



University of HUDDERSFIELD

University of Huddersfield Repository

Tu, Jiada

Three-Dimensional Finite Element Analysis of Creep Evolution and Damage at Grain Boundary Level

Original Citation

Tu, Jiada (2019) Three-Dimensional Finite Element Analysis of Creep Evolution and Damage at Grain Boundary Level. Doctoral thesis, University of Huddersfield.

This version is available at <http://eprints.hud.ac.uk/id/eprint/35273/>

The University Repository is a digital collection of the research output of the University, available on Open Access. Copyright and Moral Rights for the items on this site are retained by the individual author and/or other copyright owners. Users may access full items free of charge; copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational or not-for-profit purposes without prior permission or charge, provided:

- The authors, title and full bibliographic details is credited in any copy;
- A hyperlink and/or URL is included for the original metadata page; and
- The content is not changed in any way.

For more information, including our policy and submission procedure, please contact the Repository Team at: E.mailbox@hud.ac.uk.

<http://eprints.hud.ac.uk/>



**Three-Dimensional Finite Element Analysis of
Creep Evolution and Damage at Grain Boundary
Level**

Ph.D. Thesis



JIADA TU

**This thesis is submitted to the University of Huddersfield for
the Degree of Doctor of Philosophy**

2019

ABSTRACT

In many high-temperature structural components, creep damage is a non-negligible factor limiting its lifetime. For most alloys, the main reason for the creep damage is due to the cavitation that occurs at grain boundaries, hence it is meaningful to analyze and simulate this phenomenon. In this project, unlike the traditional approach which treats materials as isotropic, the grain and grain boundary is modeled and analyzed separately. Based on this idea, an in-house numerical procedure is developed for the Finite element simulation of creep evolution at the grain boundary level.

The development is under the Continuum Damage Mechanics theoretical framework, through this procedure the traditional solid element with simple power-law adopts to describe the creep deformation evolution of the grain part. The grain boundary part has been modeled by the Goodman element with Markus's cavitation model and Newtonian flow model

The in-house procedure was developed from a Smith's visco-plastic program P61 to solve the creep problem at grain boundary level. The theory and coding implementation of Goodman element (2D/3D) and local-global coordinate transformation techniques are summarized in detail.

This research contributes to the development of the Finite Element procedure for simulating creep evolution at the grain boundary level and provides a new understanding regarding the intrinsic relationship between stress redistribution and creep evolution.

ACKNOWLEDGEMENTS

I would like to thank Dr. Qiang Xu and Prof. Joan Lu for their ongoing guidance and encouragement. The financial support provided by the University of Huddersfield and for being awarded the Vice-chancellor scholarship.

Finally, I would like to thank my wife, my parents, and my colleagues.

Statement of Publication

The work presented in the section 6.3 has been published in the special issue (Creep and High Temperature Deformation of Metals and Alloys) of Metals.

The statement of contribution can be found in the appendix IV.

Publication: Xu, Q., Tu, J. and Lu, Z. (2019). Development of the FE In-House Procedure for Creep Damage Simulation at Grain Boundary Level. Metals, 9(6), p.656.

Contents

Chapter 1 Introduction	1
1.1 Introduction	1
1.2 Aims and Objectives	3
1.3 Thesis Structure	5
Chapter 2 Literature Review of Creep and Creep's Numerical Method	8
2.1 Introduction	8
2.2 Creep and Creep Fracture Mechanisms	10
2.2.1 Creep	10
2.2.2 Creep Fracture Mechanisms	12
2.3 Numerical Method for Creep Damage	14
2.3.1 Continuum Damage Mechanics	15
2.3.2 Objective Oriented Programming	16
2.4 The Review of Finite Element Analysis (FEA)for Creep Damage Analysis at Grain Boundary Level.	17
2.4.1 Review of FE Platform	18
2.4.2 Status of Structure Generation at Grain boundary Level	19
2.4.3 Research Status of FE Modeling at Grain Boundary Level	20
2.5 Cohesive Zone Model	23
2.6 Conventional Solid Element	24
2.6.1 Formulation of 2D Solid Element	24
2.6.2 Formulation of 3D Solid Element	27
2.7 2D Goodman Element	29
2.7.1 Mathematical Background	30
2.7.2 Coordinate System Transmission	33
2.8 3D Goodman Element	34
2.8.1 Mathematical Background	35
2.8.2 Coordinate System Transmission	38
2.9 Creep Constitutive Equation	39
2.9.1 Macro-Creep Constitutive Equations	39
2.9.2 Micro-Creep Constitutive Equations	41
2.10 Removal Function	44
2.11 Restart Facility	45

Chapter 3 Methodology	47
3.1 Introduction	47
3.2 General Methods for Developing In-house Procedure	48
3.3 Integration Method	50
3.3.1 Numerical Integration	50
3.3.2 Numerical Integration for Element.....	51
3.4 Mature Techniques and Legacy Code	52
3.4.1 Storage of Matrix	52
3.4.2 Cholesky Decomposition	53
3.4.3 Legacy Code.....	54
3.5 Creep Non-linear Iteration.....	55
3.6 Mathematical Background of Coordinate Transmission.....	56
3.6.1 2D Coordinate Transmission.....	56
3.6.2 3D Coordinate Transmission.....	56
Chapter 4 Development of the Creep Solver.....	60
4.1 Introduction	60
4.2 Programming the Creep Solver.....	62
4.2.1 Flow Diagram of Solver.....	63
4.2.2 Implementation of Removal Technique	66
4.2.3 Implementation of Restart Facility	70
4.2.4 Implementation of Auto-select Time Step Module.....	72
4.3 Development of Solver for Grain Boundary Level.....	73
4.3.1 Computational Framework	74
4.3.2 Creep Body Loads Generation of the Goodman Element	76
4.3.3 Coding Implementation of Goodman element	76
Chapter 5 Benchmark of the macro scale of In-house Procedure.....	80
5.1 Introduction	80
5.2 The Preliminary Benchmark of the Procedure.....	82
5.2.1 Validation of the Elastic Part.....	86
5.2.2 Validation of the creep part.....	93
5.3 The Validation of the In-house Procedure via the numerical investigation of the Bar 267 Notched-Bar Case Study at 660 °C	98
5.3.1 Introduction	98
5.3.2 Description of the Bar 267 Bridgman Notched Bar Case Study.....	100

5.3.3 Result and Discussion.....	102
5.3.4 Discussion	112
Chapter 6 The Validation and Application of In-house Procedure at Grain Boundary Level	113
6.1 Introduction	113
6.2 The Validation of the In-house Procedure via the numerical investigation of the Bi-grains Case Study.....	115
6.2.1 Introduction	115
6.2.2 Validation of the Elastic Part with quadrilateral element.....	117
6.2.3 Validation of the Elastic Part with triangle element	124
6.2.4 Validation of the Elastic Part of FE model with Angle	128
6.2.5 Validation of the Non-linear Creep Part with quadrilateral element.....	132
6.2.6 Validation of the Non-linear Creep Part with triangle element.....	139
6.2.7 Validation of the Non-linear Creep Part of FE model with Angle	141
6.2.8 Conclusion.....	144
6.3 The Application of the In-house Procedure via the Numerical Investigation of the Polycrystal Case Study	145
6.3.1 FE Model Generation.....	145
6.3.2 Failed element, Stress Field and Creep Damage Evolution	147
6.3.3 Discussion	158
Chapter 7 Development and Validation of the 3-dimensional Procedure at Grain Boundary Level.....	161
7.1 Introduction	161
7.2 Development of 3D Version.....	163
7.2.1 Implementation of Grain Part.....	163
7.2.2 Implementation of Grain Boundary part	164
7.3 Elastic Validation of Grain Part	167
7.3.1 FE Model	167
7.3.2 Validation of Uni-axial Loading.....	170
7.3.3 Validation of Multi-Axial Loading.....	173
7.4 Elastic Validation of Grain Boundary Part.....	175
7.4.1 FE Model	175
7.4.2 Validation of Separated Direction.....	181
7.4.3 Validation of Rotation Status	186

7.5 Validation of the Non-linear Creep	191
7.5.1 The Validation of the Stress Update	191
7.5.2 The Validation of the Integration	193
7.5.3 Error Analysis	195
7.5.4 Conclusion.....	196
Chapter 8 Conclusions and Future Work	197
8.1 Contribution and Conclusion.....	197
8.2 Future Work.....	198
Reference	199
Appendix I the description of library 'Math'	209
Appendix II Source code of Library 'Math'	212
Appendix III Tutorial on INP file.....	299
Appendix IV Publication Contribution List	304

Chapter 1 Introduction

1.1 Introduction

In the past few decades, the operating temperature of power stations has been gradually increased to obtain higher thermodynamic efficiency as a means to satisfy the growing demand for electricity. However, as the temperature rises, the creep damage/fracture is accelerated and the components' lifetime is shortened, which is a key factor limiting the increase in operating temperature. Therefore, it is necessary to study the creep damage/fracture mechanism and develop numerical models of materials for evaluating and predicting the lifetime of components. These assessments are important for the safe operation of a power plants components [1].

In order to investigate and predict the creep failure, a laboratory creep test is usually the most common procedure. However, it has some shortcomings such as high cost and time consumption. Therefore, computer-based numerical Finite Element techniques are becoming more popular, and some research groups and institutes have developed their own in-house procedures or use mature commercial software packages for the study and simulation of creep mechanical behavior. This includes 'FE-DAMAGE' from Nottingham University [2], 'Damage XX' from Manchester University [3], 'DNA' from Louisiana State University [4], and commercial software package 'ABAQU' from Dassault Systèmes [6].

Typically, the object of the finite element simulation is component or specimen, and it is idealized to be a single and uniform material [6]. The description of the creep degradation is achieved by introducing a fictitious damage variable, and the parameters of the constitutive equations are calibrated by the measured rupture time [6-8]. However, according to most experimental observations, the reasons for creep rupture or failure is due to the cavitation at grain boundaries [9], hence researchers have proposed a different modeling idea, which is to develop a constitutive model based on the micro-mechanically motivation independently to describe the creep evolution of grain boundaries.

In creep analysis, the constitutive equations are included in a Finite element method to describe the evolution of the strain, stress, damage over time.

However, traditional solid elements do not have the ability to simulate the mechanical properties of grain boundaries. For example, the deformation of traditional solid elements is measured by strain, while the deformation of grain boundaries is measured by relative displacement [10]. Hence, under the traditional Finite Element framework, a different element needs to be added in to realize grain boundaries' Finite Element Analysis, which is the significance and value of this research.

At present, the Finite Element Analysis of creep at grain boundary level can be calculated using the commercial platform ABAQUS, which is obtained through the external subroutine interface of UMAT and UEL. For instance,

Markus Vose [10] applied UMAT (User-defined Material) within the cavitation constitutive models in ABAQUS's build-in element COH3D8 (8-node 3D cohesive element) for the grain boundary part, and CH. Yu [9] reported the application of UEL (User-defined Element) can be used to develop a non-thickness 2D Goodman element which is made up of two linear rods to describe the creep deformation of the grain boundary part. However, a different approach was chosen in this project to obtain this numerical capability which is the development of an In-house procedure. Firstly, compared with commercial platforms, the development of an in-house procedure provides developers a higher freedom, such as obtaining intermediate output. Secondly, developers do not need to follow numerous complex interface standards, and there are mature frameworks and subroutine libraries that can be adopted, which make the project more efficient and flexible. Therefore, an in-house platform still has advantages and meanings [11].

1.2 Aims and Objectives

This project aims to develop an in-house procedure which is used to implement the Finite Element Analysis and Predict of creep damage evolution of the materials at grain boundary level.

The specific objectives are summarized below:

1. To provide a methodology for developing an in-house Finite Element procedure for creep simulation at grain boundary level. This procedure is

different from the traditional version, with the analysis object being grains and grain boundaries. In finite element modeling, traditional solid elements are adopted to present the grains' mechanical properties, while grain boundaries use non-thickness Goodman elements [12].

2. The existing visco-plastic Finite Element program P61 [13] needs to be refactored to satisfy the computational requirements of using two different element types in one model, and two legacy subroutine libraries 'main' and 'geom' [13] need to be integrated into the procedure to realize the specific required techniques and functions, which include the element spatial discretization, project size evaluation, solid element stiffness matrix obtainment, matrix assembly and storage and solution of the equilibrium equation.
3. Goodman element type needs to be added into this framework, which is a different element type from the traditional solid element. It is a non-thickness contact element and the two surfaces are coincident during the unloading condition. Unlike solid elements, the deformation measure uses relative surface displacement instead of strain, and the mechanical relationship follows the cohesive law [9,10,12].
4. Vöse's cavitation constitutive model [10] and Newton Viscous flow sliding model [14] need to be integrated into the procedure, which use to describe the grain boundaries' creep evolution.

1.3 Thesis Structure

Chapter 1 introduces the importance, significance and aim of this project, and reports the structure of the thesis.

Chapter 2 summarizes the literature background related to this project.

Firstly, an overview of this research significance and the creep phenomenon is presented. Secondly, it presents a review of the current research status of the creep fracture mechanism and the creep numerical method. Finally, it briefly summarizes the mathematical model and the numerical implementation adopted in this project.

Chapter 3 summarizes the general methodology, relevant techniques, and the mathematical method in developing this In-house procedure. Specifically including such techniques as: the developing method used in this procedure development, the numerical integration method for the creep constitutive equations and the element's area, the techniques for the storage of stiffness matrix and the solution of the equilibrium equation, and the generation of the polycrystalline model.

Chapter 4 summarizes the development of a 2-dimensional version of the multi-scale creep Finite Element solver. Specifically, it includes: the procedure structure of the Finite Element simulation for the macro model, the procedure structure for the microstructure simulation at GB level, the mathematical background of the elements for modeling the grain and GB respectively, the development of the program blocks and relevant subroutines for realizing the

mathematical background, and the development of accessibility functions (failed element removal techniques and jump-restart facility).

Chapter 5 summarizes the benchmarks of the macro version of the in-house procedure. The demonstration order corresponds to the development strategy. Firstly, a simple quadrilateral FE model is employed to validate the numerical accuracy and stability of the plane stress version, plane strain version, and axisymmetric version under simple stress conditions. At this stage, the verification of the three-stress state extends from the linear part to the non-linear creep part. Secondly, further verification is to verify the accuracy and reliability of the nonlinear iteration under non-uniform stress conditions. Then, a one quarter notched bar case study is chosen to validate the numerical accuracy of non-linear iteration under complex stress conditions via axisymmetric version procedure.

Chapter 6 summarizes the benchmarks of the micro version of the in-house procedure. The demo strategy is the same as in Chapter 6. Firstly, a Bi-grains FE model is chosen as the initial step to verify the numerical accuracy and stability. In this model, the grain part is meshed by the 3-node triangle solid element or the 4-node quadrilateral element, the GB part is meshed by the 4-node Goodman element. The Bi-grain structures of the two combinations (Triangle element with Goodman element and Quadrilateral element with Goodman element) were verified separately. Secondly, it demonstrates the application of the in-house procedure in the simulation of the creep evolution

with a 20 grains' polycrystal case study, and the results have been published in the special issue (Creep and High Temperature Deformation of Metals and Alloys) of Metals.

Chapter 7 summarizes the development and validation of a 3-dimensional version of the creep Finite Element solver at the GB level. In this chapter, it first introduces the development of the creep solver, followed by presenting a validation of the procedure.

Chapter 8 presents the conclusions and suggestions for further work.

Chapter 2 Literature Review of Creep and Creep's Numerical Method.

2.1 Introduction

This chapter reports the literature review and the current research state related to this project, the information of each section is reviewed below in detail, which includes:

- 1) The general overview of the research significance, creep phenomenon, and the status of the creep fracture mechanisms are presented in section 2.3. In this section, it first introduces the mechanism of creep fracture, and then discusses the research status of creep mechanism at multi-scales, which is discussed from macro and micro scales, respectively.
- 2) In section 2.3, it reviews the development of the Continuum Damage Method for creep damage analysis, and it further reviews the advantages of the OOP method in Finite Element software development.
- 3) The current research state of creep numerical implementation at grain boundary level is presented in section 2.4. In this section, we first discuss the current state of the FE platform, from the two aspects of commercial software and in-house procedures, and then introduces and analyzes the state of the microscopic creep simulation, including the method of polycrystalline structure generation and the implementation of microscopic creep FE algorithms, etc.

- 4) In Section 2.5, a literature review of the Cohesive Zone Model theory is reported. In Section 2.6, the mathematical background of 2D and 3D traditional solid units is briefly reported.
- 5) The brief discussion of the 2D and 3D Goodman element's background is presented in section 2.7 and section 2.8 respectively. In both sections, they discuss the mathematical background and the coordinate transmission system.
- 6) In section 2.9, it reports the creep constitutive equations used in this project. It includes the equations for the description of macroscopic creep mechanism: K-R, KRH, and KRH-X. And the constitutive equations for the description of microscopic creep evolution of grain boundary part.
- 7) In Section 2.10 and 2.11, it reports the literature review of the restart function and auto-time step function.

2.2 Creep and Creep Fracture Mechanisms

2.2.1 Creep

In much practical application, creep deformation/damage is a key factor that limits component life-time, and it is a non-negligible problem for the high-temperature application. Creep occurs when the component is subjected to a continuum constant load, in essence, it is a kind of visco-plastic deformation accumulated over time [16,17]. At room temperature, the process is extremely slow, and it often takes years to have visible deformation. However, when the operating temperature exceeds one-third of the melting point, the deformation will be accelerated, which makes the rapidly shortens the process [17]. Generally, the process can be divided into three stages: Primary, Secondary, and Tertiary, as shown in the typical creep curves (Figure 2.2.1).

The creep characteristics of these three stages are summarized below:

Primary Stage:

At this stage, initially, it starts with a high strain rate and then decreases to a constant value [18]. The final deformation is determined by two main reasons [19]: time-dependent strain hardening and creep recovery. The relationship between these two reasons is competition. The Strain hardening contributes to irreversible creep strain and creep recovery reduces this effect. Under unloading condition or high initial creep strains, the creep recovery plays a main role, and conversely is strain hardening.

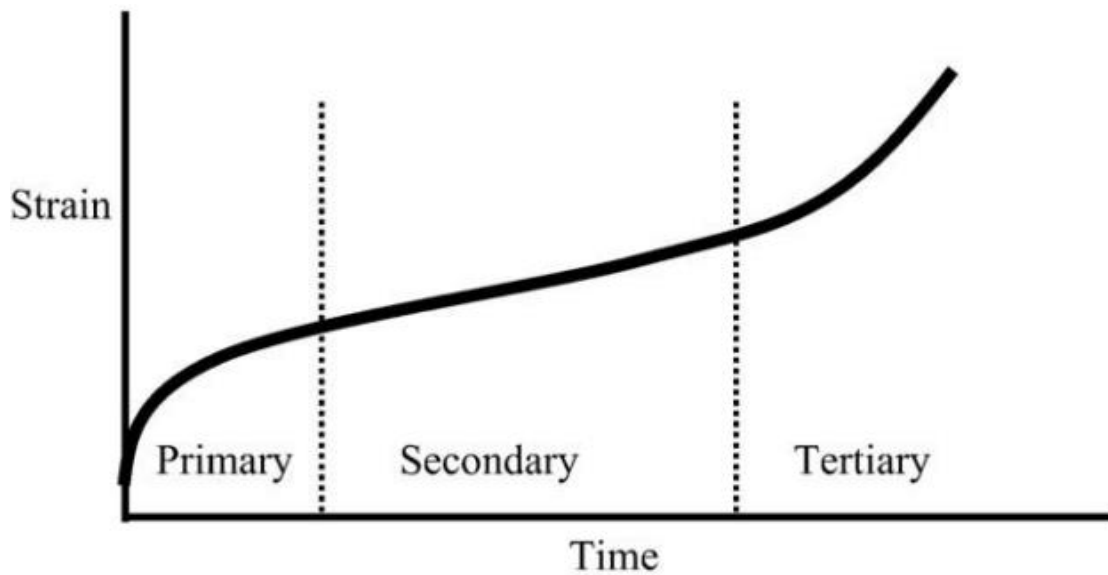


Figure 2.2.1 Basic Creep Curve with three stages. Produced by [20]

Secondary Stage:

At this stage, the creep deformation/strain rate stabilizes [19], and the main factors affecting the creep strain rate are temperature and external load. Many researchers rely on empirical formulas to quantitatively describe the relationship between these three variables [20-23]. The mechanisms that cause creep deformation at this stage are generally considered to be climb softening in the higher temperature range, gliding softening in the lower temperature range, dislocation and diffusion.

Tertiary Stage:

During this stage, the strain rate increases rapidly and failure finally occurs [24].

The main cause of failure is due to the rapid accumulation of plastic strain, which leads to microspores and cracks to form on the grain boundaries [26,26]. Usually, the macro performance at this stage is that the specimen

begins to shrink and the cross-section is reduced, which the actual load on this surface increases rapidly [27]. It accelerates the creep deformation and the rupture happens finally.

2.2.2 Creep Fracture Mechanisms

Dislocation and diffusion are the main deformation forms of creep and they have different mechanisms [28]. The former is due to the external load that makes the material prone to dislocating slip capability, while the latter is due to the diffusion of vacancies through their crystal lattice. These two forms contribute to deformation together during the whole process.

Dislocation Creep

External loading is the main reason, under the condition of high stress and being in an environment with a temperature between 0.3 and 0.7 T_m (Melting Temperature), the movement occurs between two adjacent crystal lattices.

And the higher the temperature and external loading, the greater the amount of dislocation [16,29].

There are two forms of dislocation: edge and screw [30].

➤ Edge Dislocation

It is a material defect, which is caused by the discontinuous of the atomic plane in the middle of the grain, as shown in Figure 2.2.2 (a). When stress is applied on one side of the plane, atoms adjacent to the defect pass through and breaks its normal atoms layer, and, and eventually combine with other atomic layers.

The Burgers vector is parallel to the crystal lattices.

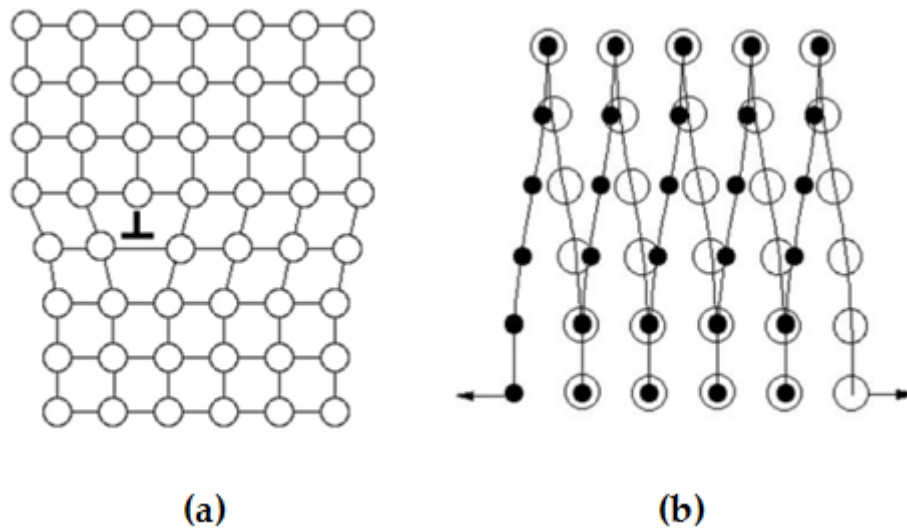


Figure 2.2.2 (a) the edge dislocation; (b) the screw dislocation.

Procedure by [31]

➤ Screw dislocation [30]

It is formed by a defect line (dislocation line) which is perpendicular to crystal lattices. The atomic plane closest to this defect line is the jump one lattice point along the direction, which is perpendicular to the Burgers vector, as shown in Figure 2.2.2(b).

Diffusion Creep

There are two forms, N-H(Nabarro-Herring) and Coble. The main mechanism is the diffusion of vacancies into grains through the crystal lattice [32].

➤ N-H diffusion

The main feature is the diffusion along the crystal lattice, and it is easy to occur in higher stress and temperature regions [33]. (as shown in Figure 2.2.3(a))

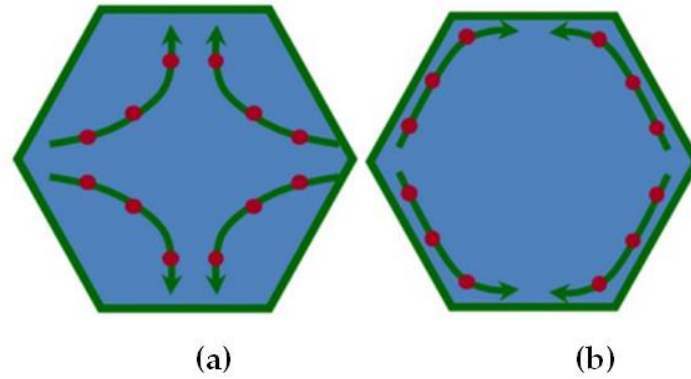


Figure 2.2.3 (a) the N-H diffusion; (b) the Cobal diffusion.
Produced by [34]

➤ Cobal diffusion

Unlike the former, the diffusion direction is along the grain boundaries, and it easily occurs in low stress and temperature regions [33,34].

