dos(['"' mypath '\CreatePartGT' '" ' '"' mypath '"']);
dos(['"' mypath '\VVFreport' '" ' '"' mypath '"']);

%read GT data
M=dlmread('VvfReport.txt');
timeGT=M(:,1);
VVFGT=M(:,2);

%read VC data
outerfile='ReadTotalVolume.rpt';
innerfile='xyCoordReport.rpt';
[vv,timeVC]=readabaqusreport(innerfile);
[vt,timeVC]=readabaqusreport(outerfile);
VVFVC=vv./vt;
VVFGTi=interp1(timeGT,VVFGT,timeVC);
VVFGTi=VVFGTi';
plot(timeGT,VVFGT,'o')
hold all
plot(timeVC,VVFVC,'.');
plot(timeVC,VVFGTi,'x');
grid on;
F=norm(VVFGTi-VVFVC);
dlmwrite('VVFVC.txt',[timeVC
VVFVC'],'newline','pc','delimiter','\t');
dlmwrite('VVFGTi.txt',[timeVC
VVFGTi'],'newline','pc','delimiter','\t');
myfolder=[mypath '/' jobname]
mkdir(myfolder);
copyfile('VVFVC.txt',[myfolder '/VVFVC.txt'])
copyfile('VVFGTi.txt',[myfolder '/VVFVGTi.txt'])
copyfile('VVFGTi0.txt',[myfolder '/VVFGTi0.txt'])
```

Appendix C: Pattern Search and Genetic Algorithm parameters

i) Pattern Search

Option	Description	Values
Cache	With `Cache` set to `'on'`, `patternsearch` keeps a history of the mesh points it polls and does not poll points close to them again at subsequent iterations. Use this option if `patternsearch` runs slowly because it is taking a long time to compute the objective function. If the objective function is stochastic, it is advised not to use this option.	`'on'` | `{'off'}`
CacheSize	Size of the history	Positive scalar | `{1e4}`
CacheTol	Positive scalar specifying how close the current mesh point must be to a point in the history in order for `patternsearch` to avoid polling it. Use if `'Cache'` option is set to `'on'`.	Positive scalar | `{eps}`
CompletePoll	Complete poll around current iterate	`'on'` | `{'off'}`
CompleteSearch	Complete search around current iterate when the search method is a poll method	`'on'` | `{'off'}`
Display	Level of display	`'off'` | `'iter'` | `'diagnose'` | `{'final'}`
InitialMeshSize	Initial mesh size for pattern algorithm	Positive scalar | `{1.0}`
InitialPenalty	Initial value of the penalty parameter	Positive scalar | `{10}`
MaxFunEvals	Maximum number of objective function evaluations	Positive integer | `{2000*numberOfVariables}`
MaxIter	Maximum number of iterations	Positive integer | `{100*numberOfVariables}`
MaxMeshSize	Maximum mesh size used in a poll/search step	Positive scalar | `{Inf}`
MeshAccelerator	Accelerate convergence near a minimum	`'on'` | `{'off'}`
MeshContraction	Mesh contraction factor, used when iteration is unsuccessful	Positive scalar | `{0.5}`
MeshExpansion	Mesh expansion factor, expands mesh when iteration is successful	Positive scalar | `{2.0}`
MeshRotate	Rotate the pattern before declaring a point to be optimum	`'off'` | `{'on'}`
OutputFcns	Specifies a user-defined function that an optimization function calls at each iteration	Function handle or cell array of function handles | `{[]}`
PenaltyFactor	Penalty update parameter	Positive scalar | `{100}`
PlotFcns	Specifies plots of output from the pattern search	`@psplotbestf` | `@psplotmeshsize` | `@psplotfuncount` | `@psplotbestx` | `{[]}`
PlotInterval	Specifies that plot functions will be called at every interval	`{1}`
PollingOrder	Order of poll directions in pattern search	`'Random'` | `'Success'` | `{'Consecutive'}`

PollMethod	Polling strategy used in pattern search	`{'GPSPositiveBasis2N'}	'GPSPositiveBasisNp1'	'GSSPositiveBasis2N'	'GSSPositiveBasisNp1'	'MADSPositiveBasis2N'	'MADSPositiveBasisNp1'`
ScaleMesh	Automatic scaling of variables	`{'on'}| 'off'`
SearchMethod	Type of search used in pattern search	`@GPSPositiveBasis2N	@GPSPositiveBasisNp1	@GSSPositiveBasis2N	@GSSPositiveBasisNp1	@MADSPositiveBasis2N	@MADSPositiveBasisNp1	@searchga	@searchlhs	@searchneldermead	{[]}`
TimeLimit	Total time (in seconds) allowed for optimization	Positive scalar`|{Inf}`
TolBind	Binding tolerance	Positive scalar`|{1e-3}`
TolCon	Tolerance on constraints	Positive scalar`|{1e-6}`
TolFun	Tolerance on function, stop if both the change in function value and the mesh size are less than `TolFun`	Positive scalar`|{1e-6}`
TolMesh	Tolerance on mesh size	Positive scalar`|{1e-6}`
TolX	Tolerance on variable, stop if both the change in position and the mesh size are less than `TolX`	Positive scalar`|{1e-6}`
UseParallel	Compute objective functions of a poll or search in parallel	`'always'|{'never'}`
Vectorized	Specifies whether functions are vectorized, see <u>Vectorize the Objective and Constraint Functions</u>	`'on'|{'off'}`

ii)   **Genetic Algorithm (GA)**

Option	Description	Values
CreationFcn	Handle to the function that creates the initial population	@gacreationuniform | @gacreationlinearfeasible
CrossoverFcn	Handle to the function that the algorithm uses to create crossover children	@crossoverheuristic | {@crossoverscattered} | @crossoverintermediate | @crossoversinglepoint | @crossovertwopoint | @crossoverarithmetic
CrossoverFraction	The fraction of the population at the next generation, not including elite children, that is created by the crossover function	Positive scalar | {0.8}
Display	Level of display	'off' | 'iter' | 'diagnose' | {'final'}