In summary, the relationship between these two mechanisms (dislocation and diffusion) are competitive. Under high stress conditions, dislocation plays a main role in creep deformation, whereas diffusion plays a major role.

2.3 Numerical Method for Creep Damage

The general numerical method of analysis for the creep is based on the observation and analysis of experiments to establish creep constitutive equations to depict and predict the creep evolution and failure, which was first proposed by Kachanov [83]. The theoretical background is Continuum Damage Mechanics and a damage variable is introduced to present the level of rupture state of the specimen [83,84]. Initially, the material is in a virgin state and the damage variable is zero, under the action of external load and temperature, the damage variable increases monotonically and reaches to the critical value finally.

In 1975, Hayhurst [85] also combined the CDM with Finite Element technology, to present the relationship between stress and damage variable during the entire creep process and found the significance of stress redistribution in the tertiary stage.

2.3.1 Continuum Damage Mechanics

Continuous damage mechanics (CDM) provides a continuous level of description of the material's macro-fracture. In order to describe the degradation mechanism of creep, based on experimental observation, many CDM models have been proposed.

The KR (Kachanov-Rabotnov) equation was originally proposed by Hayhurst in 1972 [79], which was the first time to associate the damage variable with the multi-axial-stress-state. Based on the observation of the experiment, Hayhurst identified the rupture criterion by the Von-Mises equivalent stress and the maximum principles of stress together. In 1975, Hayhurst also combined the CDM with Finite Element technology to present the relationship between the stress and damage variables during the entire creep process and found the significance of stress redistribution in the tertiary stage [80].

Follow this FE application of CDM to the creep problem, Dyson and Osgerby developed a new creep constitutive equation in 1993 [53], which is based on the mechanism of particle Hardening of alloys and it has the capability of predicting the rupture time, minimum creep strain rate, and the strain hardening. Besides, the hyperbolic sine law was introduced to replace the

power law, which increases the simulation result fit to an experiment test better.

In 1996, Hayhurst developed KRH form from Dyson's equation to obtain the FE computational capability under the multi-axial stress condition to do the numerical simulation of 0.5Cr-0.5Mo-0.25V ferritic steel, in which the damage critical value was calibrated by rupture time [58]. However, the limitation of this equation is that the parameter calibration only considers life consistency and ignores the creep strain. Therefore, an inaccurate prediction may occur.

Due to the defect, Q.Xu added two functions, f_1 and f_2 to determine the damage rate together, f_1 applies to couple creep damage with the strain and lifetime at failure and f_2 applies to describe the relationship between stress state and damage [51].

Although the virtual damage variable is introduced into the constitutive equation to describe the creep degradation process, the physical meaning of the damage value is always weak. Since the damage value has no physical meaning, it is only determined by the measured failure time or creep strain. Therefore, some researchers describe the real creep degradation mechanism by establishing the creep constitutive equation of the grain boundary part, which will be introduced in detail in section 2.4.

2.3.2 Objective Oriented Programming

Objective Oriented Programming is a programming paradigm that uses classes as objects to design applications and computer programs to improve

reusability, flexibility, and extensibility [86]. Compared with the traditional method that the main program is composed of a set of functions or a series of instructions directly, OOP allows the program to contain various independent but callable objects.

Compared with traditional programming methods, OOP has obvious advantages. First, it increases the flexibility and maintainability of the program. Second, it improves the development efficiency by separating the developing progress into blocks and applies the legacy codes to accelerate development, as it has modular characteristics and rich libraries of objects [87]. Thus, OOP is adopted for developing this in-house procedure in this project.

2.4 The Review of Finite Element Analysis (FEA)for Creep Damage Analysis at Grain Boundary Level.

Currently, FEA has been widely used to simulate and predict creep damage behavior. There are two ways to obtain this computational capability, one is based on standard commercial software packages, and the other is through the development of in-house procedures. The review of platforms is presented in Section 2.4.1, and Sections 2.4.2 and 2.4.3 give the current state in FEA of creep at grain boundary level.

2.4.1 Review of FE Platform

➤ Commercial Platform

The current mainstream commercial platforms are ABAQUS and ANSYS, and they are applied in a wide range of research areas, including fluid mechanics, electromagnetic field, heat conduction, and geotechnical mechanics, etc.

However, ABAQUS is usually chosen to solve the creep problem. The main reason is that it has a better non-linear solver, which has advantages in solving material, geometrical, and state nonlinearity.

➤ In-house Platform

Three in-house procedures have been developed for solving creep problem, *FE-DAMAGE* [2], *DAMAGE XX /DAMAGE XXX* [3], and *DNA* [4]. The development of these procedures is based on Continuum Damage Mechanic. For solving creep problems, the specific in-house procedure still has its advantages, including:

1. If material constitutive models or element types are not available in ABAQUS library, the only way to write-in these demands is by using UMAT or UEL subroutines. However, the connect interface standards and rules between ABAQUS and subroutines are complex and numerous, which makes the process complicated and inefficient.
2. It provides a higher freedom for developers and it makes the process efficient and controllable [11].

2.4.2 Status of Structure Generation at Grain boundary Level

There are three common methods of generating micro-structures, which can be summarized: Regular Simplified Morphology method, Voronoi Algorithm method, and OOF (Object-oriented-finite method).

➤ Regular Simplified Morphology Method

This method was commonly adopted in the initial modeling to simplify the modeling process, the grain is depicted by regular shapes, such as cubes, dodecahedra, truncation, and octahedrons. Although it reduces the computational complexity, it also decreases the revivification of the actual structure. This way is employed in the initial benchmark of this project, to verify the procedure's accuracy and stability of the procedure with higher efficiency [36].

➤ Voronoi algorithm method

It is applied to partition the space by Thiessen polygons [36]. It is currently used to generate polycrystalline structures, and mature software packages have been developed to implement this algorithm, such as Qhull [37], Voro++ [38], and Neper [39]. In this project, Neper is selected as the pre-processor to build the microstructure for three reasons:

1. Neper integrates the meshing function, which simplifies the modeling process.
2. Neper has a remeshing function, which can be used to insert a non-thickness element between two adjacent grains. Therefore, no additional

action is required to satisfy the requirements for grain boundaries' independent modeling.

3. Neper has the ability to adjust the morphological characteristics of grains. By using different types of mosaics, such as Poisson-Voronoi mosaic, hard core-Voronoi mosaic, centroid-Voronoi mosaic, and Laguerre-Voronoi mosaic, all used obtain a variety of microstructure.

➤ OOF

It was first proposed by ITL (Information Technology Laboratory) and MML (Material Measurement Laboratory) of NIST (National Institute of Standards and Technology, US). The object is the actual microscopic picture or the 3D scan slice picture. Compared with the previous two methods, it has the highest level of revivification. However, the higher computational cost and expensive micro-scan equipment are required [40].

2.4.3 Research Status of FE Modeling at Grain Boundary Level

Under the analysis framework at grain boundary level, homogeneous materials are modeled by grains and grain boundaries separately.

For Grain Boundary Modeling

Initially, grain boundaries were represented by inserting a fictitious elastic spring layer between two adjacent grains, and a new creep fracture model was proposed to describe the cavity evolution [41]. In 1997, Onck and Van der Giessen followed this method to study and simulate the intergranular crack growth [42]. In 2012, in the creep fracture study of polycrystalline ceramic,

Goodman element was introduced to describe the creep evolution of grain boundaries [9]. In 2014, in the creep study of Copper-Antimony-Alloy, the COH3D8 from ABAQUS build-in element library was adopted to describe the mechanism of grain boundary part.

➤ Fictitious Layer

In this method, grain boundaries are represented by fictitious elastic layers, as illustrated in Figure 2.3.1.

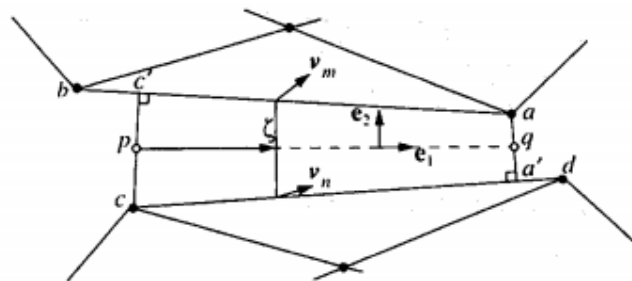


Figure 2.3.1 Fictitious layer schematic of grain boundary.
Produced by [41]

The determine process of grain boundary is as follows, as shown in Figure 2.3.1, first, node 'a' and 'c' are projected onto the adjacent grain surface to obtain their phantom node 'a'' and 'c'', and then connect the original nodes to its corresponding phantom nodes to obtain line 'a-a'' and 'c-c'', finally, use the 'p' and 'q' as the nodes of grain boundary, which 'p' and 'q' are the midpoints of line 'a-a'' and 'c-c''.

➤ Goodman Element

Goodman element is a non-thickness element that was first used to solve contact problems. In creep analysis, it is used to describe the mechanism of grain boundaries. In the 2D and 3D version, it consists of two rods and surfaces

which coincide with each other initially. This part will be explained in detail later [9,12].

➤ COH3D8

COH3D8 is from ABAQUS's element library. It is a 3D cohesive element with eight nodes and six faces, as shown in Figure 2.3.2.

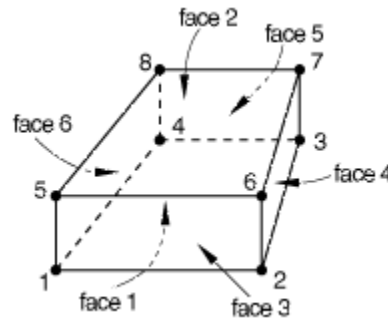


Figure 2.3.2 The schematic diagram of COH3D8.
Produced by [6]

The characteristic of this element is that the upper (face 2) and lower surface (face 1) have a thickness and follow the cohesive law [6,43].

For Grain Modeling

It can be divided into two categories:

1. Using a homogeneous and isotropic solid element with power law creep.

The grain is modeled by a homogeneous and isotropic material solid element with linear elasticity and power-law creep. According to these simulations, it can be concluded that this grain modeling method has the capability to capture the main creep damage feature that occur at the grain boundary part [9,41,42].

2. Using 12 sliding system solid element with visco-plastic crystal model

The grain is assumed to be the single FCC (Face Centered Cubic) crystal, and

its deformation includes two parts: linear elasticity and visco-plasticity. The deformation is along the sliding plane $\{1 \ 1 \ 1\}$ with direction $\langle 0 \ 1 \ 1 \rangle$, and kinematic hardening is ignored [44].

2.5 Cohesive Zone Model

Currently, CZM (Cohesive Zone Model) is widely used to describe the cohesive traction-separation law of cracking processes or analyze the localization and failure of materials [48-50]. With the development of numerical methods, Needleman [51] first used his self-defined CZM for FE analysis of crack propagation in ductile materials, and Petersson [52] and Carpinteri [53] used it in brittle fracture simulation. Compared with conventional fracture mechanics, CZM keeps the continuity of analysis in the separated state, and it avoids the stress's singularity during separation and limits the stress as the cohesive strength of the materials [54,55].

Although the grain boundary has a thickness, compared with the grain, it is only a few atom layers. Therefore, the general way to model grain boundary is to idealize it as a sharp interface between two adjacent grains. Based on this geometric characteristic, the CZM is used to describe the deformation of grain boundaries. The common traction-separation is described by constitutive equations, which relate to the creep phenomenon that occurs on grain boundaries (such as cavitation and sliding). Under the continuum mechanics framework, the deformation of grain boundaries is measured by using the displacement jump of two surfaces (upper and lower) [9,10].

2.6 Conventional Solid Element

Under different stress states, there are different element types. Generally, 3D problems can be idealized into three different stress states: plane stress, plane strain and axisymmetric. The difference between these three is the constitutive relationship between stress and strain.

2.6.1 Formulation of 2D Solid Element

The constitutive relationship between stress components and small strain components is shown below [56,88]:

Where E is Young's modulus, ν is Poisson's ratio. $[D]$ is the constitutive matrix.

1) Plane Stress

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = [D] \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad 2.6.1$$

$$[D] = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad 2.6.2$$

2) Plane Strain

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = [D] \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad 2.6.3$$

$$[D] = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \quad 2.6.4$$

3) Axisymmetric

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{rz} \end{Bmatrix} = [D] \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{rz} \end{Bmatrix} \quad 2.6.5$$

$$[D] = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \cdot \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 & \frac{\nu}{1-\nu} \\ \frac{\nu}{1-\nu} & 1 & 0 & \frac{\nu}{1-\nu} \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 1 \end{bmatrix} \quad 2.6.6$$

The relationship between displacement components $\{d\}$ and small strain components $\{\varepsilon\}$ is shown below:

where $\tilde{\mu}$ and \tilde{v} are the displacement component in two directions (x and y) [56,89].

1) Plane stress and Plane strain

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} \tilde{\mu} \\ \tilde{v} \end{Bmatrix} \quad 2.6.7$$

2) Axisymmetric

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{rz} \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial r} & 0 \\ 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & \frac{\partial}{\partial r} \\ \frac{1}{r} & 0 \end{bmatrix} \begin{Bmatrix} \tilde{\mu} \\ \tilde{v} \end{Bmatrix} \quad 2.6.8$$

The connection between node displacement and element displacement can be obtained by discretizing over element using shape functions.

where $\{\mu\}$ and $\{v\}$ store the node displacement, $[N]$ is the shape function

matrix [56].

$$\tilde{\mu} = [N]\{\mu\} \quad 2.6.9$$

$$\tilde{v} = [N]\{v\} \quad 2.6.10$$

$$\begin{Bmatrix} \tilde{\mu} \\ \tilde{v} \end{Bmatrix} = \begin{bmatrix} N & 0 \\ 0 & N \end{bmatrix} \begin{Bmatrix} \mu \\ v \end{Bmatrix} \quad 2.6.11$$

Therefore, the relationship between small element strain and node

displacement can be obtained by [56]:

1) Plane stress and Plane strain

Substitute equation 2.6.11 into equation 2.6.8, the strain-nodes displacement

matrix [B] can be obtained:

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = [B] \begin{Bmatrix} \mu \\ v \end{Bmatrix} \quad 2.6.12$$

$$[B] = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} N & 0 \\ 0 & N \end{bmatrix} \quad 2.6.13$$

2) Axisymmetric

Substitute equation 2.6.11 into equation 2.6.7, the strain-nodes displacement

matrix [B] can be obtained:

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{rz} \end{Bmatrix} = [B] \begin{Bmatrix} \mu \\ v \end{Bmatrix} \quad 2.6.14$$

$$[B] = \begin{bmatrix} \frac{\partial}{\partial r} & 0 \\ 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & \frac{\partial}{\partial r} \\ \frac{1}{r} & 0 \end{bmatrix} \begin{bmatrix} N & 0 \\ 0 & N \end{bmatrix} \quad 2.6.15$$

The element stiffness matrix can be obtained:

1) Plane stress and Plane strain

$$[K_m] = \iint [B]^T [D] [B] dx dy \quad 2.6.16$$

2) Axisymmetric

$$[K_m] = \iint [B]^T [D] [B] r dr dz d\theta \quad 2.6.17$$

In this FE program, applying Gauss-Legendre as numerical integration over element regions.

Finally,

The integration form of Equation 2.6.16 is

$$[K_m] = \sum_{i=1}^{nip} W_i \cdot \det|J|_i \cdot [B]^T [D] [B] \quad 2.6.18$$

The integration form of Equation 2.6.17 is

$$[K_m] = \sum_{i=1}^{nip} W_i \cdot \det|J|_i \cdot [B]^T [D] [B] r \quad 2.6.19$$

where W_i is weighting coefficient. $\det|J|_i$ is the Jacobian matrix.

2.6.2 Formulation of 3D Solid Element

The stress and strain constitutive relationship is shown in below [56,88]:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} = [D] \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} \quad 2.6.20$$

$$\begin{aligned}
& [D] \\
& = K \cdot \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & & & \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & & & \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & & & \\ & & & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ & & & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ & & & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \quad 2.6.21
\end{aligned}$$

$$\text{With} \quad K = \frac{E(1-\nu)}{(1+\nu) \cdot (1-2\nu)} \quad 2.6.22$$

Where E is Young's modulus, ν is Poisson's ratio. [D] is the constitutive matrix.

The relationship between displacement component {d} and small strain components { ϵ } is shown below [56,89]:

where $\tilde{\mu}, \tilde{\nu}, \tilde{\omega}$ and are the displacement component in three directions (x, y, and z).

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} \tilde{\mu} \\ \tilde{\nu} \\ \tilde{\omega} \end{Bmatrix} \quad 2.6.23$$

The connection between node displacement and element displacement can be obtained by discretizing over element using shape functions [N], and the connection between node displacement and the element displacement is:

$$\begin{Bmatrix} [\tilde{u}] \\ [\tilde{v}] \\ [\tilde{\omega}] \end{Bmatrix} = \begin{bmatrix} [N] & 0 & 0 \\ 0 & [N] & 0 \\ 0 & 0 & [N] \end{bmatrix} \begin{Bmatrix} [\mu] \\ [v] \\ [\omega] \end{Bmatrix} \quad 2.6.24$$

where $[\mu]$, $[v]$, and $[\omega]$ stores the nodes displacement vector in three directions, $[N]$ is the matrix which stores the shape function.

Substitute the equation 2.6.24 into 2.6.21, the strain-nodes displacement matrix $[B]$ can be obtained:

$$[B] = [A] \cdot [S] \quad 2.6.25$$

$$[A] = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \quad 2.6.26$$

$$[S] = \begin{bmatrix} N_1 & 0 & 0 & \dots & N_8 & 0 & 0 \\ 0 & N_1 & 0 & \dots & 0 & N_8 & 0 \\ 0 & 0 & N_1 & \dots & 0 & 0 & N_8 \end{bmatrix} \quad 2.6.27$$

Finally, the element stiffness matrix can be obtained by:

$$[K_m] = \iiint [B]^T \cdot [D] \cdot [B] dx dy dz \quad 2.6.28$$

2.7 2D Goodman Element

This element was first proposed for the FEA of 2D rock engineering numerical simulation. Due to its non-thickness geometric features, the continuity is guaranteed when analyzing contact problems. In this project, 4 nodes with 8 degrees of freedom Goodman element (2 degrees of freedom per node) is adopted. The topological direction of the node numbers is clockwise as shown

in Figure 2.7.1. Under unloading conditions, the upper and low lines coincide. When in a loaded state, the two lines are separated to generate a relative displacement in two directions, which is the normal direction 'N' (perpendicular to the surface) and the Separate direction 'S' (parallel to the surface) [12,46].

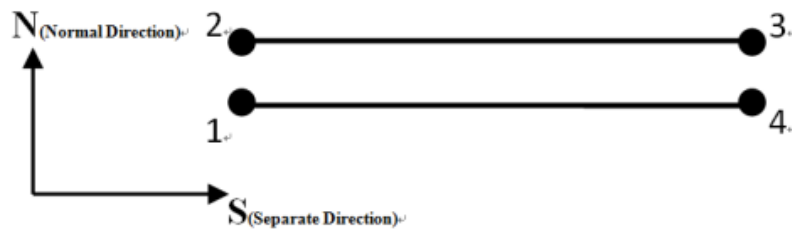


Figure 2.7.1 The schematic diagram of 2D Goodman element.
Produced from [12].

2.7.1 Mathematical Background

The element displacement vector, a^e , is defined by the eight displacement components of four nodes as [12,46]:

$$a^e = \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} \quad 2.7.1$$

with

$$a_1 = \begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix} \quad 2.7.2$$

for node 1, etc.

where u is normal direction and v is separate direction.

The displacement of the lower surface is:

$$\begin{bmatrix} u^l \\ v^l \end{bmatrix} = \frac{1}{2} \begin{bmatrix} N_1 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_4 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_4 \\ v_4 \end{bmatrix} \quad 2.7.3$$

The displacement of the upper surface is:

$$\begin{bmatrix} u^u \\ v^u \end{bmatrix} = \frac{1}{2} \begin{bmatrix} N_2 & 0 & N_3 & 0 \\ 0 & N_2 & 0 & N_3 \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} \quad 2.7.4$$

with shape functions of each node are defined as:

$$\begin{aligned} N_1 &= 1 - \frac{2x}{L} \\ N_2 &= 1 - \frac{2x}{L} \\ N_3 &= 1 + \frac{2x}{L} \\ N_4 &= 1 + \frac{2x}{L} \end{aligned} \quad 2.7.5$$

where x is the coordinate at x -axis in the local coordinate system, L is the length of Goodman element.

By using equations 2.7.3 to 2.7.4, the relative displacement $[\Phi]$ is:

$$[\Phi] = \begin{bmatrix} u^u - u^l \\ v^u - v^l \end{bmatrix} = \frac{1}{2} \cdot [B] \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{bmatrix} \quad 2.7.6$$

in which:

$$[B] = \begin{bmatrix} -A & 0 & A & 0 & B & 0 & -B & 0 \\ 0 & -A & 0 & A & 0 & B & 0 & -B \end{bmatrix} \quad 2.7.7$$

with

$$A = 1 - \frac{2x}{L}; B = 1 + \frac{2x}{L}, \quad 2.7.8$$

The relationship between stress and deformation can be implemented via cohesive law, which suits a small thickness or non-thickness at all. Therefore,

the [D] matrix which relates the two stress components to the relative displacement, thus

$$\begin{bmatrix} F_n \\ F_s \end{bmatrix} = [D] \cdot [\Phi], \quad 2.7.9$$

where

$$[F] = \begin{bmatrix} F_n \\ F_s \end{bmatrix} \quad 2.7.10$$

Here $[D] = \begin{bmatrix} k_n & 0 \\ 0 & k_s \end{bmatrix}$, k_n is the stiffness of normal direction and k_s is the stiffness of separated direction.

The potential-energy P in a linearly elastic Goodman element is shown in:

$$P = \frac{1}{2} L \Phi^T [K] \Phi = \frac{1}{2} \int_{-\frac{L}{4}}^{\frac{L}{2}} \frac{1}{4} [\Phi]^T [B]^T [D] [B] [\Phi] dx \quad 2.7.11$$

where [K] is the Goodman element stiffness per length.

Therefore, the [K] can be obtained:

$$[K] = \int_{-\frac{L}{4}}^{\frac{L}{2}} \frac{1}{4} [B]^T [D] [B] \cdot dx \quad 2.7.12$$

Substitution of the Equation 2.6.7 and 2.6.10

$$[B]^T [D] [B] = \begin{bmatrix} A^2 K_S & 0 & -A^2 K_S & 0 & -ABK_S & 0 & ABK_S & 0 \\ 0 & A^2 K_N & 0 & -A^2 K_N & 0 & -ABK_N & 0 & ABK_N \\ -A^2 K_S & 0 & A^2 K_S & 0 & ABK_S & 0 & -ABK_S & 0 \\ 0 & -A^2 K_N & 0 & A^2 K_N & 0 & ABK_N & 0 & -ABK_N \\ -ABK_S & 0 & ABK_S & 0 & B^2 K_S & 0 & -B^2 K_S & 0 \\ 0 & -ABK_N & 0 & ABK_N & 0 & B^2 K_N & 0 & -B^2 K_N \\ ABK_S & 0 & -ABK_S & 0 & -B^2 K_S & 0 & B^2 K_S & 0 \\ 0 & ABK_N & 0 & -ABK_N & 0 & -B^2 K_N & 0 & BK_N \end{bmatrix} \quad 2.7.13$$

wherein the integrals of these A^2 , B^2 , and AB for the length L are:

$$\int_{-\frac{L}{2}}^{\frac{L}{2}} A^2 = \frac{4}{3} L$$

$$\int_{-\frac{L}{2}}^{\frac{L}{2}} B^2 = \frac{4}{3} L \tag{2.7.14}$$

$$\int_{-\frac{L}{2}}^{\frac{L}{2}} AB = \frac{2}{3} L$$

Thence, the element stiffness matrix [K] is:

$$[K] = \frac{1}{6} \begin{bmatrix} 2K_s & 0 & -2K_s & 0 & -K_s & 0 & K_s & 0 \\ 0 & 2K_n & 0 & -2K_n & 0 & -K_n & 0 & K_n \\ -2K_s & 0 & 2K_s & 0 & K_s & 0 & -K_s & 0 \\ 0 & -2K_n & 0 & 2K_n & 0 & K_n & 0 & -K_n \\ -K_s & 0 & K_s & 0 & 2K_s & 0 & -2K_s & 0 \\ 0 & -1K_n & 0 & K_n & 0 & 2K_n & 0 & -2K_n \\ K_s & 0 & -K_s & 0 & -2K_s & 0 & 2K_s & 0 \\ 0 & -2K_n & 0 & -K_n & 0 & -2K_n & 0 & 2K_n \end{bmatrix} \tag{2.7.15}$$

2.7.2 Coordinate System Transmission

The stiffness matrix in the previous section is under a special condition when the two coordinates (Global and Local) are coincident. In general, these two coordinate systems need to be related by a transmission matrix [12,46].

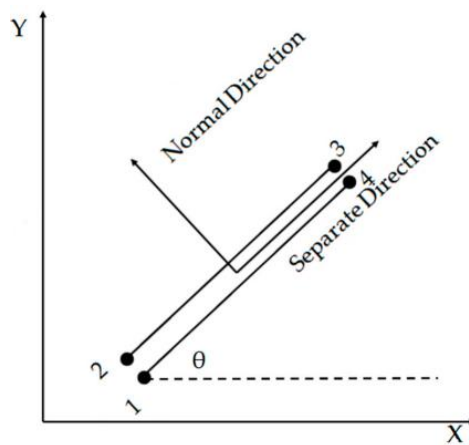


Figure 2.7.2 The schematic figure of Goodman element with angle.
Reproduced from [45]

As indicated in Figure 2.7.2, the angle between the local coordinates and the global coordinates is θ , the local element stiffness at global coordinates is [46]:

$$[K]_{Glo} = [T]^T [K]_{loc} [T] \quad 2.7.16$$

in which

$$T = \begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N & 0 \\ 0 & 0 & 0 & N \end{bmatrix}, N = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad 2.7.17$$

2.8 3D Goodman Element

In this project, 8 nodes with 24 degrees of freedom 3D Goodman element (3 degrees of freedom per node) is adopted. The topological direction of the node numbers is shown in Figure 2.8.1. Under unloading conditions, the upper and low surfaces coincide. When in a loaded state, the two surfaces are separated to generate a relative displacement in three directions, which is the normal direction ' ω ' (perpendicular to the surface) and two Separate directions ' v ' and μ (parallel to the surface) [47].

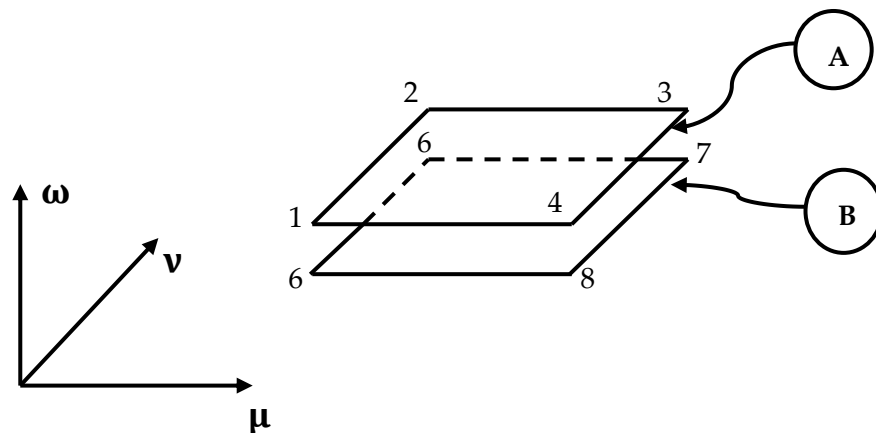


Figure 2.8.1 The schematic diagram of 3D Goodman element (Local coordinates system).

2.8.1 Mathematical Background

The element displacement vector, a^e are defined by 24 displacement components of 8 nodes as [47]:

$$a^e = \{a_1 \ a_2 \ a_3 \ a_4 \ a_6 \ a_6 \ a_7 \ a_8\}^T \quad 2.8.1$$

with

$$a_1 = \{\mu_1 \ v_1 \ \omega_1\}^T \quad 2.8.2$$

for node 1, etc.

The displacement vectors of the surface 'A' is:

$$\begin{bmatrix} \mu_A \\ v_A \\ \omega_A \end{bmatrix} = [F_1 \ F_2 \ F_3 \ F_4] \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} \quad 2.8.3$$

The displacement vectors of the surface 'B' is:

$$\begin{bmatrix} \mu_B \\ v_B \\ \omega_B \end{bmatrix} = [F_6 \ F_6 \ F_7 \ F_8] \begin{bmatrix} V_6 \\ V_6 \\ V_7 \\ V_8 \end{bmatrix} \quad 2.8.4$$

Where the matrix F stores the shape function and V stores the displacement vectors of each node, the shape function comes from quadrilateral element

[13].

$$[F_1] = \begin{bmatrix} N_1 & 0 & 0 \\ 0 & N_2 & 0 \\ 0 & 0 & N_3 \end{bmatrix} \quad 2.8.5$$

$$[V_1] = \begin{bmatrix} \mu_1 \\ v_1 \\ \omega_1 \end{bmatrix} \quad 2.8.6$$

for node i, etc.

with shape functions defined as:

$$N_1 = N_6 = \frac{1}{4}(1 - \xi)(1 - \eta) \quad 2.8.7$$

$$N_2 = N_6 = \frac{1}{4}(1 - \xi)(1 + \eta)$$

$$N_3 = N_7 = \frac{1}{4}(1 + \xi)(1 + \eta)$$

$$N_4 = N_8 = \frac{1}{4}(1 + \xi)(1 - \eta)$$

where ξ and η are the coordinates in a local system.

By applying equations 2.8.3 and 2.8.4, the relative displacement $[\Phi]$ can be

solved as:

$$[\Phi] = \begin{bmatrix} \mu_A - \mu_B \\ v_A - v_B \\ \omega_A - \omega_B \end{bmatrix} = [B] \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \\ V_8 \end{bmatrix} \quad 2.8.8$$

in which:

$$[B] = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ -F_5 \\ -F_6 \\ -F_7 \\ -F_8 \end{bmatrix}^T \quad 2.8.9$$

where matrix F has been mentioned in the Equation 2.8.5.

The mechanical relationship between stress and deformation can be

implemented via cohesive law and the matrix $[D]$ which relates the stress

components to relative displacement, thus

$$\begin{bmatrix} F_\mu \\ F_v \\ F_\omega \end{bmatrix} = \begin{bmatrix} K_\mu & 0 & 0 \\ 0 & K_v & 0 \\ 0 & 0 & K_\omega \end{bmatrix} \begin{bmatrix} \Phi_\mu \\ \Phi_v \\ \Phi_\omega \end{bmatrix} \quad 2.8.10$$

Due to the thickness of this element is zero, the integration of the stiffness

matrix is only on the two surfaces A and B. The 3D Goodman element stiffness matrix can be obtained (in local coordinate):

$$[K_M] = \frac{1}{2} \int_{-1}^{+1} \int_{-1}^{+1} B^T \cdot D \cdot B |J| d\xi d\eta \quad 2.8.11$$

Here, the implementation of numerical integration over the surface A and B regions is by the Gauss-Legendre method.

The integration form of Equation 2.8.11 is

$$[K_M] = \frac{1}{2} \sum_{i=1}^{nip} W_i \cdot \det|J|_i \cdot [B]^T [D] [B] \quad 2.8.12$$

where W_i is the weighting coefficient. $\det|J|_i$ is the determinant of the Jacobin matrix.

The mathematical background of solving $|J|_i$ is shown below: Based on matrix [B], the derivatives 'der' of the shape functions in surface A and B as (local coordinates system):

$$\text{der} = \begin{bmatrix} \frac{\partial \text{Fun}^T}{\partial \mu} \\ \frac{\partial \text{Fun}^T}{\partial \nu} \end{bmatrix} = \begin{bmatrix} -A & -B & B & A & -A & -B & B & A \\ -C & -D & D & C & -C & -D & D & C \end{bmatrix} \quad 2.8.13$$

where,

$$\begin{aligned} A &= (1-\mu) & C &= (1-\nu) \\ B &= (1+\mu) & D &= (1+\nu) \end{aligned} \quad 2.8.14$$

Therefore, the Jacobin matrix [J] can be obtained:

$$[J] = [\text{der}] \cdot [\text{coord}]_{\mu \& \nu} \quad 2.8.15$$

where:

$$[\text{coord}]_{\mu\&v} = \begin{bmatrix} \mu_1 & v_1 \\ \mu_2 & v_2 \\ \mu_3 & v_3 \\ \mu_4 & v_4 \\ \mu_6 & v_6 \\ \mu_6 & v_6 \\ \mu_7 & v_7 \\ \mu_8 & v_8 \end{bmatrix} \quad 2.8.16$$

$[\text{coord}]_{\mu\&v}$ stores the coordinates in two directions: μ and v .

Finally, the Jacobin matrix $[J]$ can be solved:

$$[J] = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \quad 2.8.17$$

with

$$\begin{aligned} J_1 &= -A \cdot \mu_1 - B \cdot \mu_2 + B \cdot \mu_3 + A \cdot \mu_4 - A \cdot \mu_6 - B \cdot \mu_6 + B \\ &\quad \cdot \mu_7 + A \cdot \mu_8 \\ J_2 &= -A \cdot v_1 - B \cdot v_2 + B \cdot v_3 + A \cdot v_4 - A \cdot v_6 - B \cdot v_6 + B \\ &\quad \cdot v_7 + A \cdot v_8 \\ J_3 &= -C \cdot \mu_1 - D \cdot \mu_2 + D \cdot \mu_3 + C \cdot \mu_4 - C \cdot \mu_6 - D \cdot \mu_6 + D \\ &\quad \cdot \mu_7 + C \cdot \mu_8 \\ J_4 &= -C \cdot v_1 - D \cdot v_2 + D \cdot v_3 + C \cdot v_4 - C \cdot v_6 - D \cdot v_6 + D \\ &\quad \cdot v_7 + C \cdot v_8 \end{aligned} \quad 2.8.18$$

where A, B, C, and D are shown in Equation 2.8.18.

2.8.2 Coordinate System Transmission

The stiffness matrix in the previous section is under a special condition when the two coordinates (Global and Local) are coincident. In general, these two coordinate systems need to be related by a transmission matrix.

The relationship between Local coordinate system and Global coordinate system is [47]:

$$\begin{Bmatrix} x' \\ y' \\ z' \end{Bmatrix} = \begin{bmatrix} \cos(x', x) & \cos(x', y) & \cos(x', z) \\ \cos(y', x) & \cos(y', y) & \cos(y', z) \\ \cos(z', x) & \cos(z', y) & \cos(z', z) \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} \quad 2.8.19$$

In Equation 2.8.19, $\cos(x', x)$ is the cosine of the angle between the x' axis of the local coordinate system and the X axis of the global coordinate system,

and so on and so forth. The element stiffness matrix in the local coordinate system has been solved in the previous section 2.8.1, and transmit it to the global coordinate system is:

$$[KM]_{\text{Global}} = T^T \cdot [KM]_{\text{Local}} \cdot T \quad 2.8.20$$

in which

$$T = \begin{bmatrix} Q & 0 & \dots & 0 \\ 0 & Q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q \end{bmatrix} \quad 2.8.21$$

with $Q = \begin{bmatrix} \cos(x', x) & \cos(x', y) & \cos(x', z) \\ \cos(y', x) & \cos(y', y) & \cos(y', z) \\ \cos(z', x) & \cos(z', y) & \cos(z', z) \end{bmatrix}$

2.9 Creep Constitutive Equation

This section reviews four sets of creep constitutive equations, which have been built into the procedure framework. The first three describe the creep evolution at the macro-scale, and the last two describe the cavitation and sliding evolution of the grain boundaries.

2.9.1 Macro-Creep Constitutive Equations

1. Kachanov-Rabatnov (KR)

It is a creep constitutive equation with Power Law Stress Sensitivity, shown in Equation 2.9.1. The description of the creep degradation behavior depends on a single damage state variable [6].

$$\dot{\varepsilon}_{ij} = \frac{3}{2} \cdot A \cdot \left(\frac{\sigma_{eq}}{1 - \omega} \right)^n \cdot \frac{S_{ij}}{\sigma_{eq}} \cdot t^m \quad 2.9.1(a)$$

$$\dot{\omega} = B \cdot \frac{\sigma_r^\chi}{(1 - \omega)^\phi} \cdot t^m \quad 2.9.1(b)$$

$$\sigma_r = \alpha \cdot \sigma_1 + (1 - \alpha) \cdot \sigma_{eq} \quad 2.9.1(c)$$

where A, B, n, m, ϕ , and α are the material constants. σ_{eq} is the equivalent stress, σ_1 is the maximum stress, S_{ij} is the deviator stress tensor, σ_r is the rupture stress, ω is the damage variable and ε is the creep strain tensor.

2. Kachanov-Rabatnov-Hayhurst (KRH)

This is a creep constitutive equation with Hyperbolic Sine Law Stress Sensitivity, shown in Equation 2.9.2. Compared with the previous KR form, the description of the creep degradation behavior depends on three state variables, which is used to depict the cavitation and the coarsening of the carbide precipitates [57].

$$\dot{\varepsilon}_{ij} = \frac{3}{2} \cdot \frac{S_{ij}}{\sigma_{eq}} \cdot A \cdot \sinh \left[\frac{B \cdot \sigma_{eq} \cdot (1 - H)}{(1 - \Phi) \cdot (1 - \omega)} \right] \quad 2.9.2(a)$$

$$\dot{H} = \frac{h \cdot \dot{\varepsilon}_e}{\sigma_{eq}} \cdot \left(1 - \frac{H}{H^*} \right) \quad 2.9.2(b)$$

$$\dot{\Phi} = \frac{K_C}{3} \cdot (1 - \Phi)^4 \quad 2.9.2(c)$$

$$\dot{\omega} = D \cdot N \cdot \dot{\varepsilon}_e \cdot \left\langle \frac{\sigma_1}{\sigma_{eq}} \right\rangle^v \quad 2.9.2(d)$$

Where $N=1$ when $\sigma_1 > 0$ and $N=0$ when $\sigma_1 < 0$. A, B, C, h, H^* , and K_C are the material constants. H, Φ , and ω are the three state variables, H ($H_{t=0} = 0$) presents the strain hardening, Φ ($\Phi_{t=0} = 0$) presents the evolution of the density of the carbide precipitates, ω ($0 \leq \omega \leq \frac{1}{3}$) presents the

cavitation damage [58].

3. Kachanov-Rabatnov-Hayhurst-Qing (KRHQ)

The KRHQ is developed from the previous KRH, shown in Equation 2.9.3, the main difference is the relationship between stress state and cavitation damage rate. Two additional functions f_1 and f_2 are added to determine the damage rate, where f_1 is used to describe the phenomenological relationship between damage, tertiary creep deformation, and creep rupture. f_2 is used to describe the coupling between creep damage evolution and the stress state effect [59].

$$\dot{\epsilon} = \frac{3}{2} \cdot \frac{S_{ij}}{\sigma_{eq}} \cdot A \cdot \sinh \left[\frac{B \cdot \sigma_{eq} \cdot (1 - H)}{(1 - \Phi) \cdot (1 - \omega)} \right] \quad 2.9.3(a)$$

$$\dot{H} = \frac{h \cdot \dot{\epsilon}_e}{\sigma_{eq}} \cdot \left(1 - \frac{H}{H^*} \right) \quad 2.9.3(b)$$

$$\dot{\Phi} = \frac{K_C}{3} \cdot (1 - \Phi)^4 \quad 2.9.3(c)$$

$$\dot{\omega} = D \cdot N \cdot \dot{\epsilon}_e \cdot f_1 \cdot f_2 \quad 2.9.3(d)$$

$$f_1 = \left(\frac{2\sigma_e}{3S_1} \right)^a \cdot \exp \left\{ b \cdot \left[\frac{3\sigma_m}{S_s} - 1 \right] \right\} \quad 2.9.3(e)$$

$$f_2 = \left(\exp \left\{ p \cdot \left[1 - \frac{\sigma_1}{\sigma_e} \right] + q \cdot \left[\frac{1}{2} - \frac{3\sigma_m}{2\sigma_e} \right] \right\} \right)^{-1} \quad 2.9.3(f)$$

where $S_s = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}$, $\sigma_m = \frac{1}{3}(\sigma_1 + \sigma_2 + \sigma_3)$, $S_1 = \sigma_1 - \sigma_m$ and σ_1 , σ_2 , and σ_3 are the principal stresses, a , b , p , and q are the material parameters.

2.9.2 Micro-Creep Constitutive Equations

For most high-temperature alloys, the main reasons of creep damage, particularly creep rupture, is due to the cavitation happening on grain boundary, most creep modeling at grain boundary level is also based on the

evolution of cavitation over time [60-66]. In this work, the deformation of grain boundary under creep conditions is represented by displacement jump, which assumed cavitation for the normal direction and sliding for the tangential directions [9,10].

The cavitation contains two aspects, cavity nucleation and cavity growth. For cavity nucleation, the mechanisms are still not fully understood, generally, according to the observation of specimens, cavities are usually found on the grain boundaries, and in the plane perpendicular to the tension's direction, the density of the cavities is higher than other planes. For the cavity growth, it suggested three mechanisms, the plastic deformation-dominated, sliding control, and constraint-dominated [67]. The cavitation models based smeared-out cavity model has been developed for the analysis of Copper-Antimony alloys under creep conditions [10,68], which accounts for cavity nucleation, cavity annihilation, and cavity growth. In this model, the failure of a single grain boundary is determined by the ratio of cavity area, and its critical value is 0.5 [10].

At normal direction, the displacement jump is determined by Vose's cavitation models, as shown in Equation 2.9.4.

$$\frac{d\bar{\beta}}{d\bar{t}} = \frac{3\beta}{2\bar{\rho}} (\bar{\alpha}_p - \bar{\alpha}_a) + \sqrt{\bar{\rho}^3} \sqrt{36h(\psi)\pi\beta^2} \frac{d\bar{a}}{d\bar{t}} \quad 2.9.4(a)$$

$$\frac{d\bar{\rho}}{d\bar{t}} = \bar{\alpha}_p(1-f) - \bar{\alpha}_a \quad 2.9.4(b)$$

$$\bar{\alpha}_a = x_3 \cdot 8\pi\bar{\rho}^2\bar{a} \frac{d\bar{a}}{d\bar{t}} \quad 2.9.4(c)$$

$$\omega = \sqrt[3]{\frac{9\pi\beta^2}{16h^2(\psi)}}; \quad 2.9.4(d)$$

$$\bar{a} = \frac{1}{\sqrt{\bar{\rho}}} \sqrt[3]{\frac{3\beta}{4h(\psi)\pi}} \quad 2.9.4(e)$$

$$f = \frac{(\eta - 1)\omega}{1 - \omega} \quad 2.9.4(f)$$

$$\eta = \exp\left(\left[x_4 \cdot 2\pi\bar{D}_{gb}(\bar{a}_{tip}(\bar{a} = 1) - \bar{a}_{tip}(\bar{a}))\bar{\rho} \left(\frac{d\bar{\mu}^p}{d\bar{t}}\right)^{-1}\right]\right) \quad 2.9.4(g)$$

$$\frac{d\bar{\mu}^p}{d\bar{t}} = \frac{\beta}{\sqrt{\bar{\rho}^3}} (\bar{\alpha}_p - \bar{\alpha}_a) + \sqrt[3]{36h(\psi)\pi\beta^2} \frac{d\bar{a}}{d\bar{t}} \quad 2.9.4(h)$$

$$q(\omega) = -2\ln\omega - (3 - \omega)(1 - \omega); \quad 2.9.4(i)$$

$$\bar{a}_{tip}(\bar{a}) = 2\bar{\gamma}_s \sin \psi / \bar{a},$$

where ρ and a are the density and the average radius of cavities, β is the damage variable, $\bar{\alpha}_p$ is the nucleation rate, $\bar{\alpha}_a$ is the annihilation rate, $\psi = 70^\circ$ (the dihedral angle), \bar{D}_{gb} is the diffusion coefficient, ω is the damaged area fraction. The deformation of the grain boundary part is quantified by the relative jump displacement and it determined by two variables together, ρ and β , and the relationship is shown in Equation 2.9.5.

$$D_n = \frac{\beta}{\sqrt{\rho}} - \frac{\beta_0}{\sqrt{\rho_0}} \quad 2.9.5$$

where the D_n is the normal jump displacement, $\beta_0(10^{-4})$ is the initial damage value, $\rho_0(10^{-3}\text{mm}^{-2})$ is the initial cavity density.

At tangential directions, the displacement jump is assumed to be determined by Newtonian Viscous Flow [14], shown in Equation 2.9.6.

$$\frac{du_{\text{sliding}}}{dt} = \frac{\sigma_{\text{sliding}}}{\eta_{\text{sliding}}} \quad 2.9.6$$

where $\frac{du_{\text{sliding}}}{dt}$ is the relative sliding velocity, σ_{sliding} and η_{sliding} are the separate stress and the sliding viscosity of grain boundary, respectively.

2.10 Removal Function

During the creep evolution, the damage value is the index of element degradation, which increases from zero in the initial state, to the critical value at the end. When damage value reaches the value, the element is treated as a failure and is unable to sustain any load. These failed elements need to be removed from the structure and the boundary value problem needs to be resolved. Specifically, the failed element needs to be identified and located, and then its element stiffness matrix needs to be removed from the global stiffness matrix [69,70]. Overall, the two keys to implementing the removal function are examination and removal.

A common approach is to capture the damage value of each element and compare it to the critical value. When the damage value is greater than or equal to this value, the failed element is marked and its element number is recorded. Since the assembly and solution of the global stiffness matrix is expensive in computer time (CPU-time), the trigger function is considered necessary [70]. An element self-examination module is performed prior to

each iteration step, when a new failed element is generated, a loop for locating this element will be triggered. In this loop, the stiffness matrixes of failed elements are removed, and the global stiffness matrix is re-assembled and re-solved. When the loop is not triggered, skip this block and jump to the next iteration step. This function avoids unnecessary calculations and improves efficiency [69,70].

The direct way to remove the failed element is to set its element stiffness matrix to zero, however, when an island effect is formed (the normal element is surrounded by failed elements and its connection to the structure is broken) or the failure occurs at the boundary, it will cause the singularity of the stiffness matrix solution, it makes the process complex. Another way to is to reduce the stiffness matrix of the failed element, Hyde adopted this method in his Notched bar case study and proved its usability [71]. Comparing the two methods, the latter is simpler, although the failed elements are not removed from the physical structure completely. In regard to this project and the project's progress, it is reasonable to choose the latter one.

2.11 Restart Facility

MT.Wong added this technique for the 'DAMAGE XXX' to allow the software to be stopped and restarted freely, however, it was not clearly documented in detail on how to implement this function [69]. In practice, since the creep simulation spends hundreds of hours or more, the ability to output the result in stages and restart calculations from breakpoints is particularly important. It

reduces the possibility of a loss of the result data if the computer was to accidentally shutdown. The key to developing this function is 'READ-IN' and 'WRITE-OUT'. The 'READ-IN' module is the required data needed to be imported into the main program accurately, such as the element status (good or failed), state variable (such as creep strain, damage value, etc.), time, iteration step number, self-equilibrating global node force etc. The 'WRITE-OUT' module needs to output the required data and the intermediate result in a standard format (the format required by 'READ-IN' module) at the specified iteration step or time point.

Chapter 3 Methodology

3.1 Introduction

In this chapter, it reports the general methodology considered in this project.

It includes the general methodology of software development, specific mathematical methods or tools, mature technologies, and legacy codes.

A brief overview of each section is presented below.

- 1) In section 3.2, it presents the general methodology of software development in this project.
- 2) In section 3.3, the numerical integration methods used in this project are introduced, including Euler, 4th order Runge-Kutta, and Gauss-Legendre
- 3) In section 3.4, the mature techniques used to implement matrix storages and solution used in this project are introduced, and the legacy codes used to implement these techniques are also presented.
- 4) In section 3.5, it reports the displacement non-linear iteration method for solving the creep boundary value problem.
- 5) In section 3.6, it reports the mathematical background used to implement Goodman element coordinate system transmission.

3.2 General Methods for Developing In-house Procedure

Generally, the development includes four stages: Planning, Developing, Benchmark, and Maintenance [11].

1. Planning

This procedure is based on CDM (Continuum Damage Mechanism) to implement Finite Element Analysis. In order to avoid duplication and improve efficiency, OOP (Object-Oriented Programming) is adopted as a programming paradigm and some existing technologies, subroutine libraries, program structures, and interface standards are used [13]. In addition, Goodman element with a cohesive zone model is used to model grain boundaries. The programming environment is Fortran 2003 within the Visual Studio 2013 platform (Version 11.0.6129.00, Microsoft, Remond, WA USA).