DistanceMeasureFcn	Handle to the function that computes distance measure of individuals, computed in decision variable or design space (genotype) or in function space (phenotype)	{@distancecrowding,'phenotype'}
EliteCount	Positive integer specifying how many individuals in the current generation are guaranteed to survive to the next generation. Not used in gamultiobj.	Positive integer | {2}
FitnessLimit	Scalar. If the fitness function attains the value of FitnessLimit, the algorithm halts.	Scalar | {-Inf}
FitnessScalingFcn	Handle to the function that scales the values of the fitness function	@fitscalingshiftlinear | @fitscalingprop | @fitscalingtop | {@fitscalingrank}
Generations	Positive integer specifying the maximum number of iterations before the algorithm halts	Positive integer |{100}
HybridFcn	Handle to a function that continues the optimization after ga terminates or Cell array specifying the hybrid function and its options structure	Function handle | @fminsearch | @patternsearch | @fminunc | @fmincon | {[]} or 1-by-2 cell array | {@solver, hybridoptions}, where solver = fminsearch, patternsearch, fminunc, or fmincon {[]}
InitialPenalty	Initial value of penalty parameter	Positive scalar | {10}
InitialPopulation	Initial population used to seed the genetic algorithm; can be partial	Matrix | {[]}
InitialScores	Initial scores used to determine fitness; can be partial	Column vector | {[]}
MigrationDirection	Direction of migration	'both' | {'forward'}
MigrationFraction	Scalar between 0 and 1 specifying the fraction of individuals in each	Scalar | {0.2}

	subpopulation that migrates to a different subpopulation	
MigrationInterval	Positive integer specifying the number of generations that take place between migrations of individuals between subpopulations	`Positive integer | {20}`
MutationFcn	Handle to the function that produces mutation children	`@mutationuniform | @mutationadaptfeasible | {@mutationgaussian}`
OutputFcns	Functions that ga calls at each iteration	`Function handle or cell array of function handles | {[]}`
ParetoFraction	Scalar between 0 and 1 specifying the fraction of individuals to keep on the first Pareto front while the solver selects individuals from higher fronts	`Scalar | {0.35}`
PenaltyFactor	Penalty update parameter	`Positive scalar | {100}`
PlotFcns	Array of handles to functions that plot data computed by the algorithm	`@gaplotbestf | @gaplotbestindiv | @gaplotdistance | @gaplotexpectation | @gaplotgeneology | @gaplotmaxconstr | @gaplotrange | @gaplotselection | @gaplotscorediversity | @gaplotscores | @gaplotstopping`
PlotInterval	Positive integer specifying the number of generations between consecutive calls to the plot functions	`Positive integer | {1}`
PopInitRange	Matrix or vector specifying the range of the individuals in the initial population	`Matrix or vector | [0;1]`
PopulationSize	Size of the population	`Positive integer | {20}`
PopulationType	String describing the data type of the population	`'bitstring' | 'custom' | {'doubleVector'} Note that linear and nonlinear constraints are not satisfied when PopulationType is set to 'bitString' or 'custom'.`
SelectionFcn	Handle to the function that selects parents of crossover and mutation children	`@selectionremainder | @selectionuniform | {@selectionstochunif} | @selectionroulette | {@selectiontournament}`
StallGenLimit	Positive integer. The algorithm stops if there is no improvement in the objective function for StallGenLimit consecutive generations.	`Positive integer | {50}`
StallTimeLimit	Positive scalar. The algorithm stops if there is no improvement in the objective function for StallTimeLimit seconds.	`Positive scalar | {Inf}`
TimeLimit	Positive scalar. The algorithm stops after running for TimeLimit seconds.	`Positive scalar | {Inf}`
TolCon	Positive scalar. TolCon is used to determine the feasibility with respect to	`Positive scalar | {1e-6}`

	nonlinear constraints.	
`TolFun`	Positive scalar. The algorithm runs until the cumulative change in the fitness function value over StallGenLimit is less than TolFun.	`Positive scalar	{1e-6}`
`UseParallel`	Compute fitness functions of a population in parallel.	`'always'	{'never'}`
`Vectorized`	String specifying whether the computation of the fitness function is vectorized	`'on'	{'off'}`

Appendix D: Where UMAT fits into ABAQUS/Standard

Figure below shows the basic flow of data and actions from the start of an ABAQUS/Standard analysis to the end of a step and indicates where a UMAT is fit into ABAQUS / Standard. It is important to note that the UMAT is run at each increment and for each iteration. In this figure a circle signifies a decision point in the code or a specific state (i.e. beginning of an increment) during the analysis and a rectangle signifies an action that is taken during the analysis.
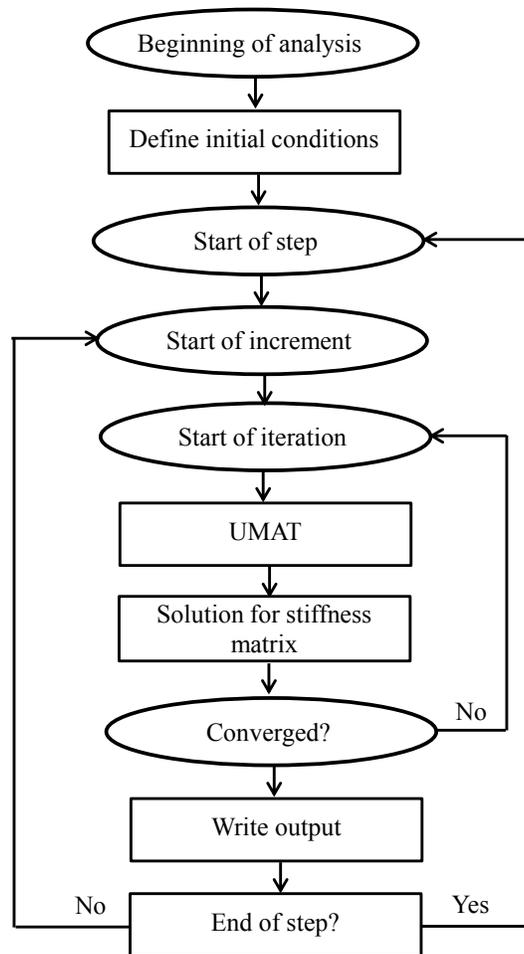
***Figure D1**- Flow of data and actions from the start of an ABAQUS/Standard analysis to the end of a step[1].*

[1] (ABAQUS training manual, Writing User Subroutine with ABAQUS, available at http://imechanica.org/files/Writing%20User%20Subroutines%20with%20ABAQUS.pdf)

Appendix E: The void elimination UMAT (FORTRAN code), the Gurson UMAT provided by TATA is not included.