2. Developing

The developing logic is from linearity to non-linearity, from 2D to 3D, and from macro-scale to micro-scale. During the development process, the linear version adopts the existing framework and legacy code to implement the following four techniques [13]:

- 1) size allocation of dynamic arrays.
- 2) evolution of the mesh information to allocate the matrix size of the global stiffness matrix size.
- 3) the assembly and solution of the global stiffness matrix.
- 4) retrieve the elastic stress field in the structure.

The main program is refactored from P61 [13], and the displacement iteration method adopts to implement residual stress updating [3]. The specific amends are summarized as follows:

- 1) add removal function.
- 2) add automatic time step function.
- 3) add the restart function.
- 4) upgrade from 'Single-Material& Singe-Element Type' to 'Multi-Material& Multi-Element Type' version.

3. Benchmark

In order to make the benchmark logical and efficient, the benchmark process is designed from simple to complex, from the linearity to nonlinearity.

Specifically, the validation of the macro solver contains two stages, the simple regular structure is chosen as the initial step to demonstrate the accuracy and stability under the uniform stress distribution and the results can be benchmarked with the theoretical results. The notched bar structure is chosen as a further step to verify the procedure under the non-uniform stress distribution and the result can be benchmarked with Hyde's Notched bar case study [71]. Similarly, the simple bi-crystal structure is chosen as the first step to verify the accuracy and stability.

4. Maintenance

Complete documentation and clear code comments benefit to improve the readability of the program, it makes the job easier for subsequent developers to maintain the procedure and update the version more efficiently [72].

3.3 Integration Method

In this project, the numerical integration adopts the Euler method and 4th order Runge-Kutta Integration Method, and the region integration uses the Gauss-Legendre method.

3.3.1 Numerical Integration

The creep constitutive equation is a type of partial differential equation, and its solution is to integrate it with time. The commonly used numerical engineering integration methods are Forward Euler and 4th order Runge-Kutta integration method.

1. Euler Integration Method

At i th step, the increase rate of variable y is $R_i(y_i)$, and the result of the variable y at $i+1$ step is

$$y_{i+1} = y_i + R_i(y_i) \cdot \Delta t \quad 3.3.1$$

Where Δt is time step.

2. 4th order Runge-Kutta Integration Method

At i step, the increase rate of variable y is determined by K_1 , K_2 , K_3 , and K_4

$$\begin{aligned}
K_1 &= R_i(y_i) \\
K_2 &= R_i\left(y_i + K_1 \cdot \frac{1}{2} \Delta t\right) \\
K_3 &= R_i\left(y_i + K_2 \cdot \frac{1}{2} \Delta t\right) \\
K_4 &= R_i(y_i + K_3 \cdot \Delta t)
\end{aligned}
\tag{3.3.2}$$

The final increase rate R_i is determined by

$$R_i = \frac{1}{6} (K_1 + 2K_2 + 2K_3 + K_4) \tag{3.3.3}$$

The result of variable y at $i+1$ step is

$$y_{i+1} = y_i + R_i(y_i) \cdot \Delta t \tag{3.3.4}$$

Where Δt is time step.

Compared with Euler, Runge-Kutta has its advantages for creep damage analysis. Although it increases the computational effort, it can accept larger time-step while ensuring accuracy and stability, and this advantage is even more obvious in large-scale case studies [74,75].

3.3.2 Numerical Integration for Element

In most FE programs, the analytical numerical integration over the element regions adopts Gauss-Legendre [73]. The quadrature rules are all with the form [13],

$$\begin{aligned}
\int_{-1}^1 \int_{-1}^1 f(\xi, \eta) d\xi d\eta &\approx \sum_{i=1}^n \sum_{j=1}^n \omega_i \omega_j f(\xi_i, \eta_j) \\
&\approx \sum_{i=1}^{nip} W_i f(\xi, \eta)_i
\end{aligned}
\tag{3.3.5}$$

where nip represents the quantity of Gauss integrating points, W_i weighting coefficients ($W_i = \omega_i \cdot \omega_j$), (ξ, η) are the local coordinates of the sampling

point (Gauss integrating point).

The typical values of the weights and coordinates in Equation 3.3.5 are shown in Table 3.3.1.

Table 3.3.1 Typical weights and coordinates in Gauss-Legendre integration formula. Reproduced from [13]

n	nip	(ξ_i, η_j)	$\omega_i \cdot \omega_j$	W_i
1	1	(0,0)	(0,0)	4
2	4	$\left(\pm \sqrt{\frac{1}{3}}, \pm \sqrt{\frac{1}{3}}\right)$	(1,1)	1
3	9	$\left(\pm \sqrt{\frac{3}{5}}, \pm \sqrt{\frac{3}{5}}\right)$	$\left(\frac{5}{9}, \frac{5}{9}\right)$	$\frac{25}{81}$
		$\left(\pm \sqrt{\frac{3}{5}}, 0\right)$	$\left(\frac{5}{9}, \frac{8}{9}\right)$	$\frac{40}{81}$
		$\left(0, \pm \sqrt{\frac{3}{5}}\right)$	$\left(\frac{8}{9}, \frac{5}{9}\right)$	$\frac{40}{81}$
		(0,0)	$\left(\frac{8}{9}, \frac{8}{9}\right)$	$\frac{64}{81}$

In this project, this method is used to do the numerical integration over regions for conventional solid element and 3D Goodman element to obtain the element stiffness matrix.

3.4 Mature Techniques and Legacy Code

This section introduces mature techniques (matrix storage and solution) and the legacy code used to implement the above techniques.

3.4.1 Storage of Matrix

The stiffness matrix is a kind of symmetric, sparse, and positive-definite

matrix. There are two common methods of storing such matrices in scientific computing: banded matrix storage and skyline matrix storage. Comparing the two methods, the former stores all entries in the half bandwidth, and the latter only stores the first nonzero entry to the last non-zero entry in each column [76,77].

In Finite Element programs, the skyline storage method has been widely used to store stiffness matrices. Because the skyline of the stiffness matrix is relatively small and the Cholesky decomposition (commonly method to solve the stiffness matrix in Finite Element Analysis) preserves the skyline [76-78].

3.4.2 Cholesky Decomposition

The code of Finite Element boundary problem is to solve the global equilibrium equation,

$$[K_M]\{U\} = \{F\} \quad 3.4.1$$

where K_M is the global stiffness matrix, $\{F\}$ is the external node-loads, and $\{U\}$ is the global node displacement array which is the solution objective.

The method general adopts in Finite Element Method is to perform

$[K_M]$ Cholesky decomposition [78] to

$$[K_M] = LL^* \quad 3.4.2$$

in which L is a lower triangular matrix and L^* is the conjugate transpose matrix of L .

Substitution of the 3.4.2 into 3.4.1

$$LY = \{F\} \quad 3.4.3$$

where $Y = L^*\{U\} \quad 3.4.4$

Then, based on forward substitution to solve Y in Eq.3.4.3, and solving for {U} in Eq.3.4.6 by back substitution.

$$L^*\{U\} = Y \quad 3.4.5$$

Finally, stress and strain at each Gauss-sampling-points can be retrieved from the {U}.

3.4.3 Legacy Code

The above mature techniques have been developed and packaged as open-source subroutines; therefore, they are used in this project to avoid duplication. Details of these legacy subroutines are summarized below.

1. 'fsparv' for Skyline storage

In this module, it is integrated into subroutine 'fsparv'. The description of this subroutine is: it performs the symmetric skyline storage method of the global stiffness matrix, and it returns lower triangle mass of the global stiffness matrix as a vector, and output an array 'kdiag' which contains the location information of the diagonal element of the global stiffness matrix [13].

2. Subroutines for Cholesky Decomposition

In this module, two subroutines have been developed to implement this function [13].

1) 'sparin'

It performs a Choleski factorization of the Global stiffness matrix $[K_M]$ stored

as skyline. The mathematical background is Eq.3.4.2 and the stored method is has mentioned before.

2) 'spabac'

It returns the displacement array $\{U\}$ by forward-substitution and back-substitution on Choleski factorised vector $[K_M]$ by subroutines 'sparin'. The mathematical background is Eq.3.4.4 and Eq.3.4.5.

3.5 Creep Non-linear Iteration

The general solution of the creep boundary value problem relies on displacement iteration to update the residual stress caused by creep deformation [79]. This section provides a mathematical background for creep residual stress updating, which used in the fourth part of section 4.2.1.

Within each iteration step, the global node displacement $\{U\}$ by Eq.3.4.1

For each element, the total strain contains two parts: elastic and creep

$$\varepsilon_{tol} = \varepsilon_e + \varepsilon_c \quad 3.5.1$$

where ε_{tol} , ε_e , and ε_c are total, elastic, and creep strain respectively.

The effective stress depends on the elastic stress, which can be obtained by

$$\sigma_e = [D] \cdot \varepsilon_e = [D] \cdot (\varepsilon_{tol} - \varepsilon_c) \quad 3.5.2$$

Where σ_e is effective elastic stress, $[D]$ is the stress-strain matrix.

This stress causes creep deformation and generates creep body loads in the structure.

3.6 Mathematical Background of Coordinate Transmission

The background of the coordinate transmission system and transmission matrix of Goodman element has been mentioned in the previous section 2.7 (2D version) and section 2.8 (3D version). The core to implement this module is to obtain the angle θ to calculate the analytical solution of this transmission matrix. In this section, the mathematical background is described in detail.

3.6.1 2D Coordinate Transmission

The angle θ between the global coordinate system and the local coordinate system can be obtained by a geometric calculation, as indicated in Figure 2.7.2. Using the upper line as the sample,

$$\begin{aligned}\sin \theta &= \frac{(y_3 - y_2)}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}} \\ \cos \theta &= \frac{(x_3 - x_2)}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}}\end{aligned}\tag{3.6.1}$$

Where coordinate of node No.2 is (x_2, y_2) , and No.3 is (x_3, y_3) .

3.6.2 3D Coordinate Transmission

In order to obtain the angle θ more convenient, another system M (X', Y', Z') is added. In this system, the origin point coincides with the origin of the global system, and the Z' axis is along the normal direction of the surface, the X' axis and Y' axis are parallel to the surface, as indicated in Figure 3.6.2.

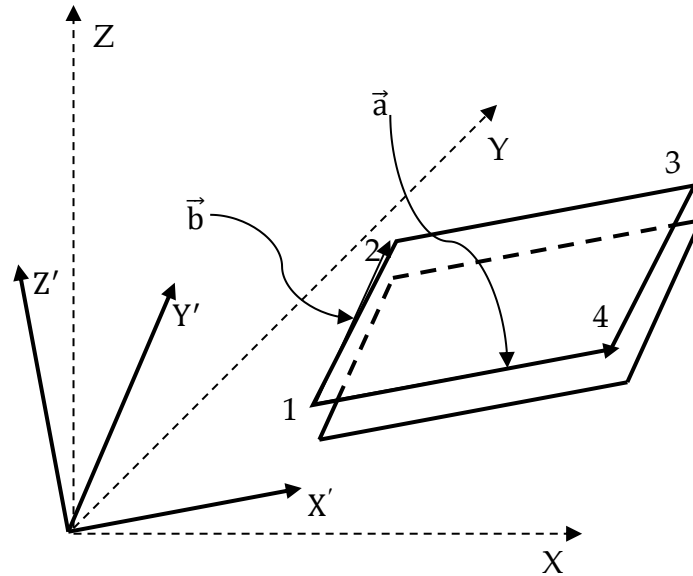


Figure 3.6.2 The schematic figure of new add coordinate system

In this module, the mathematical tool of vector cross product is adopted. The specific process of obtaining this coordinate system is as follows. First, the vector \vec{a} and vector \vec{b} can be constructed from the element global node coordinates, which equal $\overrightarrow{1,4}$ (Node No.4 to No.1) and $\overrightarrow{1,2}$ (Node No.2 to No.1 node), respectively.

$$\begin{aligned}\vec{a} &= \overrightarrow{1,4} = ((x_4 - x_1) \quad (y_4 - y_1) \quad (z_4 - z_1)) \\ \vec{b} &= \overrightarrow{1,2} = ((x_2 - x_1) \quad (y_2 - y_1) \quad (z_2 - z_1))\end{aligned}\tag{3.6.2}$$

where the node coordinates of No.1 is (x_1, y_1, z_1) , No.2 is (x_2, y_2, z_2) , and No.4 is (x_4, y_4, z_4) .

The vector Z' can be obtained by the cross product of \vec{a} and \vec{b} , which is perpendicular to the element surface containing these two vectors, follow the right-hand law.

$$\vec{Z}' = \vec{a} \times \vec{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ A & B & C \\ D & E & F \end{vmatrix} \quad 3.6.3$$

where $A=x_4 - x_1, B=y_4 - y_1, C=z_4 - z_1$

$$D=x_2 - x_1, E=y_2 - y_1, F=z_2 - z_1$$

Thus,

$$\vec{Z}' = ((B \cdot F - C \cdot E) \quad (C \cdot D - A \cdot F) \quad (A \cdot E - B \cdot D)) \quad 3.6.4$$

The vector \vec{Y}' axis equals the cross product of \vec{X}' and \vec{Z}' , which can be found in Equation 7.3.19 and Equation 7.3.21, respectively.

$$\vec{Y}' = \vec{Z}' \times \vec{X}' = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ Z_1 & Z_2 & Z_3 \\ A & B & C \end{vmatrix} \quad 3.6.5$$

where $Z_1=B \cdot F - C \cdot E, Z_2=C \cdot D - A \cdot F, Z_3=A \cdot E - B \cdot D$

Thus,

$$\vec{Y}' = (Y_1 \quad Y_2 \quad Y_3) \quad 3.6.6$$

where $Y_1 = C^2 \cdot D - A \cdot C \cdot F - A \cdot B \cdot E + B^2 \cdot D$

$$Y_2 = A^2 \cdot E - A \cdot B \cdot D - B \cdot C \cdot F + C^2 \cdot E$$

$$Y_3 = B^2 \cdot F - B \cdot C \cdot F - A \cdot C \cdot D + A^2 \cdot F$$

The elements in the transmission matrix (In Equation 2.8.19) can be obtained as following,

$$\cos(x', x) = \frac{A}{\sqrt{A^2 + B^2 + C^2}}$$

$$\cos(x', y) = \frac{B}{\sqrt{A^2 + B^2 + C^2}}$$

$$\cos(x', z) = \frac{C}{\sqrt{A^2 + B^2 + C^2}}$$

$$\cos(y', x) = \frac{Y_1}{\sqrt{Y_1^2 + Y_2^2 + Y_3^2}}$$

$$\cos(y', y) = \frac{Y_2}{\sqrt{Y_1^2 + Y_2^2 + Y_3^2}}$$

$$\cos(y', z) = \frac{Y_3}{\sqrt{Y_1^2 + Y_2^2 + Y_3^2}}$$

$$\cos(z', x) = \frac{Z_1}{\sqrt{Z_1^2 + Z_2^2 + Z_3^2}}$$

$$\cos(z', y) = \frac{Z_2}{\sqrt{Z_1^2 + Z_2^2 + Z_3^2}}$$

$$\cos(z', z) = \frac{Z_3}{\sqrt{Z_1^2 + Z_2^2 + Z_3^2}}$$

3.6.7

Chapter 4 Development of the Creep Solver

4.1 Introduction

In this chapter, it reports the development details of the in-house FE procedure at multi-scales. At macro scale version, the conventional triangle or quadrilateral solid element is adopted to solve different stress conditions, which include plane stress, plane strain, and axisymmetric. At grain boundary level, the grain and the grain boundary are modeled independently, and the simulation considers the mechanism of grain boundaries, which include the cavity evolution and sliding of grain boundary respectively. In this project, the grain boundary part is modeled by Goodman element to ensure the continuity, thus it allows the simulation to be implemented within the traditional FEM framework.

The developed platform of this procedure is the Fortran 2013 with the Visual Studio 2013 and the main structure of this procedure modified from the program P61, which is expanded from homogeneous into non-homogeneous versions. The brief introduction of each section is described in the below.

1. Section 4.2 introduces how to reformat the nonlinear iteration module of legacy program P61 to solve the creep problem under multi stress states (plane stress, plane strain, and axisymmetric states), and also introduces the implementation of the Removal Technique module, Restart Facility module, and the auto-select Time Step module.

2. In section 4.3, it reports the development detail of the in-house FE solver at grain boundary level. Specifically, it includes the structure of the Computational Framework, the creep body loads generation of grain boundary element, and the coding implementation of this element.

4.2 Programming the Creep Solver

In this project, an accurate and stable FE solver for the creep boundary value problem is the foundation. The development of this module is achieved by refactoring the P61 program, which is used to solve the nonlinear visco-plasticity of material originally [13]. In refactoring, these modules or blocks are retained, including:

1. Evaluate the project and allocate the size for matrices.
2. The blocks and loops for assembling, storing, and solving the global stiffness matrix.
3. The block and loop for the non-linear iteration of updating residual stress.

Compared with P61, three changes have been made, including:

1. In P61, the nonlinear iteration contained in one module is a fixed time step. However, the rate variables of the first and third stages of creep are higher and require smaller time steps, while the rate variables of the second stage are lower and larger time steps can be used. Therefore, in order to balance convergence and computational efficiency, this module is divided into two sub-modules: The first sub-module is used to obtain the applied elastic stress field to obtain the rate variable of all elements, and then to solve an acceptable time step based on the maximum rate variable of all elements. The second is used for numerical integration of constitutive equations and updating creep residual stress.

2. The constitutive equation used to describe material non-linearity was changed from visco-plasticity to creep.
3. Add modules and loops for removal function and restart facility.

4.2.1 Flow Diagram of Solver

The solver can be divided into four main blocks: import and initialization module, restart facility module, removal function module, and non-linear iteration module, as shown in Figure 4.2.1.

1. Import and initialization

It is the first module in the structure, which is used to import the input file into the main program by a fixed channel (in this program, the channel number is '11'). In this file, it includes the element information (type, number, node quantity per element, and node topology), node information (coordinates and DOF per node) boundary condition, material properties, loading information, and activation status of the restart facility. Based on the case information, the total number of (non-zero) node DOF is calculated to allocate the matrix size for the storage of the global stiffness matrix.

2. Restart Facility

This is the second module in the structure, which is used to restart the calculation at a specified breakpoint. When this module is activated, it will import the restart file into the main program by an independent channel (in this program, the channel number is '12'), and the stiffness matrix is

reassembled and solved before the next module. On the contrary, it jumps to the next module directly.

3. Removal Function

The third module in the structure, which is used to implement the removal function of failed elements. Firstly, it will compare the state of elements in the current iteration step with the previous step. If a new element failed, a submodule is triggered for the global stiffness matrix to be re-assembled and re-inversed (excludes the element stiffness contribution of failed element). If not, it will skip this module and enter the next module directly.

4. Non-linear iteration

It is the fourth module in the structure, which is refactored from the non-linear iterative module of P61 (marked with a red dotted line box). In the original program, the time step is determined (based on the unconditional numerical stability time step of the von Mises material [80]) before the iteration module, and it is fixed during the iteration. However, due to the high creep strain/damage rates in the first and third stages, the smaller time step is required to satisfy the convergence, while in the second stage (steady-state), low rates can use a larger time step, reducing the simulating time.

Therefore, this module is refactored to consider both convergence and efficiency.

The iteration of P61 is implemented in an element loop, and the flow of this loop is:

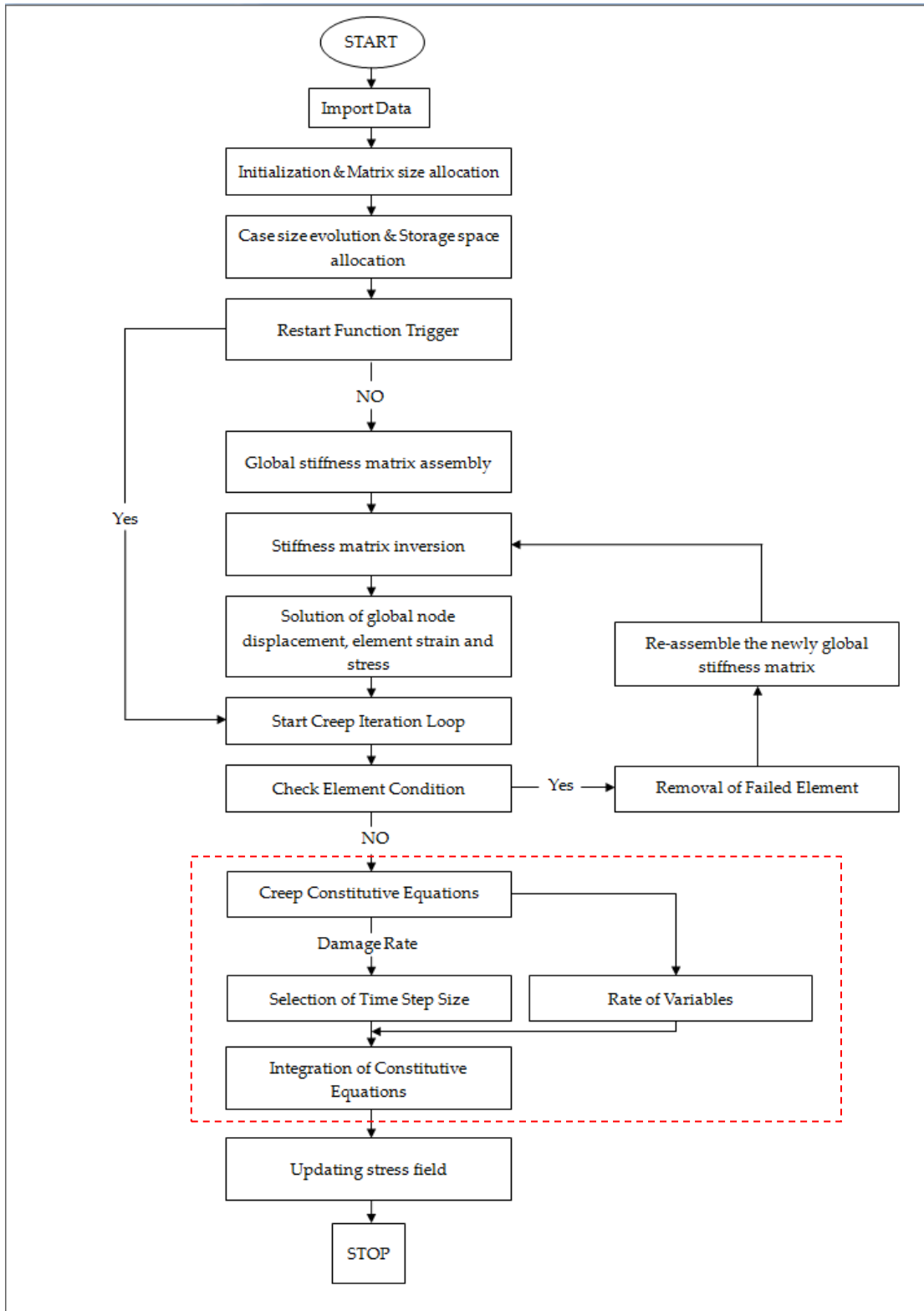


Figure 4.2.1 Flow Diagram of the Finite Element Solver.

First, import the node displacement into this loop to retrieve the elastic stress on the Gauss Integration point for each element, and then the constitutive equation is integrated by a fixed time step. Finally, the residual stress is updated.

The refactored non-linear iterative module divided the previous loop into two sub-loops. The first loop is used to retrieve the stress and calculate the rate-dependent variables in constitutive equations of all elements, and find out the maximum damage rate to calculate the acceptable time step size. Then in the second loop, the rate-dependent variable and time step obtained in the previous loop are imported into this cycle to integrate the constitutive equation and update the residual stress.

4.2.2 Implementation of Removal Technique

The introduction of the removal technique has been introduced in section 2.10, however, the implementation details are not explicitly documented. In this solver, the element's damage value ω becomes greater than the criterion with time, this element is considered to be a failure one and no longer have the ability to transmit any load or force [69,70].

This technique includes two parts:

1. Trigger module

Due to the assembly and re-solving of the stiffness matrix being expensive in CPU time, therefore, these two actions only are triggered again until the next element fails. In response to this demand, a subroutine 'execute' was

developed to implement this trigger function and the mechanism this subroutine is shown in Figure 4.2.2. In order to explain easy, a simple example with 5 elements and assume the NO.3 element fails at step i.

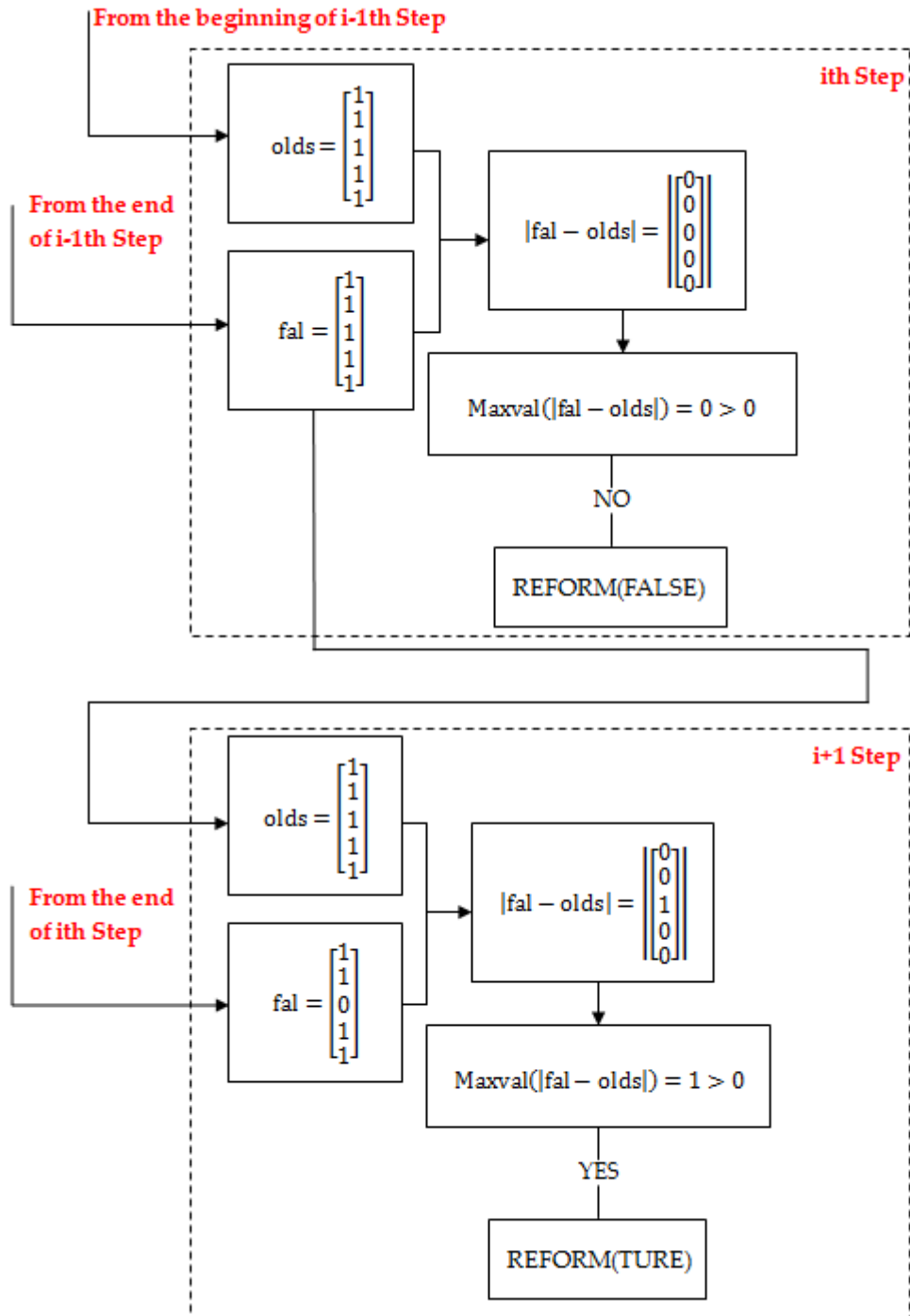


Figure 4.2.2 Flow Diagram Showing the process detail of the subroutine 'execute' to implement the trigger technique

The process of the trigger mechanism in the above simple example can be summarized as follows:

- 1) After each iteration step, the array [olds] stores the initial status of all elements in the previous iteration step, and the array [fal] stores the end status of all elements in the previous iteration step. In step (i), these two array 'fal' ($[1 \ 1 \ 1 \ 1 \ 1]^T$) and 'old' ($[1 \ 1 \ 1 \ 1 \ 1]^T$) of step i-1 are imported into the subroutine 'execute' at the beginning of this step.
- 2) In subroutine 'execute, these two arrays [fal] and [olds] are evaluated and subtracted to obtain a new array $[[fal - olds]]$ (this is a non-negative array, any negative elements are changed to their opposites). Based on the Fortran built-in function 'maxval', the maximum value of this array is found. If the maximum value is '1', it means that there is a new element failure, the logical variable 'reform' is set to true, otherwise, it is set to false when the maximum value is '0'. In this case, on failure occurs at step i-1, therefore the [fal] and [olds] are the same and the array $[[fal - olds]] = [0 \ 0 \ 0 \ 0 \ 0]^T$. Therefore, the maximum value of the array $[[fal - olds]]$ is 0 and the logical variable 'reform' is set to false, the trigger is not activated. After the subroutine 'execute', [fal] overwrites array [olds], then the array [olds] is ($[1 \ 1 \ 1 \ 1 \ 1]^T$). At the end step i, due the failure happens in element No.3, the array 'fal' is updated to $[fal]=[1 \ 1 \ 0 \ 1 \ 1]^T$.

3) At the beginning of step (i+1), the two arrays [fal] ($[1 \ 1 \ 0 \ 1 \ 1]^T$) and [olds] ($[1 \ 1 \ 1 \ 1 \ 1]^T$) of step i are imported into the subroutine 'execute', as mentioned in previous step, the output array $|[fal - olds]| = [0 \ 0 \ 1 \ 0 \ 0]^T$ and the maximum value of this array is '1', therefore the logical variable 'reform' is set to 'true' and the trigger is activated.

2. Application Module of Failure Conditions

The objective of this stage is to locate and remove the failed element in the structure based on the rupture criterion. The failed element is not able to participate in the calculation and assume any loads. The elastic stress of this element disappears immediately, and the deformation of the failed element is provided only by the permanent creep deformation. During the non-linear iteration, the stiffness and the creep body loads of this failed element may then be removed from the structure. The program starts a loop for assembling and re-solving the global stiffness matrix. The specific process in this loop is as follows: at the beginning, the new updated array 'fal' is imported into a loop which cycles all elements and assembles a global stiffness matrix, when the corresponding number in array 'fal' is '1', the element stiffness remains unchanged, otherwise, when the number is '0', the corresponding stiffness matrix of this element is re-evaluated. The new global stiffness matrix $[K]$ is stored by the lower triangle method, and the new inverse of the global stiffness matrix $[K]^{-1}$ is resolved by Gauss Factorization. After these calculations, the node displacement can be obtained by multiplying $[K]^{-1}$ by

total node loads, which includes the external loads and new creep body loads, and the creep body loads have removed the contribution of the failed element.

4.2.3 Implementation of Restart Facility

The introduction of the restart facility has been introduced in section 2.10, however, the implementation details are not explicitly documented. In this solver, the restart module has been incorporated in this procedure. In the procedure, two switches are set, one is used to output the 'restart.dat' file which contains the necessary information needed to restart the procedure at the selected iteration step, the other one is used to activate the restart function and open the channel to import the 'restart.dat' file to continue the calculation from the breakpoint.

Initially, the parameter 'restep' is imported into the main program to control the output frequency of the 'restare.dat' file, which allows the procedure to print a data set for this file after every 'restep' iteration steps. The details of the data set are summarized in Table 4.2.1.

Table 4.2.1 The variables and arrays in the data set.

Name	Description
evp	stores total of creep strains of grain element.
slide	the jump displacement of grain boundary element at the sliding direction.
esd	the jump displacement of grain boundary element at the normal direction.
p	the cavity density ρ of grain boundary part.
b	the damage variable β of grain boundary part.
bdylds	the self-equilibrating global node loads.
t	the total simulation time.
iters	the iteration step number.
fal	the state array of grain boundary element. (fine or failed)

The switch needs to be turned on when the procedure needs to use the restart function. The specific way to activate this switch is to assign the variable 'tri' in the 'inp' input file to a value of '1'. After that, the procedure will read in the 'restart.dat' file through the allocated channel (the default channel in this solver is 14). Finally, the non-linear iteration is continued by import the parameters required (as shown in Table 4.2.1).

4.2.4 Implementation of Auto-select Time Step Module

During the simulation, creep deformation and creep damage can be obtained by integrating the rate-dependent variables in the time domain. Under practical conditions, the structural-nonlinearity causes the stress concentration with the local high increase rate usually, and the creep feature occurs at a high increase rate in the primary and tertiary stages making the integral of the constitutive equations sensitive to the time step. Thus, the selection of the appropriate time step is meaningful for practical applications. In order to implement this function, a new pre-processing module is added to the main structure before the constitutive equation integration structure. The function of this module is to determine the time-step by the instantaneous damage increase rate since the final target value of the damage variable is the critical value (it is '1' in usual). In order to avoid the stress oscillation caused by big increments, and leads the nonlinear iteration un-converge, so the increment of each iteration step is controlled within ' $1/X$ ', where 'X' is the total number of expected iteration steps, and then the time step is ' $1/(X \cdot \dot{\omega})$ ', where $\dot{\omega}$ is the damage increase rate. By looping all elements to find the minimum value as the time-step in the current iteration step. After completing this pre-processing module, the program enters the Euler integration module. In the current version, as the mathematical background of the time step size control of the micro-constitutive equation has not been solved, it reduces the time step to satisfy the convergence of the calculation. Therefore, although

this module has been integrated into the procedure, it is only activated in the notched bar case study. When this module is not activated, a small step size will be used to avoid non-convergence of nonlinear.

4.3 Development of Solver for Grain Boundary Level

In this section, the main purpose is to show the creep mechanics constitutive equations and modeling of grain boundaries, which are included within the previous general FE solver to simulate the creep evolution of grain boundary with times. The procedure was developed from the previous general version, and the following techniques and blocks are retained:

1. The non-linear displacement iteration method.
2. The blocks for obtaining the solid element stiffness matrix for grain part.
3. The techniques for assembling, storing, and calculating of global stiffness matrix.

Compared with the previous solver, the solver for micro-creep is upgraded from solving single homogeneous material to bi-materials. In this project, the material is modeled by grain and grain boundary respectively, in which the grain part is modeled by traditional solid elements with a simple creep power law, and the grain boundary part is modeled by the Goodman element with a cohesive law mechanism.

4.3.1 Computational Framework

The structure of the micro-version is similar to the structure of the macro version. The main difference is the Goodman element is adopted to simulate the creep mechanism of the grain boundary part. Figure 4.3.1 shows the computational framework and the additional developed part is marked by red and blue dotted line boxes.

The additional block is developed to implement these functions:

1. Obtain the element stiffness matrix of the Goodman element and add the Goodman element stiffness matrix into the global stiffness matrix.
2. Obtain the elastic stress field (avoid negative relative displacement/negative normal stress) and integrate the creep constitutive equation of the grain boundary.
3. Calculate the un-balance creep body node loads of the Goodman element.

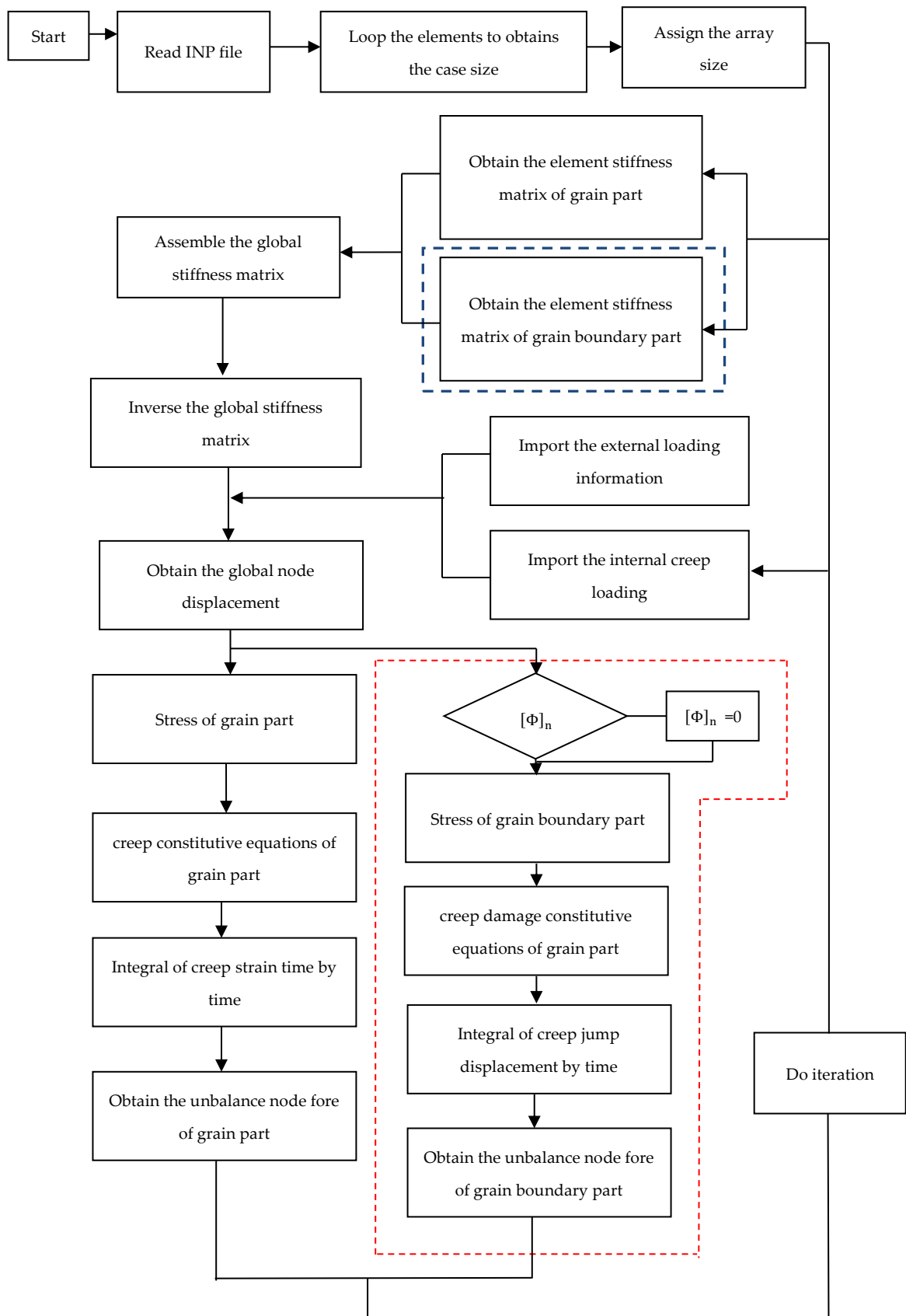


Figure 4.3.1 The flow diagram structure of the in-house procedure.

Reproduced from [45].

($[\Phi]_n$ is the relative displacement at normal direction)

4.3.2 Creep Body Loads Generation of the Goodman Element

The mathematical background of the 4-node Goodman element' stiffness matrix has been mentioned in section 2.7, here, we apply the same method for the solid element to calculate the creep body loads of Goodman element [3,11,13]. Compared with the body force generation of the conventional element type, the main difference is the numerical integration. Generally, the Gauss-Legendre method is adopted to evaluate the body loads at an element area. However, the integration of the body loads at the Goodman element length adopts the analytical integration method directly.

The creep body loads P_{CGB} :

$$P_{CGB} = \int [[B] \cdot [T]]^T * [D] * [\Phi] dL, \quad 4.3.1$$

where $[B]$ is the node-relative displacement matrix (E.Q 2.7.7), $[D]$ is the stress-relative displacement matrix (E.Q 2.7.9), $[\Phi]$ is the creep jump displacement, $[T]$ is the local-global coordinate transfer matrix and L is the length. Finally,

$$P_{CGB} = [[B] \cdot [T]]^T * [D] * [\Phi] * L \quad 4.3.2$$

4.3.3 Coding Implementation of Goodman element

In order to implement the mathematical background for obtaining the stiffness matrix of Goodman element, which has been mentioned in section 2.7, a block with three new subroutines ('element_inf', 'Loc-Gol', and 'new_km') and one legacy subroutine ('fsparv') have been developed to calculate the element stiffness matrix of Goodman element.

The flowchart of this block is presented in the Figure 4.3.2, this block is used to implement the blue marked part in the Figure 4.3.1.

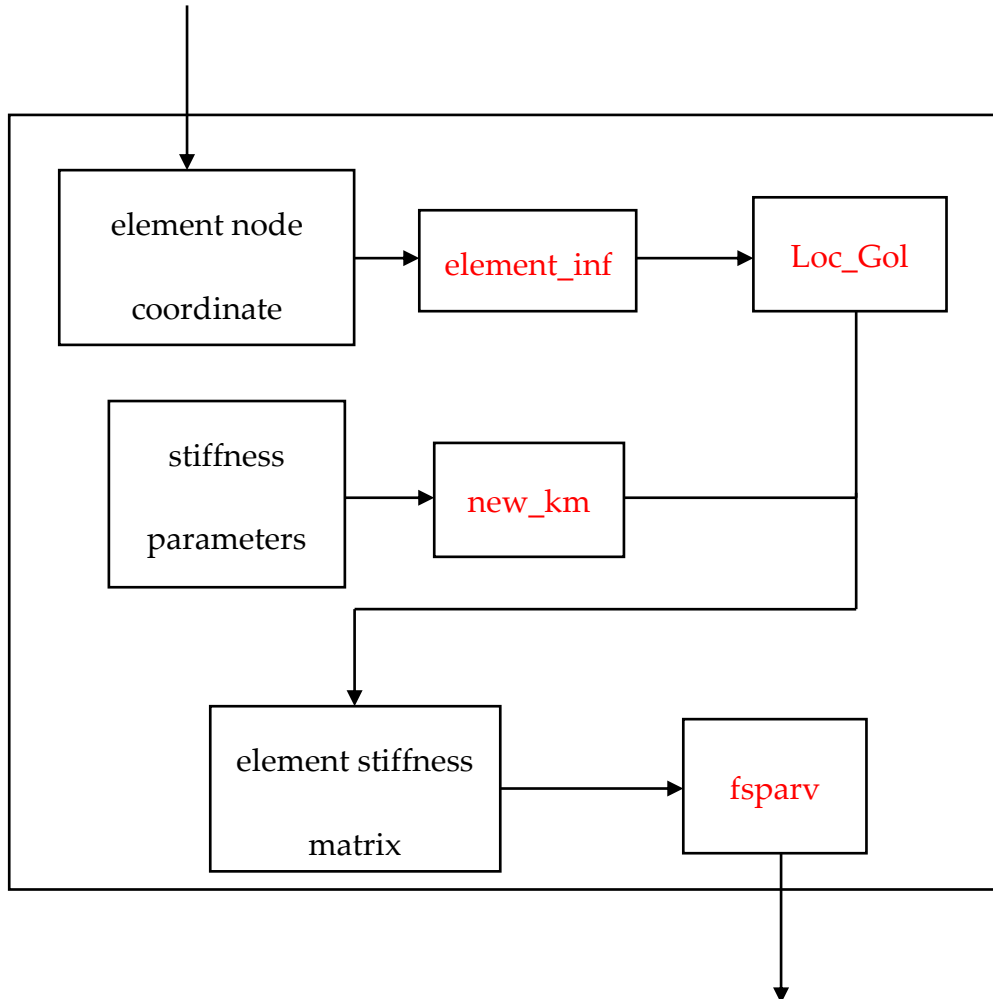


Figure 4.3.2 The flow diagram structure for obtaining the Goodman element stiffness matrix. (subroutines are marked by

The mathematical background of the Goodman element has been mentioned in section 2.7, and the above block is developed to implement it. The details of the block and the subroutines can be summarized below.

1. Introduction of this Block

The block is developed to obtain the stiffness matrix of Goodman element, in the beginning, the node coordinate and stiffness parameters of the element

are imported into this block, through two newly developed subroutines 'element_inf', 'Loc_Gol', and 'new_km' to obtain the element information (the length and the angle), rotation matrix ([T]), and stiffness matrix([D]) respectively. Then according to equation 2.7.12, we import the three variables directly into this equation to solve the element stiffness matrix. Finally, adopt the subroutine 'fsparv' to assemble the element matrix into the global stiffness matrix.

2. Introduction of Subroutines

1) element_inf

This subroutine is developed to return two pieces of information of the Goodman element: length and the angle matrix, which are adapted to obtain the element stiffness matrix in global coordinate system.

The import information of this subroutine is the matrix 'coord2', which store the global coordinate of element nodes. As shown in Figure 2.7.1, the upper one is selected as a demonstration here due to the upper and lower surface having the same angle and length.

Length (double-precision number)

The subroutine returns the length of the Goodman element, and it is named 'L' in the procedure.

$$L = \sqrt{(y_3 - y_2)^2 + (x_3 - x_2)^2} \quad 4.3.3$$

*Angle Matrix (2*2 matrix)*

$$\sin \theta = \frac{(y_3 - y_2)}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}}$$
$$\cos \theta = \frac{(x_3 - x_2)}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}}$$
4.3.4

The subroutine returns the matrix N (E.Q 2.7.16), and it has been named

'Angle' in the procedure.

2) 'Loc_Gol'

This subroutine is developed to return the rotation matrix, which are used to doing the coordinate system transmission of the Goodman element from local to global. The subroutine imports the matrix 'Angle' and returns a matrix T (16*16) (E.Q 2.7.16), and it has been named 'Angle_T' in the procedure.

3) new_km

This subroutine is developed to return the rigidity matrix [D], which is a 8*8 size matrix and named 'km' in the procedure. The subroutine imports the matrix 'kcoh', which stores the normal and separate rigidity of the Goodman element.

4) fsparv

This subroutine is used in assembling the element stiffness matrix into the global stiffness matrix, and the details of this subroutine have been mentioned in section 3.4 [13].

Chapter 5 Benchmark of the macro scale of In-house Procedure

5.1 Introduction

This chapter reports the benchmark progress of the In-house procedure. In order to make benchmark efficient and logical, a common way from simple to complex, linear to nonlinear was chosen [13]. It can be divided into two main parts: 1) a simple quadrilateral structure is chosen as the preliminary step to demonstrate the accuracy under uni-axial loading condition. Through this case study, the numerical stability and accuracy were verified, which paves the way for the subsequent multi-axis load condition. 2) the notched bar structure is chosen as the second step to demonstrate the accuracy under complex stress state. Based on this case study, it verifies the accuracy of the nonlinear iteration under the non-uniform stress field. The benchmark process can be summarized as follow:

- 1) The validation of the in-house procedure under the simple stress condition and the validation corresponds to the development progress. Firstly, verify the elastic module. According to the stress state, the benchmark contains three sub-case studies: plane stress, plane strain, and axisymmetric. In this stage, the techniques such as import interface, the element stiffness solution, and assembly, solution of equilibrium equation have been validated. Secondly, verify the creep module. In this stage, the

integral accuracy and the non-linear iterative stress update is verified.

- 2) The validation of in-house procedure under stress concentration condition by notched based FE model. The nonlinear iteration accuracy of the in-house procedure is verified by comparing the simulation results with Hyde's output from the rupture time and damage pattern.

This chapter primarily consists of three sections: 1) Introduction. 2) The preliminary benchmark of the procedure. 3) The validation of the in-house procedure via the numerical investigation of the Bar 267 notched-bar case study at 660 °C

5.2 The Preliminary Benchmark of the Procedure

The preliminary benchmark of the procedure verifies the numerical accuracy, numerical stability, and convergence of non-linear creep iterations via a simple Finite Element model. In this section, it presents the preliminary verification, which paves the way for the subsequent notched bar case study. The benchmark process corresponds to the sequence of the development, from linear elasticity to non-linear creep, from plane stress to plane strain to axisymmetric version. The strategy has been adopted to verify the 'HITSI' by D.Liu, which is efficient and logical [11]. Here, the uniform 4-node quadrilateral element type is used to generate the FE model. In this stage, three basic technique modules can be validated and be summarized below:

- The input interface which is used to import the case information. It includes element information, node information, topological structure information, material parameters, loading information and boundary condition.
- The existing techniques for calculating the element stiffness matrix, the assembling the global stiffness matrix, the storage of the global stiffness matrix, the assessment of case size to allocate the matrix size, and the solution of the equilibrium equation.
- The techniques for the non-linear iterative techniques for creep problems, including the calculation of the internal body loads due to creep deformation, the re-balancing of internal body loads, and integration of

creep constitutive equations.

The FE model applies in the benchmark is shown in Figure 5.2.1

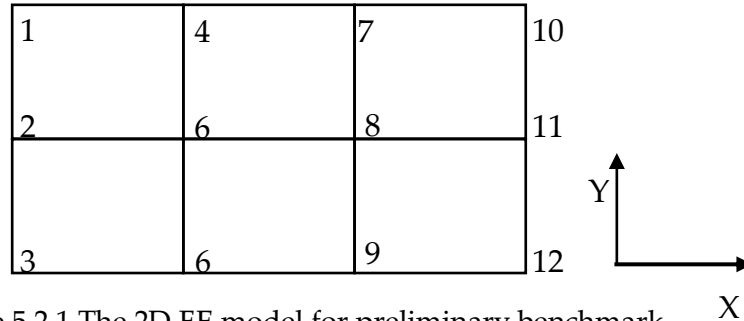


Figure 5.2.1 The 2D FE model for preliminary benchmark.