```fortran
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C     UMAT for VOID ELIMINATION MODEL
C
C     MYHT: Healing time,MTSEQ: equivalent stress,MYP: hydrostatic
C     pressure
C     MYHPERCENT: healing percentage, CPERCENT: closure percentage,
C     TIMECLOS: closure time
C     MYELS: element status, MYINDEX: element index
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,STRAN,DSTRAN,TIME,DTIME,
     2 TEMP,DTEMP,PREDEF,DPRED,CMNAME,NDI,NSHR,NTENS,
     3 NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,CELENT,
     4 DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)

      INCLUDE 'aba_param.inc'

      CHARACTER*8 CMNAME

      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1     DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2     STRAIN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3     PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3)

C     PROPERTIES
      DIMENSION MYRES(PROPS(22),6)
      PARAMETER (ZERO=0.000000000)

      REAL*8 TOLD,TNEW
      REAL*4 MYHT
      REAL MYP,MYVVFF,MYHPERCENT,CPERCENT
      REAL MYf_0,MYDOTEPS_0, TIMECLOS ,DelT
         s
      INTEGER TOTALELEM,COUNT,MYELS,MYINDEX,NCALL


      SAVE MYINDEX, MYHPERCENT, CPERCENT, NCALL,TOLD

      MYDOTEPS_0 = PROPS(6)     !strain rate
      MYf_0      = PROPS(11)    !initial VVF
      TOTALELEM  = PROPS(22)    !total number of elements
      MYVVFF     = PROPS(23)    !VVF at closure
      DelT       =DTIME

      IF (STATEV(1).EQ.ZERO) THEN

        STATEV(1) =1
        STATEV(3) =MYf_0    !void volume fraction
        STATEV(6) =ZERO     !element status
        STATEV(7) =ZERO     !TIMECLOS
        STATEV(8) =ZERO
        STATEV(9) =ZERO     !MYHT
        MYHPERCENT=ZERO
```