- *The Geometry information, Boundary Condition and Loading Information and Material Parameters.*

The generation of this FE model is implemented by two relevant standard subroutines, 'geom_rect' and 'mesh_size', to obtain the node coordinates and the topology of the element. They come from the library 'geom' directly. In this model, it contains 6 quadrilateral elements with 12 nodes. The coordinates of these 12 nodes is shown in Table 5.2.1 and the topology information are shown in Table 5.2.2. The topological orientation of the element is clockwise.

Table 5.2.1 The coordinates of the node. (unit: mm)

Node NO.	X direction	Y direction	Node NO.	X direction	Y direction
1	0	0	7	2	0
2	0	-1	8	2	-1
3	0	-2	9	2	-2
4	1	0	10	3	0
6	1	-1	11	3	-1
6	1	-2	12	3	-2

Table 5.2.2 The topology information of the element.

Element NO.	Topology Information (clockwise)
1	2, 1, 4, 6
2	3, 2, 6, 6
3	6, 4, 7, 8
4	6, 6, 8, 9
6	8, 7, 10, 11
6	9, 8, 11, 12

The boundary condition is imposed on the bottom-line nodes such that the displacement components to the Y direction and the left line nodes such that the displacement components to the X direction are always zero respectively. The implementation of the boundary condition is through add the constraint of the node, in this case, the node constraint information is listed in Table 5.2.3.

Table 5.2.3 The constrained node of boundary condition.

The Constraint Direction	Node Number
X direction	1, 2, 3
Y direction	3, 6, 9, 12

In this benchmark, the uniform loads are on the top surface. Based on the node loading factor calculation method, the equivalent node loading factor information of plane stress and plane strain case is the same, which is shown in Table 5.2.4 (a), and the axisymmetric case is shown in Table 5.2.4 (b).

Table 5.2.4 The equivalent node loading factor information.

Node NO.	X	Y
(a) Plane Stress and Plane Strain		
1	0.0	0.5
4	0.0	1.0
7	0.0	1.0
10	0.0	0.5
(b) Axisymmetric		
1	0.0	0.1666667
4	0.0	1.0
7	0.0	2.0
10	0.0	1.333333

5.2.1 Validation of the Elastic Part

Young's modulus E and Poisson's ratio ν are set to $1 \times 10^3 \text{MPa}$ and 0.3 respectively. A uniformly linear distributed load 60MPa is applied on the top line and the loading factor has been mentioned in Table 5.2.4. The assessment and discussion of the result under three stress states are shown below.

➤ *Plane stress case.*

In this plane stress case, the theoretical stress in the Y direction is 60MPa , in the X and the shear directions should be zero. According to the constitutive relationship between stress and strain which has been mentioned in section 2.6.1, the strain of three directions is:

$$\begin{aligned}\varepsilon_x &= \frac{b}{b^2 - a^2} \cdot \sigma_y \\ \varepsilon_y &= -\frac{a}{b^2 - a^2} \cdot \sigma_y \\ \gamma_{xy} &= 0\end{aligned}\tag{5.2.1}$$

where, $a = \frac{E}{1-\nu^2}$, $b = \frac{\nu \cdot E}{1-\nu^2}$.

According to the Equation 5.2.1 and the parameters, the theoretical strain in the X direction is -1.8×10^{-2} , in the Y direction is 6.0×10^{-2} . The simulated stress and strain are shown in Table 5.2.5, which have been shown in good agreement with the theoretical value and the error is negligible.

Table 5.2.5 The elastic stress field and strain field for the FE model under 60MPa uniform loading of plane stress state. (At Gauss Point)

Elastic Stress Field (Unit: MPa)			
Element No.	X Direction	Y Direction	Shear Direction
1	1.066×10^{-14}	60.000	8.006×10^{-15}
2	1.776×10^{-14}	60.000	3.155×10^{-14}
3	3.197×10^{-14}	60.000	1.334×10^{-14}
4	1.421×10^{-14}	60.000	-2.402×10^{-14}
5	2.487×10^{-14}	60.000	-5.338×10^{-15}
6	0.000	60.000	2.669×10^{-15}
Elastic Strain Field			
1	-1.799×10^{-2}	5.999×10^{-2}	2.082×10^{-17}
2	-1.799×10^{-2}	6.000×10^{-2}	8.204×10^{-17}
3	-1.799×10^{-2}	6.000×10^{-2}	3.469×10^{-17}
4	-1.800×10^{-2}	6.000×10^{-2}	-6.245×10^{-17}
6	-1.799×10^{-2}	6.000×10^{-2}	-1.388×10^{-17}
6	-1.800×10^{-2}	5.999×10^{-2}	6.939×10^{-18}

Plane strain case.

In this plane stress case, the theoretical stress in the Y direction is 60 MPa, in the X and the shear directions should be zero. According to the constitutive relationship between stress and strain which has been mentioned in section 2.6.1, the strain of the three directions is:

$$\begin{aligned}\varepsilon_x &= \frac{b}{b^2 - a^2} \cdot \sigma_y \\ \varepsilon_y &= -\frac{a}{b^2 - a^2} \cdot \sigma_y \\ \gamma_{xy} &= 0\end{aligned}\tag{5.2.2}$$

where, $a = \frac{E \cdot (1-\nu)}{(1+\nu) \cdot (1-2\nu)}$, $b = \frac{E \cdot \nu}{(1+\nu) \cdot (1-2\nu)}$.

According to the Equation 5.2.2 and the parameters, the theoretical strain in the X direction is -2.34×10^{-2} , in the Y direction is 6.46×10^{-2} . The simulated stress and strain are shown in Table 5.2.6, which have been shown in good agreement with the theoretical value and the error is negligible.

Table 5.2.6 The elastic stress field and strain field for the FE model under 60MPa uniform loading of plane strain case. (At Gauss Point)

Elastic Stress Field (Unit: MPa)			
Element No.	X Direction	Y Direction	Shear Direction
1	1.066×10^{-14}	60.000	2.669×10^{-16}
2	-3.663×10^{-16}	60.000	-1.868×10^{-14}
3	-3.663×10^{-16}	60.000	-1.001×10^{-14}
4	2.487×10^{-14}	60.000	1.068×10^{-14}
6	-1.776×10^{-14}	60.000	6.338×10^{-16}
6	7.106×10^{-16}	60.000	4.003×10^{-16}
Elastic Strain Field			
1	-2.339×10^{-2}	-6.469×10^{-2}	6.939×10^{-18}
2	-2.339×10^{-2}	-6.469×10^{-2}	-4.867×10^{-17}
3	-2.339×10^{-2}	-6.469×10^{-2}	-2.602×10^{-17}
4	-2.339×10^{-2}	-6.469×10^{-2}	2.776×10^{-17}
6	-2.339×10^{-2}	-6.469×10^{-2}	1.388×10^{-17}
6	-2.339×10^{-2}	-6.469×10^{-2}	1.041×10^{-17}

Axisymmetric case.

In this case study, the theoretical stress in the Y direction is 60 MPa, in the X and the shear directions should be zero. However, compared with the previous two cases, only one Gaussian Point cannot satisfy the accuracy requirement for the element integration, therefore, here, it is realized by four Gaussian Points.

According to the constitutive relationship between stress and strain which has been mentioned in section 2.6.1, the strain in three directions are:

$$\begin{aligned}\varepsilon_X &= \frac{b}{2 \cdot b^2 - a^2 - a \cdot b} \cdot \sigma_Y \\ \varepsilon_Y &= \left(\frac{a+b}{b}\right) \cdot \left(\frac{b}{2 \cdot b^2 - a^2 - a \cdot b}\right) \cdot \sigma_Y\end{aligned}\quad 5.2.3$$

$$\varepsilon_X = \varepsilon_Z$$

$$\gamma_{XY} = 0$$

where, $a = \frac{E \cdot (1-\nu)}{(1+\nu) \cdot (1-2\nu)}$, $b = \frac{E \cdot \nu}{(1+\nu) \cdot (1-2\nu)}$.

According to the Equation 5.2.3 and the parameters, the theoretical strain in the X and the Z direction are -1.8×10^{-2} , in the Y direction is 6.0×10^{-2} .

The simulated stress and strain are shown in Table 5.2.7, which have been shown in good agreement with the theoretical value and the error is negligible.

Table 5.2.7 The elastic stress field and strain field for the FE model under 60MPa uniform loading of axisymmetric case. (At Gaussian Point)

Elastic Stress Field (Unit: MPa)					
Element No.	Gaussian Point No.	X	Y	Shear	Z
1	1	-1.924×10^{-7}	69.999	1.493×10^{-7}	-1.924×10^{-7}
	2	-6.116×10^{-8}	69.999	1.361×10^{-7}	-6.116×10^{-8}
	3	-1.216×10^{-7}	69.999	6.179×10^{-8}	-1.216×10^{-7}
	4	9.678×10^{-9}	69.999	4.763×10^{-8}	9.678×10^{-9}
2	1	-1.496×10^{-9}	69.999	2.800×10^{-8}	-1.496×10^{-9}
	2	2.877×10^{-8}	69.999	2.922×10^{-8}	2.877×10^{-8}
	3	-7.647×10^{-9}	69.999	7.828×10^{-9}	-7.847×10^{-9}
	4	2.272×10^{-8}	69.999	9.038×10^{-9}	2.272×10^{-8}
3	1	-1.119×10^{-8}	60.000	-4.994×10^{-9}	-1.291×10^{-8}
	2	-1.623×10^{-8}	69.999	-3.071×10^{-9}	-1.739×10^{-8}
	3	-8.841×10^{-10}	60.000	-1.360×10^{-9}	2.396×10^{-8}
	4	-1.236×10^{-8}	69.999	6.623×10^{-10}	4.478×10^{-9}
4	1	-2.328×10^{-8}	69.999	-1.916×10^{-8}	7.738×10^{-9}
	2	-1.018×10^{-8}	69.999	-1.460×10^{-8}	-1.083×10^{-8}
	3	-7.233×10^{-9}	69.999	6.419×10^{-9}	1.409×10^{-8}
	4	8.218×10^{-9}	69.999	7.639×10^{-10}	2.266×10^{-8}
6	1	-2.014×10^{-8}	69.999	1.322×10^{-8}	-1.621×10^{-8}

	2	3.681×10^{-10}	60.000	1.286×10^{-8}	3.476×10^{-9}
	3	-1.063×10^{-8}	60.000	-8.499×10^{-10}	3.824×10^{-9}
	4	8.346×10^{-9}	60.000	-1.207×10^{-9}	1.973×10^{-8}
6	1	-6.463×10^{-9}	60.000	6.382×10^{-9}	1.366×10^{-8}
	2	3.763×10^{-9}	60.000	7.122×10^{-9}	1.946×10^{-8}
	3	-6.086×10^{-9}	60.000	-2.472×10^{-9}	1.746×10^{-8}
	4	-2.707×10^{-9}	60.000	-1.732×10^{-9}	2.196×10^{-8}
Elastic Strain Field					
1	1	-1.800×10^{-2}	6.999×10^{-2}	3.882×10^{-10}	-1.800×10^{-2}
	2	-1.800×10^{-2}	6.999×10^{-2}	3.614×10^{-10}	-1.800×10^{-2}
	3	-1.799×10^{-2}	6.999×10^{-2}	1.607×10^{-10}	-1.799×10^{-2}
	4	-1.799×10^{-2}	6.999×10^{-2}	1.238×10^{-10}	-1.799×10^{-2}
2	1	-1.799×10^{-2}	6.999×10^{-2}	7.281×10^{-11}	-1.799×10^{-2}
	2	-1.799×10^{-2}	6.999×10^{-2}	7.696×10^{-11}	-1.799×10^{-2}
	3	-1.799×10^{-2}	6.999×10^{-2}	2.036×10^{-11}	-1.799×10^{-2}
	4	-1.799×10^{-2}	6.999×10^{-2}	2.349×10^{-11}	-1.799×10^{-2}
3	1	-1.800×10^{-2}	6.000×10^{-2}	-1.298×10^{-11}	-1.800×10^{-2}
	2	-1.800×10^{-2}	6.999×10^{-2}	-7.986×10^{-11}	-1.800×10^{-2}
	3	-1.800×10^{-2}	6.000×10^{-2}	-3.636×10^{-11}	-1.799×10^{-2}
	4	-1.800×10^{-2}	6.999×10^{-2}	1.462×10^{-12}	-1.799×10^{-2}
4	1	-1.800×10^{-2}	6.999×10^{-2}	4.982×10^{-11}	-1.799×10^{-2}

	2	-1.800×10^{-2}	6.999×10^{-2}	3.769×10^{-11}	-1.799×10^{-2}
	3	-1.799×10^{-2}	6.999×10^{-2}	1.409×10^{-11}	-1.799×10^{-2}
	4	-1.799×10^{-2}	6.999×10^{-2}	1.960×10^{-12}	-1.799×10^{-2}
6	1	-1.800×10^{-2}	6.000×10^{-2}	3.438×10^{-11}	-1.800×10^{-2}
	2	-1.800×10^{-2}	6.000×10^{-2}	3.346×10^{-11}	-1.800×10^{-2}
	3	-1.800×10^{-2}	6.000×10^{-2}	-2.210×10^{-12}	-1.799×10^{-2}
	4	-1.800×10^{-2}	6.000×10^{-2}	-3.139×10^{-12}	-1.799×10^{-2}
6	1	-1.800×10^{-2}	6.000×10^{-2}	1.669×10^{-11}	-1.799×10^{-2}
	2	-1.800×10^{-2}	6.000×10^{-2}	1.862×10^{-11}	-1.799×10^{-2}
	3	-1.800×10^{-2}	6.000×10^{-2}	-6.426×10^{-12}	-1.799×10^{-2}
	4	-1.800×10^{-2}	6.000×10^{-2}	-4.602×10^{-12}	-1.799×10^{-2}

5.2.2 Validation of the creep part

The benchmark of the creep part of three stress states is performed in this section, the FE model has been mentioned before. The uniformly distributed linear load is 60MPa and the KRH creep damage constitutive equation is adopted in this verification. In this case study, it simulates the creep evolution of 0.6Cr0.6Mo0.26V ferritic steel at 690°C. The material constants for this material are given below Table 5.2.8 [75]. In the creep iteration, the time step is set as $\Delta t = 0.5h$.

Table 5.2.8 The material constants for 0.6Cr0.6Mo0.26V ferritic steel at 690°C.

Reproduced from [81]

A	$2.1618 \times 10^{-9} \text{MPa h}^{-1}$	H*	0.6929
B	0.20624 MPa^{-1}	K_C	$9.2273 \times 10^{-6} \text{MPa}^{-3} \text{h}^{-1}$
C	1.8637	v	2.8
h	$2.4326 \times 10^6 \text{ MPa}$		

Plane stress case.

In this FE model, it is a kind of uniform stress condition. During the creep non-linear creep iteration, since each element has the same creep deformation, the generated body loads are equal in opposite directions, and there is no stress redistribution. This case is a kind of uni-axial stress state. Here, the NO.1 element is chosen as a sample to present the results, this non-linear iterative process lasted for 73936 steps, the failure time is 36967.5 hours, and the creep strain at failure is 0.179934297613029. The simulated rupture time and strain at failure have been shown in good agreement with the reference result [59]. The creep strain cure and damage evolution curve are shown in Figure 5.2.2 and Figure 5.2.3, respectively.

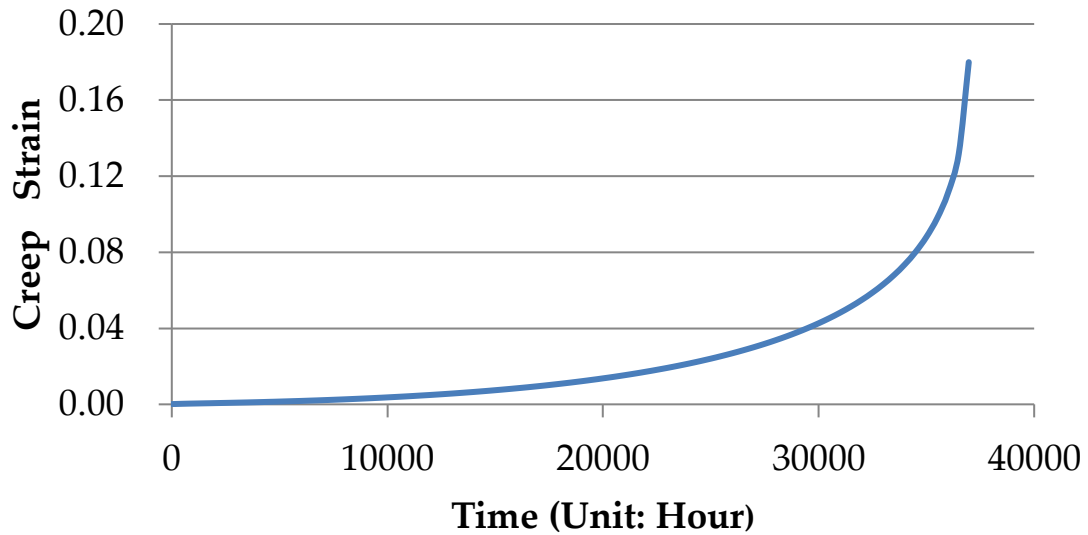


Figure 5.2.2 The creep strain curve of the plane stress case.

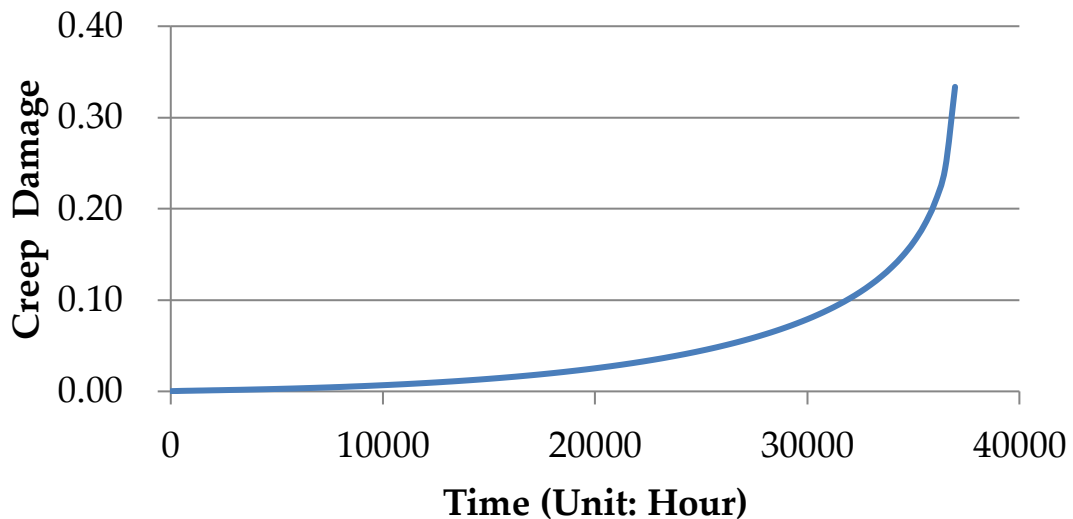


Figure 5.2.3 The creep damage evolution curve of plane stress case.

Plane strain case.

In the plane strain case, under the same parameter constants, constitutive equations, and loading. Here, the NO.1 element is chosen as a sample to present the results, the iteration lasted for 73936 steps, the failure time is 36967.5 hours, and the creep strain at failure is 0.179934297613012. The simulated rupture time and strain at failure have been shown in good

agreement with the reference result [51]. The creep strain cure and damage evolution curve are shown in Figure 5.2.4 and Figure 5.2.5, respectively.

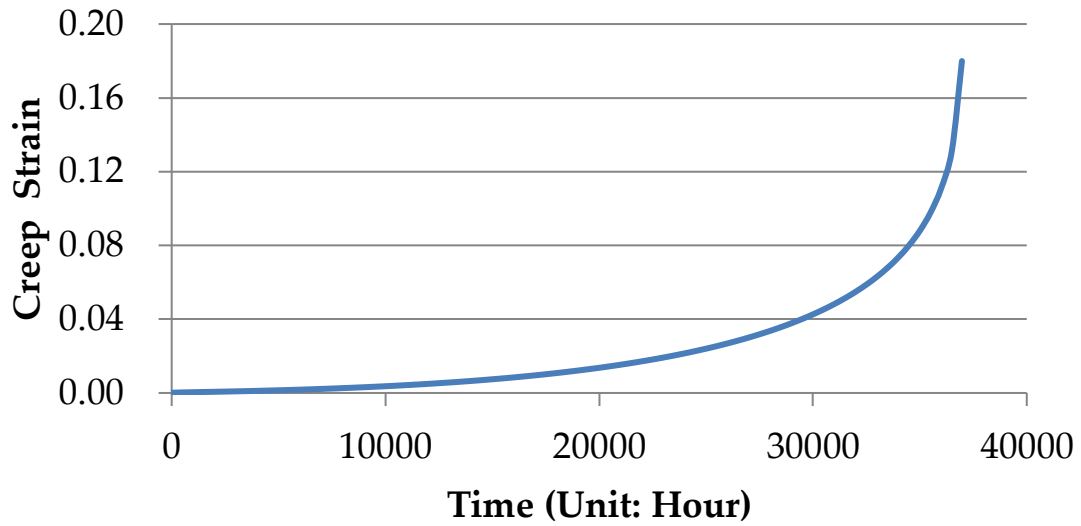


Figure 5.2.4 The creep strain curve of the plane strain case.

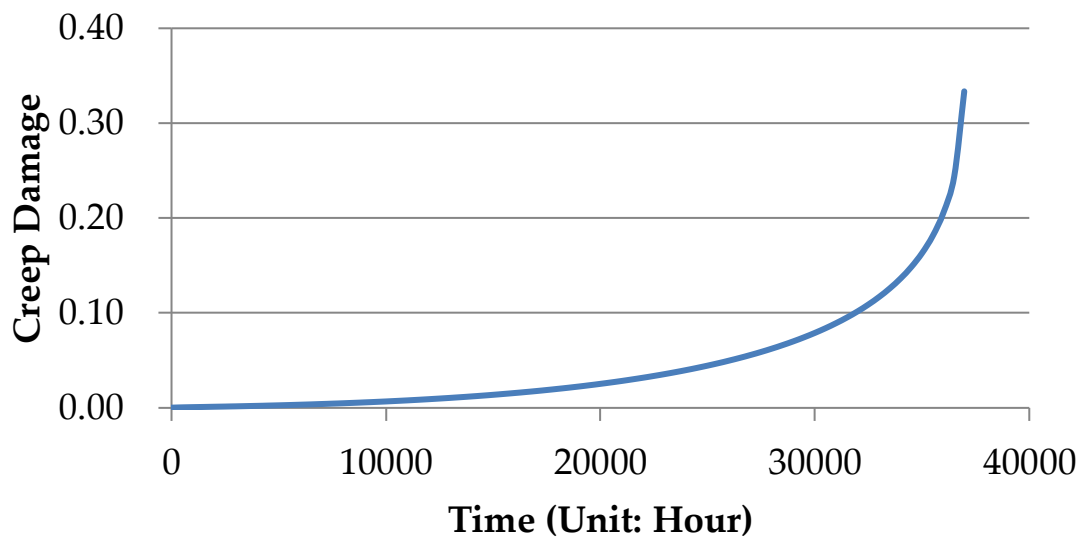


Figure 5.2.5 The creep damage evolution curve of plane stress case.

Axisymmetric case.

In axisymmetric case, under the same parameter constants, constitutive equations, and loading. However, the element numerical integration is implemented by four Gaussian Points.

Here, the No.1 Gaussian point of the NO.1 element is chosen as a sample to present the results, the iteration lasted for 73936 steps, the failure time is 36967.5 hours, and the creep strain at failure is 0.179934294619527. The simulated rupture time and strain at failure have been shown in good agreement with the reference result [59]. The creep strain cure and damage evolution curve are shown in Figure 5.2.6 and Figure 5.2.7, respectively.

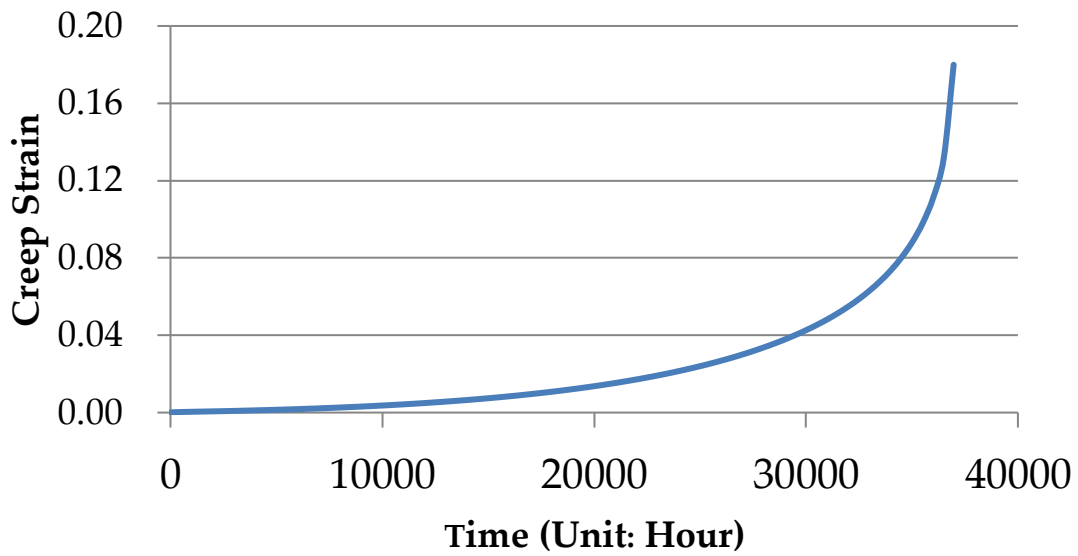


Figure 5.2.6 The creep strain curve of the axisymmetric case.

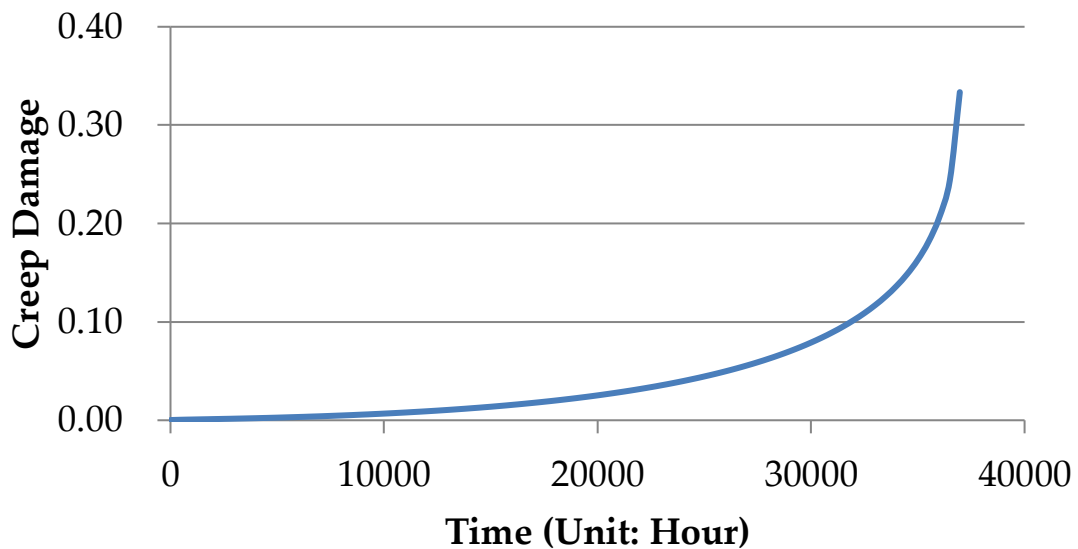


Figure 5.2.7 The creep damage evolution curve of axisymmetric case.

5.3 The Validation of the In-house Procedure via the numerical investigation of the Bar 267 Notched-Bar Case Study at 660 °C

5.3.1 Introduction

The 2-dimensional FE in-house procedure for creep damage simulation has been developed from the P61[10]. The previous verification of the procedure is under the uniform load condition, which is a simple stress condition.

However, the ultimate practical significance of the procedure is to apply the numerical techniques at the high-stress gradients and complex state of stress conditions. Therefore, it is indispensable to benchmark the stability and accuracy of the procedure under moderate stress concentration. The general way is to do the benchmark simulation of the notched bar test.

In the creep test, there are typically two types of the notched bar specimen that are commonly used, which are the circular Bridgman notch and the British Standard notch. These two structures typify two different stress conditions, in the body of the circular Bridgman notch is subjected by the uniform state of complex multi-axial stress condition and in the British Standard notch, the high gradients and concentration of the stress are observed close to the notch part. For the notched bar specimen, the indicator of the performance can be expressed by the mean stress act on the minimums section of the notched bar. In the creep simulation, since the axisymmetric of the notched bar, the problem is reduced to a 2D axisymmetric case and a quarter of the specimen is selected to establish the FE model.

In order to make the verification process efficient and logical, the convenient way is to find an appropriate notched bar analysis that has been validated. Therefore, a notched bar case has been chosen to verify the solution of the procedure, which the FE simulation and the experiment have been done by Hyde [71]. The reason for adopting this case is that: complete parameters, given the dimensions of the notched bar, and published the accurate rupture time and damage contour. In this case, the specimen is a Bridgman notched bar with Bar 267 material, and the test temperature is 660 °C. These fellow aspects will be controlled during the verification to keep the same with Hyde's simulation: boundary condition, loading, mesh, geometry, parameters, and constitutive equation. The generation and mesh of the FE model are based on the Abaqus package and the model information read into the procedure via the interface by modifying the INP file into a procedure-acceptable format, and finally displaying the results through the FEMGV platform.

A general benchmark strategy is employed here as mentioned before is from the linear elasticity to the creep non-linearity. The solution of the creep is a kind of the initial value problem; therefore, the accuracy of the initial elastic stress field affects the final simulation result directly. The second stage is to verify the creep part, which mainly includes the following aspects: 1) the position of the first failure element. 2) the evolution of the damage growth. 3) the damage pattern. 4) the rupture time.

5.3.2 Description of the Bar 267 Bridgman Notched Bar Case Study

The geometry of the Bridgman notched bar at one quadrant at the cylindrical polar coordination system is defined in Figure 5.3.1, the main size ration of this sample is $R/a = 0.67$ and $b/a = 1.67$, where the R is the ratio of the notch, a is the ration of the minimums section and b is the ratio of the maximums section. The material of the specimen is Bar 267, which is a special type of P91 steel. The characteristic of this steel is that the rupture strength is much lower than the standard average value of P91 steel.

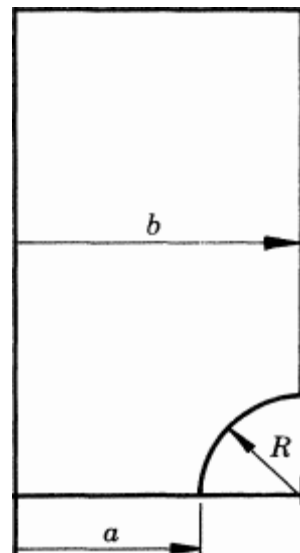


Figure 5.3.1 The geometry of the Bridgeman circular notch bar.
Produced by [3].

FE model.

The main dimensions of the test specimen are $a = 3.76$ mm, $b = 6.26$ mm, and $R = 2.6$ mm. The notched bar has been tested at the temperature of 660 °C and has been loaded on the top surface to produce the nominal stress 93 MPa in the minimum notch section. For the Finite Element Analysis, this problem can be idealized to a 2D axisymmetric case. The FE model is shown in Figure 5.3.2.

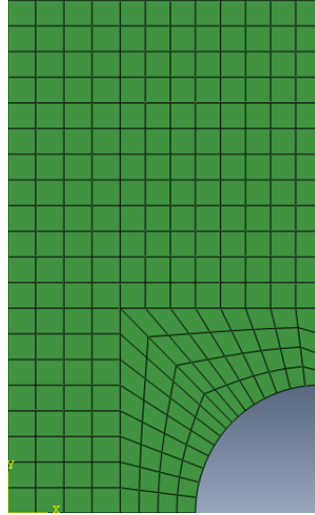


Figure 5.3.2 The FE model of the Bridge circular notched bar.

The generation and meshing of this model are based on the Abaqus package, and the meshing pattern follows Hyde's early form. The model consists of 240 axisymmetric quadratic elements with 8 nodes and the area integration is implemented by 4 Gaussian integration points. The boundary conditions are imposed on these left line nodes such that the displacement components to the X direction and bottom-line nodes such that the displacement components to the Y direction are always zero respectively.

Constitutive Equations.

The Kachanov type constitutive equations was adopted here to describe the creep behavior of the materials. It developed from the power law stress sensitivity. The damage state is described by a single variable to depict the creep evolution under the multi-axial stress condition. The details of this constitutive equations have been mentioned in section 3.2.1, therefore are not present here. The parameters of B267 at the temperature of 660°C have been determined by Hyde, shown in Table 5.3.1.

Table 5.3.1 The constants in the constitutive equations of Bar 267 steel. (at 660 °C)
 Reproduced from [71]

Material	A'	n	m	B'	ϕ	χ	α
Bar 267	1.092×10^{-20}	8.462	-4.764×10^{-4}	3.637×10^{-17}	7.346	6.879	0.216

The Solution of Hyde.

In the notched bar analysis, the predicted rupture time with the axisymmetric Finite Element Method was 996.2h. The damage contour is shown in Figure 5.3.3.

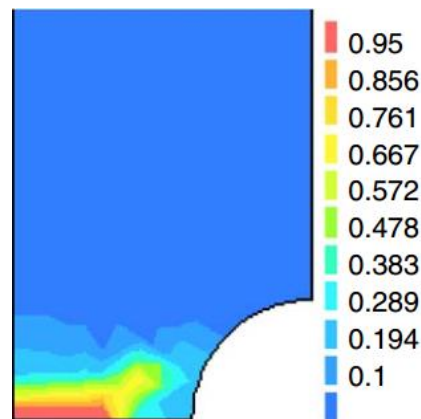


Figure 5.3.3 The Hyde's FE solution of the damage pattern at times at the rupture time $t_f = 996.2h$. Produced by [71]

The damaged area happens at the root of the specimen and the failure area width is approximately 1/2 of the minimum cross-sectional area. The tendency for the failure growth is towards the notch at a direction of about 46 degrees from the bottom axis.

5.3.3 Result and Discussion

In the FE analyses, the determination of the failure element is based on the damage value at the Gaussian point, in which at least one of four Gaussian point's damage reaches the threshold value ($\omega = 1$). At that point, the loading capability of the failure element disappears. Such elements needed to be

removed from the structure, however, in the current version, the alternative method is to rapidly reduce the stiffness of the element.

The loading and Boundary Condition

In this geometry, the uniform loads 33 MPa on the top surface can produce 93 MPa mean stress in the minimum notch section. The equivalent node loading factor information is shown in Table 5.3.2.

Table 5.3.2 The equivalent nodal loading factor information in axial direction.

Node Number	Loading Factor	Node Number	Loading Factor
39	0.0	740	1.1667239
667	0.106382612	169	0.626047777
129	0.106382612	716	1.33346721
670	0.316147636	168	0.708419427
128	0.210766036	692	1.600210499
672	0.626912633	167	0.791791444
127	0.316147648	668	1.666966433
674	0.737677611	166	0.876163206
38	0.398231496	644	1.833697233
777	0.833237707	166	0.968634439
161	0.468304606	619	2.000440622
764	0.999980636	43	0.620963042
160	0.641676136		

The implementation of the boundary condition is through adding the constraint of the node, in this case, a total of 67 nodes were imposed constraints. The coordinate origin NO. 6 is fixed. The boundary conditions are listed in Table 5.3.3.

Table 5.3.3 The constrained node of boundary conditions.

The Constraint Direction	Node Number
X direction	6, 6, 9, 12, 16, 18, 21, 24, 27, 39, 130, 131, 132, 133, 134, 136, 136, 137, 138, 139, 140, 301, 319, 336, 363, 370, 387, 404, 421, 469, 478, 487, 496, 606, 614, 623, 632, 641,660,669,668
Y direction	2, 3, 6, 47, 48, 49, 66, 67, 68, 279, 282, 286, 288, 293, 296, 299, 302,

The validation of the Elastic Part.

As mentioned before, the solution of creep is an initial value problem, therefore, the accuracy of the beginning elastic stress field needs to be controlled. The effective way is to benchmark with the elastic stress field with the result of ABAQUS.

The elastic stress contour is shown in Figure 5.3.4.

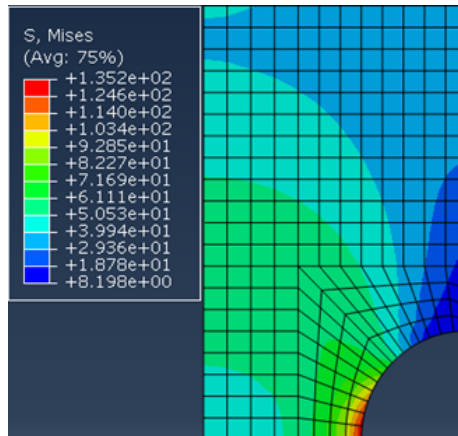


Figure 5.3.4 The elastic stress field obtained from ABAQUS.

Randomly select 6 elements from different stress gradients, compare the stress with the output of in-house procedure and do the error analysis of the Von Mises stress at each Gaussian Point (GP). According to the percentage errors shown in Table 5.3.4, it clearly shows the elastic stress field obtained from the in-house procedure which is in good agreement with the result from the Abaqus and the percentage error is negligible.

Table 5.3.4 The percentage error of stress field between Abaqus and In-house Procedure.

Element No.	GP No.	Abaqus	In-house	Error
4	1	102.067	102.067	0.00030269%
	2	103.189	103.189	0.00047641%
	3	124.614	124.614	0.00036609%
	4	126.799	126.799	0.00006449%
8	1	43.0187	43.019	0.00002809%
	2	43.4998	43.499	0.00010638%
	3	42.726	42.726	0.00006827%
	4	43.2064	43.206	0.00004174%
36	1	62.7046	62.704	0.00001890%
	2	61.1109	61.1112	0.00004836%
	3	63.6302	63.630	0.00007079%
	4	61.3676	61.368	0.00006047%
64	1	69.3284	69.328	0.00002231%
	2	69.166	69.166	0.00006491%
	3	69.2306	69.230	0.00000868%
	4	68.8782	68.878	0.00004690%
238	1	16.4681	16.468	0.00003876%
	2	14.6474	14.647	0.00012486%
	3	12.6828	12.683	0.00016033%

	4	11.3892	11.389	0.00030269%
--	---	---------	--------	-------------

Damage Evolution and Error Analysis.

The implementation of the post-process in this procedure is based on the FEMGV package. The evolution of the damage pattern is shown in Figure 5.3.5, 5.3.6, 5.3.7 and 5.3.8. The entire simulation process lasted 26334 steps and the simulation time was 1007 h. Failure occurs first at the root of the notched bar and then toward the notch laterally. The specific simulation data is listed as: the first failure is NO.8 element, which occurs in the 10092th step and the simulation time is 982h, the second failure is NO.7 element, which occurs in the 16647th step and the simulation time is 997h, the third failure element is NO.6 element, which occurs in the 20682th step and the simulation time is 1003h, the fifth failure element is NO.6 element, which occurs in the 26334th step and the simulation time is 1007h.

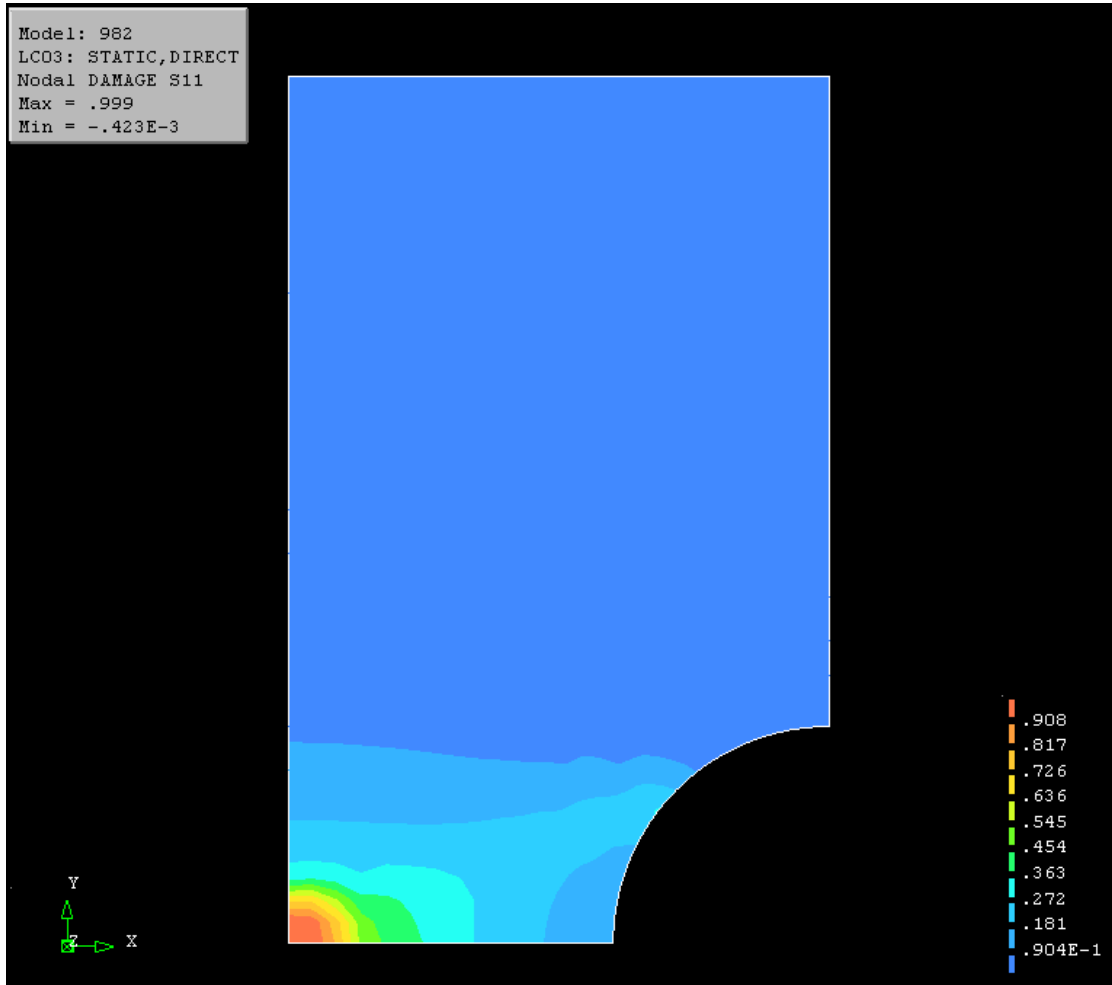


Figure 5.3.5 The evolution of the damage pattern at 982h.

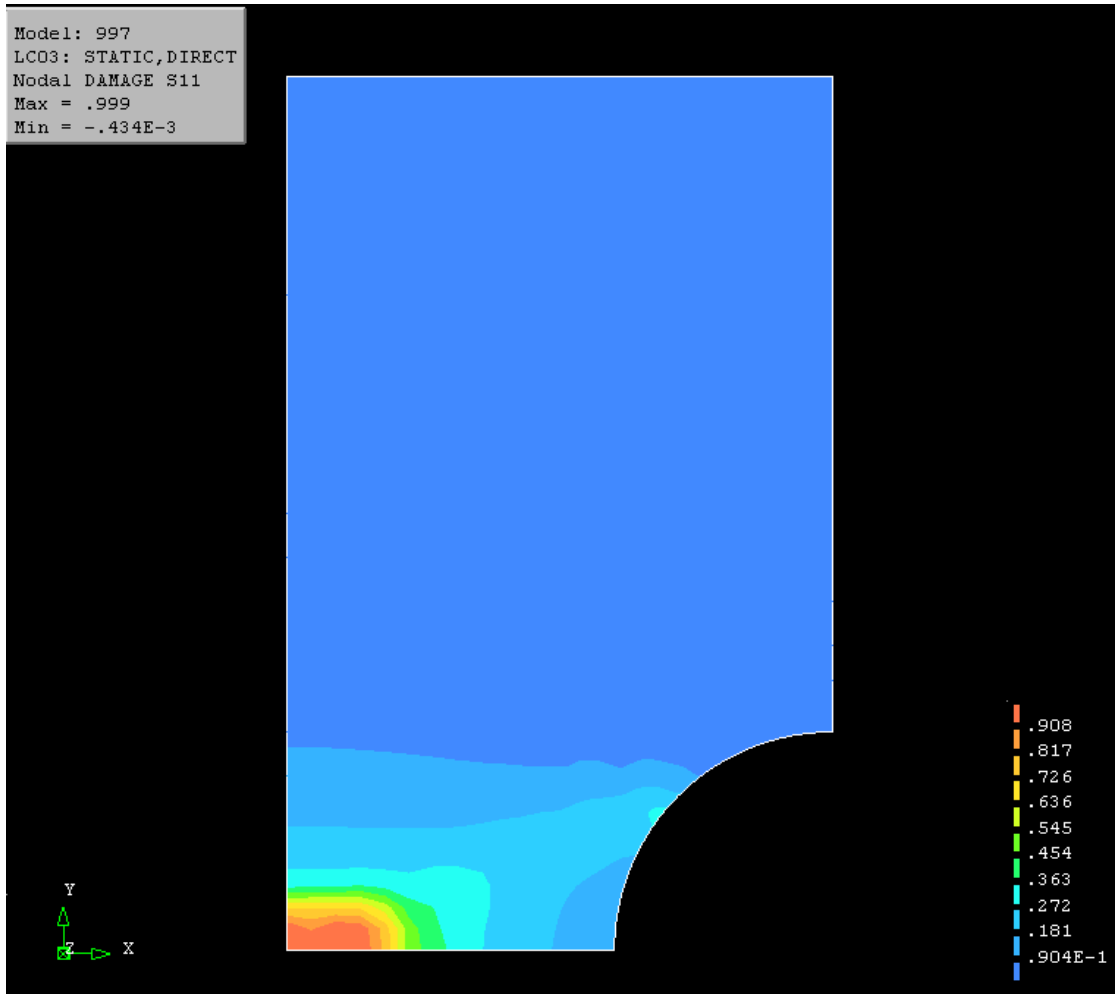


Figure 5.3.6 The evolution of the damage pattern at 997h.

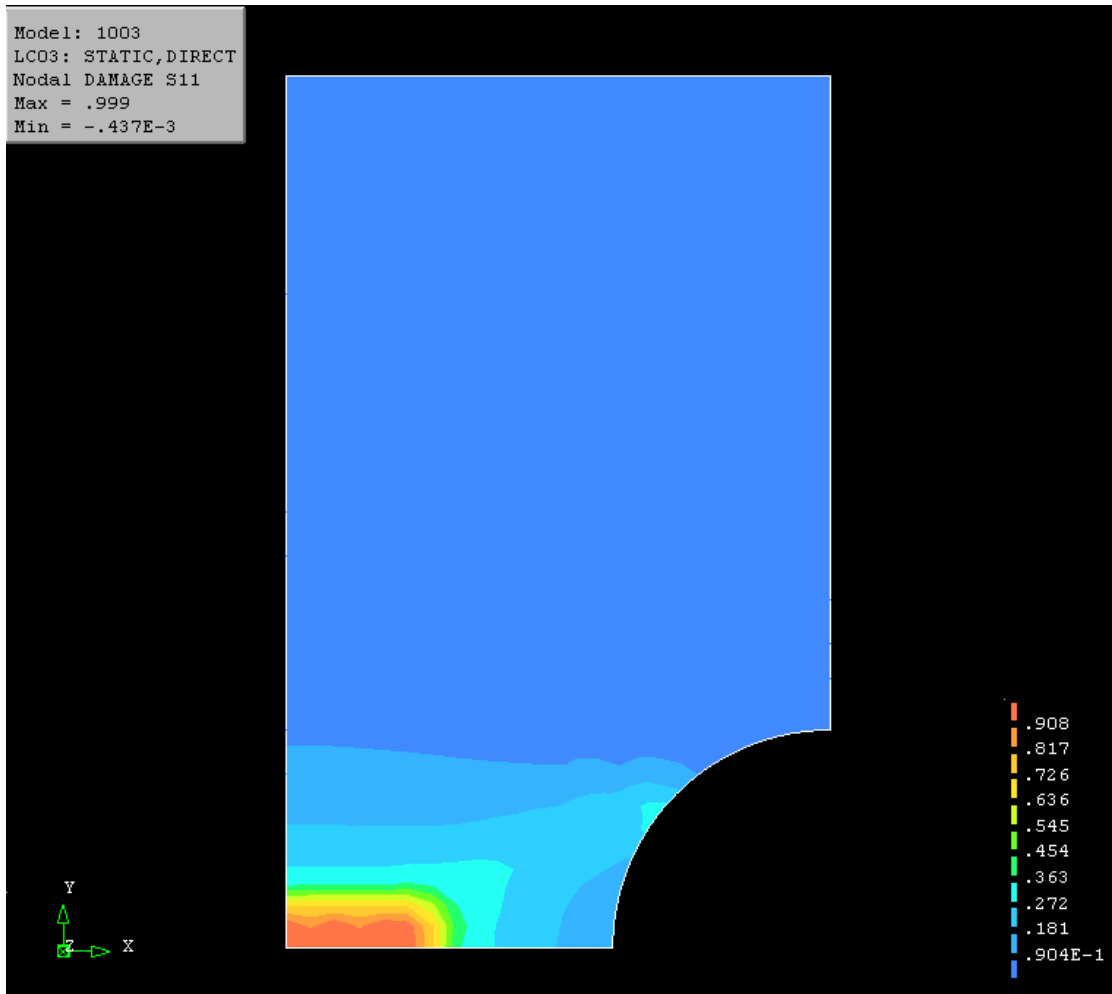


Figure 5.3.7 The evolution of the damage pattern at 1003h.