```
        CPERCENT  =ZERO
        NCALL     =1.0
    ENDIF

    IF (MYHPERCENT.GE.100.0) THEN
        PRINT *,' '
        PRINT *,'HEALING COMPLETED ... '
        PRINT *,' '
    ENDIF

    MYINDEX =MYINDEX+1
    MYHT    =STATEV(9)
    TIMECLOS=STATEV(7)

    IF (MYINDEX.GT.TOTALELEM) MYINDEX=1

    TNEW=TIME(2)

    IF (MYINDEX.EQ.8.0) TOLD=TNEW       ! 8 integration points

    IF ((TNEW.GT.TOLD).AND.(MYINDEX.EQ.1)) THEN
        NCALL=1
    ELSE
        IF ((MYINDEX.EQ.1).AND.(TOLD.EQ.TNEW)) NCALL=NCALL+1
    ENDIF

    IF((STATEV(3).GT.ZERO).AND.(STATEV(3).LE.MYVVFF)
   1    .AND.(MYHPERCENT.LT.100.0).AND.(STATEV(6).NE.2.0))THEN

            CALL PRINCIPALSTRESS(STRESS,MYP,NTENS)

            IF (STATEV(8).NE.1.0) THEN
                STATEV(8)=1.0
                TIMECLOS=TIME(2)
                STATEV(6)=1.0 ! Element closed

                IF (NCALL.EQ.1.0)
   1                CPERCENT =CPERCENT+(1.0/TOTALELEM)*100.0

                CALL HealTime(MYP,MYDOTEPS_0,MYHT,MYVVFF,MYINDEX)

                STATEV(9)=MYHT
                STATEV(7) =TIMECLOS
            ENDIF

            CALL FindF(MYP,MYDOTEPS_0,MYVVFF,STATEV(3),DelT)

            IF ((TIME(2).GE.((MYHT+TIMECLOS))).AND.
   1            (STATEV(6).EQ.1.0)) THEN
                 STATEV(6)=2.0 ! Element healed
                 STATEV(3)=ZERO
                 IF(NCALL.EQ.1.0)
   1                 MYHPERCENT=MYHPERCENT+(1.0/TOTALELEM)*100.0
            ENDIF
            OPEN(UNIT=311,FILE=
   1        'G:/umat13/HP.TXT'
   2        ,FORM='FORMATTED')
            IF(NCALL.EQ.1.0) WRITE(311,*)'HP1=',MYHPERCENT,'CP=',
   1                        CPERCENT,'Time=',Time(2),'P=',MYP
```

253

```fortran
      ENDIF

      CALL UMAT2(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1        RPL,DDSDDT,DRPLDE,DRPLDT,STRAN,DSTRAN,TIME,
     2        DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,NDI,
     3        NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     4        CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KIN)

      RETURN
      END
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE PRINCIPALSTRESS(STRESS,MYP,NTENS)

C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      REAL II,III,Q,R,S,T,MYP
      COMPLEX STRESSP1, STRESSP2, STRESSP3
      REAL*8 I,STRESS(NTENS)

      I = STRESS(1)+STRESS(2)+STRESS(3)
      II = STRESS(1)*STRESS(2)+STRESS(2)*STRESS(3)+
     1 STRESS(1)*STRESS(3)-(STRESS(4)**2+STRESS(5)**2+STRESS(6)**2)
      III = STRESS(1)*STRESS(2)*STRESS(3)+2*STRESS(4)*STRESS(5)*
     1 STRESS(6)-(STRESS(1)*STRESS(4)**2+STRESS(2)*STRESS(5)**2+
     2 STRESS(3)*STRESS(6)**2)

      Q = (3*II-I**2)/9
      R = (-9*I*II+27*III+2*I**2)/54
      S = (R+(Q**3+R**3)**(0.5))**(1/3)
      T = (R-(Q**3+R**3)**(0.5))**(1/3)

      STRESSP1 = S+T+(1/3)*I
      STRESSP2 = (-1/2)*(S+T)+(1/3)*I+(1/2)*i*3**0.5*(S-T)
      STRESSP3 = (-1/2)*(S+T)+(1/3)*I-(1/2)*i*3**0.5*(S-T)
      MYP = -I/3.0

      END

C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

C     Calculates healing time

      SUBROUTINE HealTime(MYP,MYDOTEPS_0,MYHT,MYVVFF,MYINDEX)
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      INCLUDE 'aba_param.inc'

      Real g,DELVVF,VVF,Y,SUM,DELf,dg,VVF1
      Real r0,T,h0
      Real half
      Real MYP,MYDOTEPS_0,MYHT,MYVVFF,MYf
      PARAMETER (ZERO=0)
      Integer N, I,MYINDEX

      half    = 0.5
      N       = 80.0
      r0      = 5E-3
      T       = 1273.15
```

254

```fortran
      VVF1      = MYVVFF
      h0        = 1E-2
C Calculate subinterval length
C and initialize the approximating SUM and vvf
      DELVVF = -VVF1/REAL(N)
      VVF    = VVF1
      SUM    = 0.0
C Now calculate
      OPEN(UNIT=308,FILE=
     1 'G:/umat13/dfdtp.TXT'
     2 ,FORM='FORMATTED')

      OPEN(UNIT=309,FILE=
     1 'G:/umat13/MYf.TXT'
     2 ,FORM='FORMATTED')

      DO 10 I = 1, N

        MYf = VVF

        Y1 = g(r0,MYf,T,MYP,MYDOTEPS_0,h0)

        DELf = DELVVF

        VVF=VVF+DELVVF
        MYf = MYf + DELf

        IF (I.EQ.N) THEN
            MYf   =1.0E-18
            VVF   =1.0E-18
        ENDIF

        IF (MYf.LT.ZERO) THEN
            MYf=1.0E-10
        ENDIF

        Y2 = g(r0,MYf,T,MYP,MYDOTEPS_0,h0)
        dg = Y2+Y1
        Y=(Y1+Y2)/2.0*DELf
        SUM = SUM + Y

   10 CONTINUE
      MYHT=SUM
      End

C%%%%%%%%%%%%%%%%%%%%%
C The integranl g(f)
C Pilling model
C%%%%%%%%%%%%%%%%%%%%%

      FUNCTION g(r0,MYf,T,MYP,MYDOTEPS_0,h0)

      Real g,denom,denomdf,TOL,half,zero,rh
      Real K,SurfEn,d,Dgbc,Ap,Qp,np,Apl,Qpl,npl,Ra,Qgb,AV
      REAL r0,MYf,T,MYP,MYDOTEPS_0,MYVVF
      Real sigmaz, sigmatheta, sigmar, sigma, Dgb, r, edot, edotr
      Real dfdtp, dfdtd, dfdtt, dvdtd,h0,dtdfd,dtdfp
      Integer Ni
```

255

```fortran
      AV        = 1.78E-29    !Atomic volume
      K         = 1.38E-23    !Boltzman constant
      Dgbc      = 3.0E-12     !Grain boundary diffusion coefficient
      SurfEn    = 1.7         !Surface energy
      d         = 3.0E-5      !Grain size
      half      = 0.50
      zero      = 0.0
      TOL       = 1e-20
      Ap        = 3.80E+4     !Rate constant for plastic flow
      Qp        = 150000.0    !Activation energy for plastic flow
      np        = 1.43        !Stress exponent for plastic flow
      Apl       = 7.5E-2      !Rate constant for creep
      Qpl       = 150000      !Activation energy for plastic flow
      npl       = 4.3         !Creep stress exponent
      Ra        = 8.314       !Gas constant
      Qgb       = 202000.0    !Activation energy
      pi        = 3.141593

      rh=r0*(MYf)**half
      sigmaz=-(MYP*1000000.0)*r0**2/(r0**2-rh**2)/1000000.0
      sigmar=((MYP*1000000.0)-2*SurfEn/r0)*((Myf)**half-1)/
     1      (1-MYf)/1000000.0
      sigmatheta=((-MYP*1000000.0)-2*SurfEn/r0)*((Myf)**half+1)/
     1          (1-MYf)/1000000.0
      sigma=(half*((sigmar-sigmatheta)**2+(sigmatheta-sigmaz)**2+
     1    (sigmaz-sigmar)**2))**0.5

      edot=(Ap*exp(-Qp/(Ra*T))/T)*sigma**np+
     1    (Apl*exp(-Qpl/(Ra*T))/T)*sigma**npl
      edotr=edot/sigma*(sigmar-half*(sigmatheta+sigmaz))
      dfdtp=-2*edotr*(1-MYf)
      denom=Log(1/MYf)-((1-MYf)/2.0)
      Ni=int(1+MYf*h0/d)
      Dgb=Dgbc*exp(-Qgb/(Ra*T))

      IF (denom.LT.TOL) THEN
          denom=TOL
      ENDIF

      dvdtd=-(2*pi*Dgb*(sigmaz*1000000.0)/(K*T)*(1-MYf)/
     1      denom)*AV
      dfdtd=-Ni*(dvdtd/(pi*r0**2*h0*MYf))
      denomdf=(dfdtd+dfdtp)
      IF (denomdf.eq.zero) THEN
          denomdf=1e-20
      ENDIF
      dfdtt=1.0/denomdf
      g=dfdtt
      RETURN
      END
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      SUBROUTINE FindF(MYP,MYDOTEPS_0,MYVVFF,STATEV3,DelT)
C     Calculates void volume fraction from Pilling model.
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      INCLUDE 'aba_param.inc'

      Real r0,T,h0,DelT
      Real MYP,MYDOTEPS_0,MYVVFF
      REAL*8 STATEV3,MYfN,DELTAF
```

```
      MYfN     = STATEV3
      T        = 1273.15
      h0       = 1E-2
      r0       = 5E-3

      DELTAF= DF(MYfN,T,MYP,MYDOTEPS_0,r0,h0,DelT)
      IF (DELTAF.LT.MYfN) THEN
         MYfN=MYfN+DELTAF
      ELSE
         MYfN=0.0
      ENDIF
      STATEV3=MYfN
      RETURN
      END
C%%%%%%%%%%%%%%%%%%%%%%%
C Function DF
C%%%%%%%%%%%%%%%%%%%%%%%

      FUNCTION DF(MYfN,T,MYP,MYDOTEPS_0,r0,h0,DelT)

      Real DF,denom,denomdf,TOL,half,zero
      REAL B1,B2,B,DelT
      Real K,SurfEn,d,Dgbc,Ap,Qp,np,Apl,Qpl,npl,Ra,Qgb,AV
      REAL r0,rh,T,MYP,MYDOTEPS_0,MYVVF
      Real sigmaz, sigmatheta,sigmar,sigma,Dgb,edot,edotr
      Real dfdtp, dfdtd, dfdtt, dvdtd,h0,dtdfd,dtdfp
      REAL*8 MYfN
      Integer Ni

      AV       = 1.78E-29
      K        = 1.38E-23
      Dgbc     = 3.0E-12
      SurfEn   = 1.7
      d        = 2.0E-5
      half     = 0.50
      zero     = 0.0
      TOL      = 1e-20
      Ap       = 3.80E+4
      Qp       = 150000.0
      np       = 1.43
      Apl      = 7.5E-2
      Qpl      = 150000
      npl      = 4.3
      Ra       = 8.314
      Qgb      = 202000.0
      pi       = 3.141593

      rh=r0*(MYfN)**half
      sigmaz=(-MYP*1000000.0)*r0**2/(r0**2-rh**2)/1000000.0
      sigmar=((MYP*1000000.0)-2*SurfEn/r0)*((MyfN)**half-1)/
     1      (1-MYfN)/1000000.0
      sigmatheta=((-MYP*1000000.0)-2*SurfEn/r0)*((MyfN)**half+1)/
     1          (1-MYfN)/1000000.0
      sigma=(half*((sigmar-sigmatheta)**2+(sigmatheta-sigmaz)**2+
     1    (sigmaz-sigmar)**2))**0.5
      edot=(Ap*exp(-Qp/(Ra*T))/T)*sigma**np+
     1      (Apl*exp(-Qpl/(Ra*T))/T)*sigma**npl
      edotr=edot/sigma*(sigmar-half*(sigmatheta+sigmaz))
```

257

```
dfdtp=-2.0*edotr*(1.0-MYfN)
Ni=int(1.0+MYfN*h0/d)
Dgb=Dgbc*exp(-Qgb/(Ra*T))
B1=-Ni*2.0*AV/K     !*Dgb*(sigmaz*1000000.0)
B2=Dgb*(sigmaz*1000000.0)/(T*r0**2.0*h0)
B=B1*B2

denom=Log(1.0/MYfN)-((1.0-MYfN)/2.0)
DF = (B*(1.0-MYfN)/(MYfN*denom)-2.0*edotr*(1.0-MYfN))
1     *DelT
```

RETURN

END

Appendix F:

(i)   Products of rolling

The ingot or the continuous casting product is hot rolled into a bloom or a slab. Blooms have a square cross section greater than 230 cm^2. Slabs have a rectangular cross section greater than 100 cm^2. Blooms and slabs are further rolled into parts such as plate, sheet, strip, coil, billets, bars and rods. Many of these products will be the starting material for subsequent manufacturing operations such as forging, sheet metal working, wire drawing, extrusion, and machining.
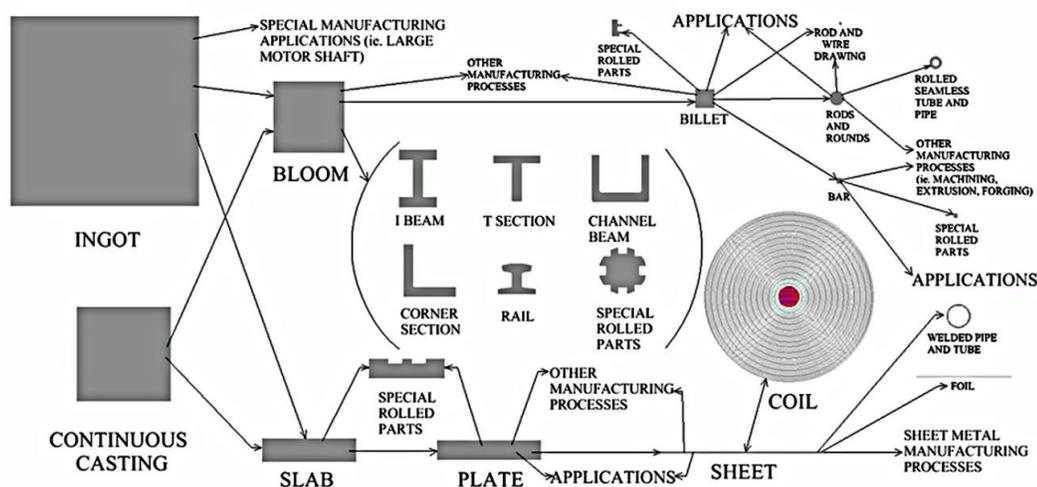

Figure F1*- Metal forming process hierarchy of rolling operations in modern manufacturing industry[2].*

(ii)   Different rolling mill configurations

A roll mill consists of rolls stands, bearings, housing, motors, and other mechanical equipment. Rolling mills are different in the type, number, and position of rolls. Rolling mill arrangements commonly used in manufacturing industry today include the two high mill, the three high mill, the four high mill and the cluster mill as shown in figure below.

---

[2] Image from: The library of manufacturing, available at
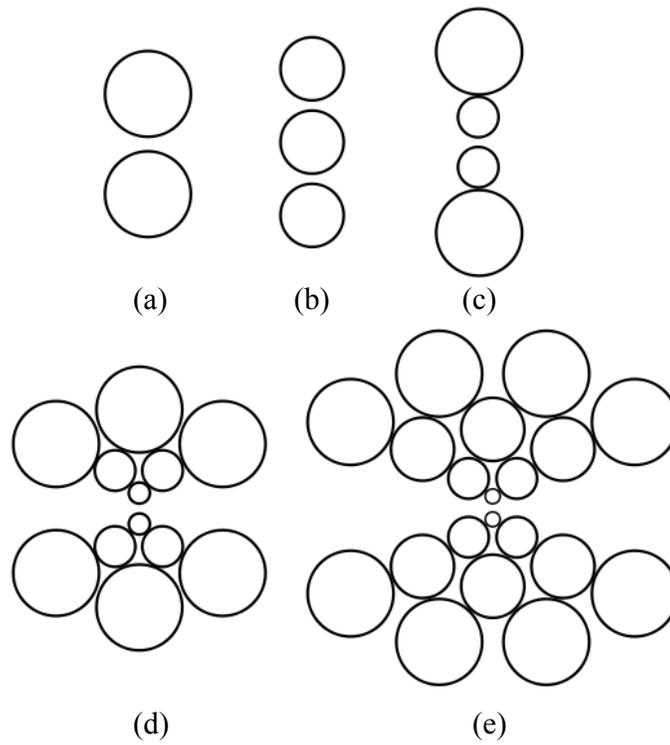http://thelibraryofmanufacturing.com/metal_rolling.html

Figure F2-*Different types of metal rolling mills: (a) Two high rolling mill, (b) Three high rolling mill, (c) four high rolling mill and (d, e) Cluster mills.*

Appendix G: Non uniform porosity distribution (PYTHON Code)

i)   Linear distribution of porosity

```
from abaqus import *
from abaqusConstants import *
import __main__
import section
import regionToolset
import displayGroupMdbToolset as dgm
import part
import material
import assembly
import step
import interaction
import load
import mesh
import job
import sketch
import visualization
import xyPlot
import displayGroupOdbToolset as dgo
import connectorBehavior

mymodelname='B-Roll-3D-25SA'
myinp      ='C:/sa809/Billet.inp'
mycae      ='C:/sa809/Billet.cae'
mypartname ='PRODUCT-1'
myset      ='Set-sa'
fo=0.03

# Read from a cae file
openMdb(pathName=mycae)
mymodel = mdb.models[mymodelname].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=mymodel)
myelements= mymodel.allSets[myset].elements
mynodes   = mymodel.allSets[myset].nodes
ne        = len(myelements)
nn        = len(mynodes)
npe       = len(myelements[0].connectivity)
xmin      = []
xmax      = []
xm        = []
ymin      = []
ymax      = []
ym        = []
nlabels   = []

for k in range(0,nn):
     nlabels.append(mynodes[k].label)
for i in range(0,ne):
     x=[]
     y=[]
     for j in range(0,npe):
          nlabel=myelements[i].connectivity[j]
          nindex=nlabels.index(nlabel+1)
          tmp=mynodes[nindex].coordinates[0]
```

```
            tmpp=mynodes[nindex].coordinates[1]
            x.append(tmp)
            y.append(tmpp)
        xmin.append(min(x))
        xmax.append(max(x))
        xm.append((min(x)+max(x))/2.0)
        ymin.append(min(y))
        ymax.append(max(y))
        ym.append((min(y)+max(y))/2.0)

myfile=open('c:/sa809/NONUNIF.txt', 'w')
f      =[]
c      =min(ym)-(max(ym)-min(ym))/(max(xm)-min(xm))*min(xm)
for i in range(0,ne):
    yl     =(max(ym)-min(ym))/(max(xm)-min(xm))*xm[i]+c
    if ym[i]>=yl:
                ftmp =-fo*(ym[i]-max(ym))/(max(ym)-min(ym))
    else:
                ftmp =-fo*(xm[i]-max(xm))/(max(xm)-min(xm))
    f.append(ftmp)
    myfile.write(str(myelements[i].label)+'    '+str(xm[i])+'
'+str(ym[i])+'    '+str(f[i])+'\n')
myfile.close()
```

i) <u>Gaussian distribution of porosity</u>

```
from abaqus import *
from abaqusConstants import *
import __main__
import section
import regionToolset
import displayGroupMdbToolset as dgm
import part
import material
import assembly
import step
import interaction
import load
import mesh
import job
import sketch
import visualization
import xyPlot
import displayGroupOdbToolset as dgo
import connectorBehavior

mymodelname='B-Roll-3D-25SA'
myinp     ='C:/sa809/Billet.inp'
mycae     ='C:/sa809/Billet.cae'
mypartname ='PRODUCT-1'
myset     ='Set-sa'
fo=0.03
sigx=50.0
sigy=50.0

# Read from a cae file
openMdb(pathName=mycae)
mymodel = mdb.models[mymodelname].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=mymodel)

myelements= mymodel.allSets[myset].elements
mynodes   = mymodel.allSets[myset].nodes
ne        = len(myelements)
nn        = len(mynodes)
npe       = len(myelements[0].connectivity)
xmin      = []
xmax      = []
xm        = []
ymin      = []
ymax      = []
ym        = []
nlabels   = []
for k in range(0,nn):
      nlabels.append(mynodes[k].label)

for i in range(0,ne):
      x=[]
      y=[]
      for j in range(0,npe):
            nlabel=myelements[i].connectivity[j]
            nindex=nlabels.index(nlabel+1)
```

```
            tmp=mynodes[nindex].coordinates[0]
            tmpp=mynodes[nindex].coordinates[1]
            x.append(tmp)
            y.append(tmpp)
        xmin.append(min(x))
        xmax.append(max(x))
        xm.append((min(x)+max(x))/2.0)
        ymin.append(min(y))
        ymax.append(max(y))
        ym.append((min(y)+max(y))/2.0)

myfile=open('c:/temp610/nonunif2/NONUNIF.txt', 'w')
f       =[]
yo      =min(ym)
xo      =min(xm)
for i in range(0,ne):
        ftmp    =fo*exp(-((xm[i]-xo)**2.0/(2.0*sigx**2.0)+(ym[i]-
yo)**2.0/(2.0*sigy**2.0)))
        f.append(ftmp)
        myfile.write(str(myelements[i].label)+'     '+str(xm[i])+'
'+str(ym[i])+'     '+str(f[i])+'\n')
myfile.close()
```

Appendix H:

$$\frac{df_h}{dt} = \underbrace{-2\dot{\varepsilon}_r(1-f_h)}_{I} - N_i \left(\underbrace{\frac{2}{r_o^2 h_o} \frac{D_{gb}\delta\Omega\sigma_z}{KT}}_{B} \frac{(1-f_h)/f_h}{\ln(1/f_h) - (1-f_h)/2} \right)$$

$$\underbrace{\phantom{- N_i \left(\frac{2}{r_o^2 h_o} \frac{D_{gb}\delta\Omega\sigma_z}{KT} \frac{(1-f_h)/f_h}{\ln(1/f_h) - (1-f_h)/2} \right)}}_{II}$$

$$\lim_{f_h \to 0} I = -2\dot{\varepsilon}_r$$

$$\lim_{f_h \to 0} II = \lim_{f_h \to 0} \frac{B(1-f_h)/f_h}{\ln(1/f_h) - (1-f_h)/2} = \lim_{f_h \to 0} \frac{-B/f_h^2}{-1/f_h + 1/2} = \infty$$

$$\lim_{f_h \to 0} \frac{df_h}{dt} = -2\dot{\varepsilon}_r + \infty = \infty \qquad \to \qquad \lim_{f_h \to 0} \frac{dt}{df_h} = 0$$

Appendix I: List of publications

Published:

Afshan, S., Balint, D., Lin, J., Farrugia, D. (2011) Automated calibration of a void closure model for high-temperature deformation, *Journal of Multiscale Modelling*, 3, 79-90

Afshan, S**.**, Balint, D., Lin, J., Farrugia, D. (2013), Micromechanical modelling of void healing, *Advanced Structured Materials*, 19, 1-8.

Afshan, S., Balint, D., Farrugia, D., Lin, J. (2013), New experimental method for identifying the conditions necessary for diffusion bonding in free cutting steels, *Materials Science & Engineering A*, 586, 25-30

Under preparation:

Afshan, S., Balint, D., Farrugia, D., Lin, J. Hybrid Gurson, plasticity and diffusion bonding for micromechanical modelling of void healing, to be submitted to *Computational Materials Science*.