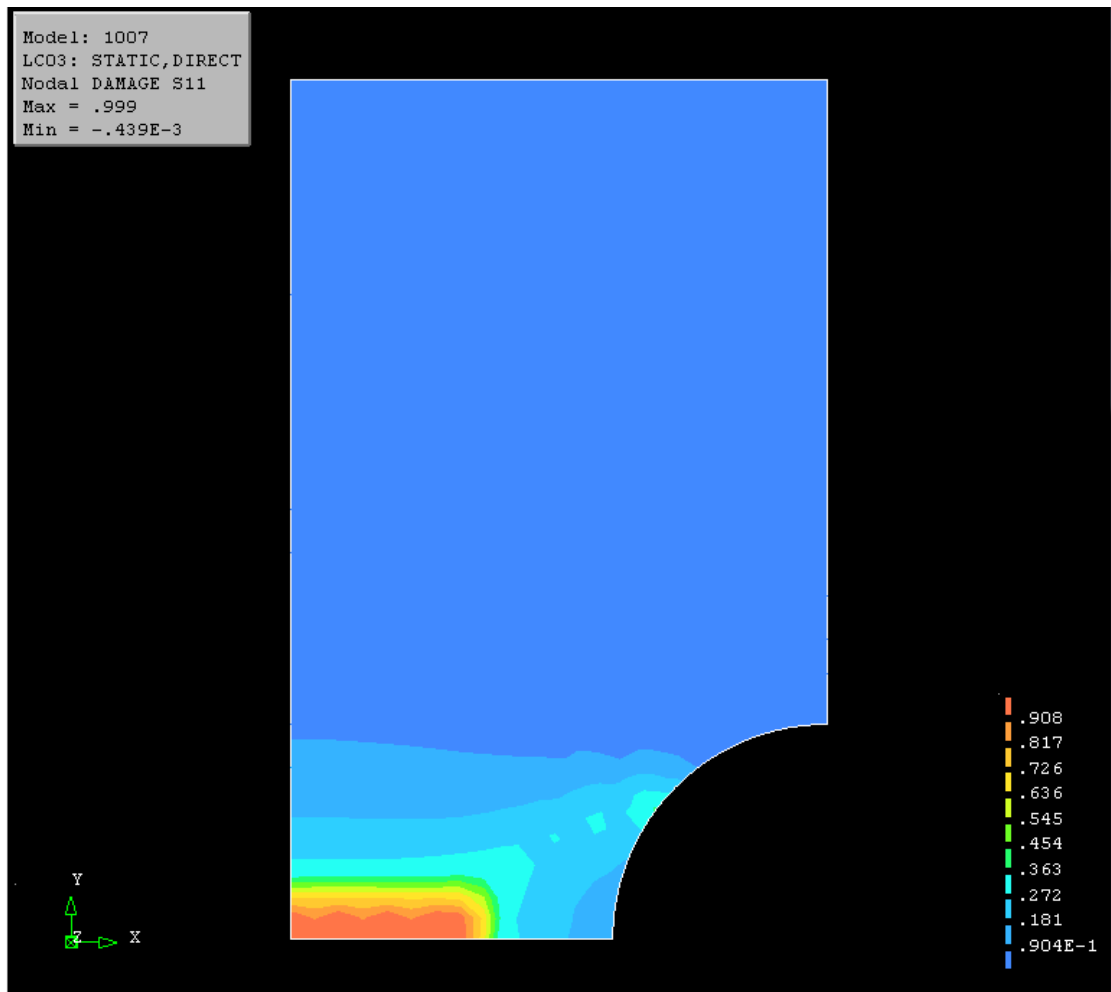


Figure 5.3.8 The evolution of the damage pattern at 1007h.

According to the damage contour of the simulations from Hyde, the failure definition of the notched bar is that the damaged width on the root is about half of the minimum cross-sectional length. To reach this damage level, Hyde's simulation spent 996.2h and the in-house procedure spent 1007h, and the percentage error is shown in Table 5.3.5.

Table 5.3.5 The percentage error of rupture time between Hyde's result and in-house procedure's output.

$$\text{The percentage error of Rupture time } t_f = \frac{1007-996.2}{996.2} = 1.08\%$$

5.3.4 Discussion

The in-house procedure has been developed and applied for the FE analysis of the creep damage evolution of the Bar 267 Bridgman notched bar case. The purpose of the section is to benchmark the accuracy and reliability of the procedure under complex stress state. More specifically, it first reports the elastic validation part, the elastic stress field have been shown in good agreement with the results from the Abaqus package. Through this part, these parts of the procedure are verified: 1) the assembling, storing and solving of the global stiffness matrix. 2) the stability and accuracy of the interface used to import the 'INP' files, boundary conditions and loading information into the procedure. Secondly, according to the benchmark of the non-linear creep solution, these parts of the procedure are verified: 1) the accuracy of the integral method of the KRH creep constitutive equation. 2) the accuracy and stability of the non-linear iterations, including a) the generation of the creep body node. b) the combination of internal and external node force. c) the reliability of the removal function. d) the reliability of the automatic time step selection function. 3) the reliability of the interface used to import the result information into the FEMGV to display the damage contour.

Chapter 6 The Validation and Application of In-house Procedure at Grain Boundary Level

6.1 Introduction

This chapter reports the benchmark and application of the In-house procedure for creep damage mechanics at grain boundary level. In order to make the benchmark logical and efficient, initially a bi-grain model was employed to verify the numerical stability and accuracy of procedure at grain boundary level. Then, applies the procedure to implement a polycrystal case study.

The benchmark stages in this chapter can be summarized as following:

1. Bi-grain benchmark test. At this stage, the validation relies on two FE models, the first one is the grain boundary parallels to the x-axis and the stress in a normal direction, which demonstrates the accuracy and stability in the normal direction. In the second FE model, the grain boundary has an angle with the x-axis, which demonstrates the accuracy and stability in the separated direction. In these two FE models, the strategy of the validation corresponds to the development stages, the main process is from linear elasticity to creep non-linearity.
2. Polycrystalline application. At this stage, the procedure applies to simulate the grain boundary level's creep evolution of Copper-antimony alloy at 823K. In this case study, the FE model contains 20 grains and 162

Grain boundaries.

6.2 The Validation of the In-house Procedure via the numerical investigation of the Bi-grains Case Study

6.2.1 Introduction

In this section, the validation logic is from uni-axial to multi-axial, from linear to nonlinear, therefore, the Bi-grains FE model proposed by Yu.C [9] is used to simulate the uni-axial stress state. It preliminarily verifies the procedure's numerical stability and accuracy and paves the way for the subsequent polycrystalline case study.

Two types of bi-grains structures were employed in the benchmark. In the first case, no shear sliding happens, the main purpose is to validate the numerical stability and accuracy of the procedure at normal direction. It consists of two rectangular grains and a single non-thickness grain boundary, as depicted in Figure 6.2.1.

Initially, at the unloaded state, the upper and lower surface of the grain boundary is coincident. For simplicity, we only consider the plane strain condition here. The grain part is modeled by the solid plane strain element and the grain boundary is modeled by the Goodman element.

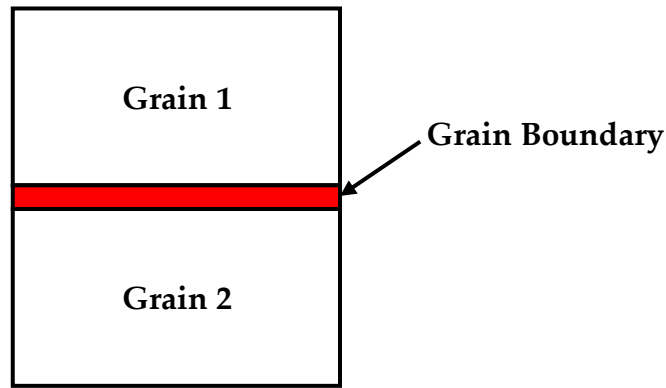


Figure 6.2.1 The schematics showing the bi-grains structure (The red area is the no thickness grain boundary). Reproduced from [9]

The validation can be devised of two stages:

1) Linear benchmark.

The three modules of the procedure are verified

- a) the interface part, which used to import the geometry information, the boundary condition, the loading information, etc.
- b) the solution the 2D Goodman element stiffness matrix
- c) the assembly of the global stiffness matrix, which is achieved by combining the stiffness matrix of the Goodman element with the traditional element stiffness matrix.

2) Non-linear benchmark.

The numerical accuracy and stability are verified

- a) the accuracy of the integral subroutine for the grain boundary constitutive equations.
- b) the generation of Goodman's body load due to the creep deformation.
- c) the generation of the global body load due to the grain's and grain

boundary's deformation.

- d) the accuracy of the elastic stress field for each non-linear iteration step.

Validation of Elastic Part

6.2.2 Validation of the Elastic Part with quadrilateral element

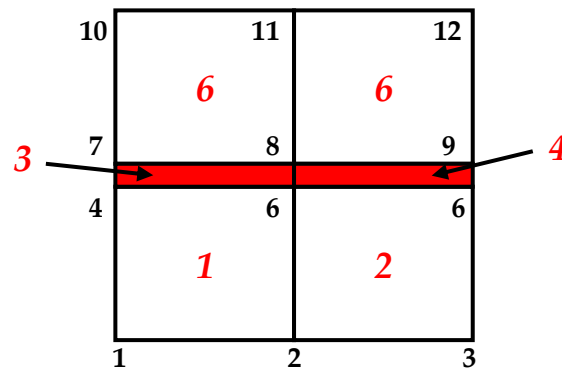


Figure 6.2.2 The mesh of the Bi-grains structure with quadrilateral element used in this case study (The red number is the element NO. and the black number is the Node NO.).

This FE model consists of two element types (as shown in Figure 6.2.2), the four nodes quadratic plane strain element with one Gaussian integration point for the grain part and the four nodes Goodman element for the grain boundary part.

- *The Geometry information, Boundary Condition and Loading Information and Material Parameters.*

In this FE model, there are 12 nodes to form 6 elements. The coordinates of these nodes and the topology information are shown in Table 6.2.1 and Table 6.2.2 respectively.

Table 6.2.1 The coordination of the nodes. (units: μm)

Node NO.	X direction	Y direction	Node NO.	X direction	Y direction
1	0.0	0.0	7	0.0	0.6
2	1.0	0.0	8	1.0	0.6
3	2.0	0.0	9	2.0	0.6
4	0.0	0.6	10	0.0	1.0
6	1.0	0.6	11	1.0	1.0
6	2.0	0.6	12	2.0	1.0

Table 6.2.2 The topology information of the element.

	Element NO.	Topology Information(clockwise)
Grain 1	6	7, 10, 11, 8
	6	8, 11, 12, 9
Grain 2	1	1, 4, 6, 2
	2	2, 6, 6, 3
Grain Boundary	3	4, 7, 8, 6
	4	6, 8, 9, 6

The boundary condition is imposed on the bottom-line nodes such that the displacement components to the Y direction and the left line nodes such that the displacement components to the X direction are always zero respectively.

The implementation of the boundary condition is through adding the constraint of the node, in this case, the node constraint information is listed in

Table 6.2.3.

Table 6.2.3 The constrained node of boundary condition.

The Constraint Direction	Node Number
X direction	1, 4, 7, 10
Y direction	1, 2, 3

In this geometry, the uniform loads 30 MPa on the top surface and the equivalent node loading factor information is shown in Table 6.2.4.

Table 6.2.4 The equivalent node loading factor information.

Node Number	Loading Factor	
	X direction	Y direction
10	0.0	0.6
11	0.0	1.0
12	0.0	0.6

Note: Node Loads Force equal loading factor times uniform loads stress.

In order to simplify the process, the elastic parameters are set and based on these parameters and the theoretical result can be obtained. In this case, the elastic parameters have shown in Table 6.2.5.

Table 6.2.5 The elastic parameters for the validation.

Grain	Young's Modulus		100000MPa
	Passion Ratio		0.3
Grain Boundary	Elastic Modulus	Normal Direction	6000000N/ μm^3
		Separate Direction	3000000 N/ μm^3

➤ *Result and Discussion.*

In this case, the contribution of the global stiffness matrix comes from two parts, the solid element stiffness of grain and the Goodman element stiffness of grain boundary. The assembly method used of the global stiffness matrix is "Element-By-Element" and the solution method of the global stiffness matrix adopted here is "Choleski". Initially, after the global stiffness matrix has been assembled, the node load information is read into the procedure and the node displacements for this FE model have been obtained. Therefore, the deformation for each element can be calculated by the [B] matrix in this stage of recovering the deformation at interpolation, for grain, the element deformation represents by the strain at the Gaussian Point, for grain boundary, the element deformation is represented by the relative displacement at the original point under the local coordinate. Then, the element deformation for the FE model has been output and shown in Table 6.2.6. The element stresses at interpolation point can be obtained by the calculation of the element's deformation and the [D] matrix, for grain, it is the strain-stress matrix and for the grain boundary, it is the relative displacement-stress matrix. The stress is shown in Table 6.2.7.

Table 6.2.6 The elastic deformation for the FE model under 30MPa uniform loading.

Grain (Strain)			
Element No.	X Direction	Y Direction	Shear Direction
1	-1.170×10^{-4}	2.730×10^{-4}	8.674×10^{-19}
2	-1.170×10^{-4}	2.730×10^{-4}	-9.768×19
6	-1.170×10^{-4}	2.730×10^{-4}	1.301×10^{-18}
6	-1.170×10^{-4}	2.730×10^{-4}	-6.606×10^{-19}
Grain Boundary (Relative Displacement) (μm)			
Element No.	X Direction	Y Direction	
3	3.263×10^{-19}	0.600×10^{-4}	
4	-1.804×10^{-19}	0.600×10^{-4}	

Table 6.2.7 The element elastic stress for the FE model.

Grain (Unit: MPa)			
Element No.	X Direction	Y Direction	Shear Direction
1	-3.730×10^{-14}	30.000	3.336×10^{-14}
2	-3.020×10^{-14}	29.999	-3.763×10^{-14}
6	-3.66×10^{-16}	30.000	6.004×10^{-14}
6	6.161×10^{-14}	30.000	-2.602×10^{-14}
Grain Boundary (Unit: MPa)			
Element No.	X Direction	Y Direction	
3	4.879×10^{-14}	30.000	
4	-3.663×10^{-16}	30.000	

For the grain, the theoretical stress in the Y direction is 30MPa and in X and shear direction should be zero. For the grain boundary, the theoretical stress in the Y direction is 30MPa and in the X direction should be zero. According to Table 6.2.7, the results show in good agreement with the theoretical stress. Thus, the technique for the elastic solution of this procedure has been validated.

Through the benchmark of the procedure for the Bi-grains' model, these techniques have been validated: 1) obtained the Goodman element for the grain boundary part. 2) the assembly and solution of the global stiffness matrix. 3) the recovering of the element stress at the interpolation point.

➤ *Validation of Compression Condition*

The previous section verified that the relative displacement occurs when the grain boundary is under tensile stress (positive loading). However, under actual conditions, some grain boundaries may be compressed. Due to the impermeability of the grain part, negative relative displacement is not allowed. Using the same FE model, mesh, boundary conditions, and loading point coefficients, replace the tensile stress with compressive stress (-30MPa). The elastic deformation of each element is shown in Table 6.2.8. The element stresses at the interpolation point (Gaussian Point) is shown in Table 6.2.9. Since the grain boundary is under a compressed state, in order to stop the two grains being inserted into each other, the upper and lower surfaces of the

grain boundary are locked, and no relative displacement of the grain boundary occurs.

Table 6.2.8 The elastic deformation for the FE model under -30MPa uniform loading.

Grain (Strain)			
Element No.	X Direction	Y Direction	Shear Direction
1	1.169×10^{-4}	-2.729×10^{-4}	1.084×10^{-19}
2	1.170×10^{-4}	-2.730×10^{-4}	-1.423×10^{-19}
6	1.169×10^{-4}	-2.730×10^{-4}	-8.936×10^{-20}
6	1.170×10^{-4}	-2.730×10^{-4}	5.421×10^{-20}
Grain Boundary (Relative Displacement) (μm)			
Element No.	X Direction	Y Direction	
3	0.000	0.000	
4	0.000	0.000	

Table 6.2.9 The element elastic stress for the FE model.

Grain (Unit: MPa)			
Element No.	X Direction	Y Direction	Shear Direction
1	-1.243×10^{-14}	-30.000	4.170×10^{-15}
2	-1.065×10^{-14}	-30.000	-5.473×10^{-15}
6	-7.105×10^{-15}	-30.000	-3.436×10^{-15}
6	0.000	-30.000	-2.085×10^{-15}
Grain Boundary (Unit: MPa)			
Element No.	X Direction	Y Direction	
3	0.000	0.000	
4	-3.663×10^{-16}	30.000	

6.2.3 Validation of the Elastic Part with triangle element

The results presented in this section has been published in the special issue (Creep and High Temperature Deformation of Metals and Alloys) of Metals, and it can be found in section 3.1 of the publication [45]. Q.Xu designed this Bi-grain structure with the triangle element type to eliminate the effect of unbalanced shear stress on the result(as shown in Figure 6.2.3). In this FE model, eight triangle plane strain elements are used to form two grains and one Goodman element for the grain boundary. The area integration of the triangle element is implemented by one Gaussian integration point.

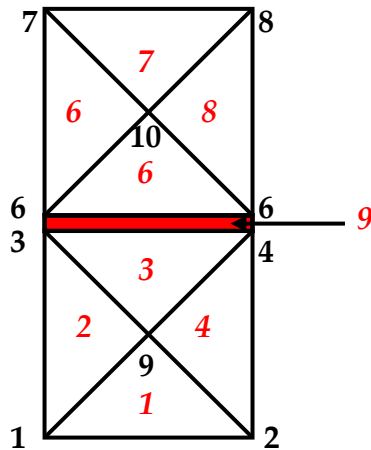


Figure 6.2.3 The FE model of the bi-grains structure with triangle element (the red area is the no thickness grain boundary). Reproduced from [45]

➤ *The Geometry information, Boundary Condition and Loading Information and Material Parameters.*

In this FE model, there are ten nodes to form eight elements. The coordinate of eight nodes and the topology information are shown in Table 6.2.10 and

Table 6.2.11 respectively.

Table 6.2.10 The coordinate of the nodes. (units: μm)

Node NO.	X direction	Y direction	Node NO.	X direction	Y direction
1	0.0	0.0	6	1.0	1.0
2	1.0	0.0	7	0.0	2.0
3	0.0	1.0	8	1.0	2.0
4	1.0	1.0	9	0.6	0.6
6	0.0	1.0	10	0.6	1.6

Table 6.2.11 The topology information of the element.

	Element NO.	Topology Information (clockwise)
Grain 1	1	1, 9, 2
	2	1, 3, 9
	3	3, 4, 9
	4	4, 2, 9
Grain 2	6	6, 10, 6
	6	6, 7, 10
	7	7, 8, 10
	8	8, 6, 10
Grain Boundary	9	3, 6, 6, 4

The boundary condition is imposed on the bottom-line nodes such that the displacement components to the Y direction and the left line nodes such that the displacement components to the X direction are always zero respectively.

The node constraint information is listed in Table 6.2.12.

Table 6.2.12 The constrained node of boundary condition.

The Constraint Direction	Node Number
X direction	1, 2
Y direction	1, 3, 6, 7

In this geometry, the uniform loads 20 MPa on the top surface. The equivalent node loading factor information is shown in Table 6.2.13.

Table 6.2.13 The equivalent node loading factor information.

Node Number	Loading Factor	
	X direction	Y direction
7	0.0	0.6
8	0.0	0.6

Note: Node Loads Force equal loading factor times uniform loads stress.

In order to simplify the procedure, the elastic parameters are set and based on these parameters, the theoretical result can be obtained. In this case, the elastic parameters have shown in Table 6.2.14.

Table 6.2.14 The elastic parameters.

Grain	Young's Modulus		100000MPa
	Passion Ratio		0.3
Grain Boundary	Elastic Modulus	Normal Direction	6000000N/ μm^3
		Separate Direction	3000000 N/ μm^3

➤ *Result and Discussion.*

The specific point of stress for the elastic model has been shown in Table 6.2.15. For the grain, the theoretical stress in the Y direction is 20MPa and in the X and the shear direction should be zero. For the grain boundary, the theoretical stress in the Y direction is 20MPa and in the X direction should be zero. According to the Table 6.2.15, for the grain element part, the maximum stress in the X direction is 1.066×10^{-14} occurs in NO.2 element, and in shear direction, the maximum stress is 4.170×10^{-16} MPa occurs in NO.6 element, which is negligible as expected. For the grain boundary part, the stress at

separated directions are negligible (2.033×10^{-16} MPa). In summary, the results show good agreement with the theoretical stress. Thus, the technique for the elastic solution with the triangle element of this procedure has been validated.

Table 6.2.15 The simulation stress for each element.

Grain (Unit: MPa)			
Element No.	X Direction	Y Direction	Shear Direction
1	0.000	20.000	-3.128×10^{-16}
2	1.066×10^{-14}	20.000	-2.086×10^{-16}
3	0.000	20.000	-1.043×10^{-16}
4	-6.329×10^{-16}	20.000	-2.086×10^{-16}
6	0.000	20.000	3.126×10^{-16}
6	-6.329×10^{-16}	20.000	4.170×10^{-16}
7	-7.106×10^{-16}	20.000	2.086×10^{-16}
8	1.776×10^{-16}	20.000	0.000
Grain Boundary (Unit: MPa)			
Element No.	X Direction	Y Direction	
9	2.033×10^{-16}	20.000	

6.2.4 Validation of the Elastic Part of FE model with Angle

The main purpose of the previous three case studies (Section 6.2.1, Section 6.2.2, and Section 6.2.3) was to benchmark the in-house procedure under the mesh with two different solid elements, quadrilateral and triangle type.

However, these two previous cases only verified a special condition where the grain boundary is parallel to the X axial, in which case no separated stress is generated. Therefore, this section considers a benchmark case study of a bi-crystal structure with angle.

The FE model of two grains is modeled by two triangle plane strain elements and a Goodman element for modeling the single grain boundary (shown in Figure 6.2.4.).

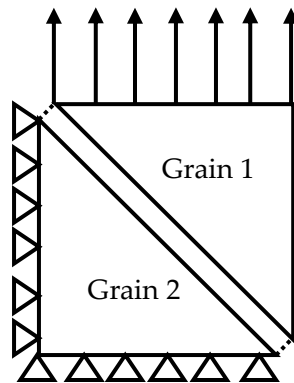


Figure 6.2.4 The FE model of the bi-grains structure with angle. Reproduced from [45]

- *The Geometry information, Boundary Condition and Loading Information and Material Parameters.*

In this FE model, there are a total of six nodes to form three elements. The rotation angle of the grain boundary direction to the X axial (anti-clockwise) is 135° . In order to be logical and efficient, the parameters and the normal stress on the grain boundary need to be consistent with these two previous case studies. Therefore, based on the geometric relationship, a uniform load of 40 MPa is applied on the top surface in the Y direction. The node coordinate and

the topology information are shown in Table 6.2.16 and Table 6.2.17 respectively.

Table 6.2.16 The coordination of the nodes. (units: μm)

Node NO.	X direction	Y direction	Node NO.	X direction	Y direction
1	0.0	0.0	4	1.0	1.0
2	1.0	0.0	6	1.0	0.0
3	0.0	1.0	6	0.0	1.0

Table 6.2.17 The topology information of the element.

	Element NO.	Topology Information
Grain 1	1	1, 3, 2
Grain 2	2	4, 6, 6
Grain Boundary	3	3, 6, 6, 2

The boundary condition is imposed on the bottom-line nodes to the Y direction and the left line to the X direction are always zero respectively. The node constraint information is listed in Table 6.2.18.

Table 6.2.18 The constrained node of boundary condition.

The Constraint Direction	Node Number
X direction	1, 3
Y direction	1, 2

The equivalent node loading factor information is shown in Table 6.2.19.

Table 6.2.19 The equivalent node loading factor information.

Node Number	Loading Factor	
	X direction	Y direction
4	0.0	0.6
6	0.0	0.6

➤ *Result and Discussion.*

The simulation results should be compared with the theoretical result. In this case study, the theoretical result of the grain part is: the stress at Y direction is 40MPa and the stress in the other two directions (X direction and shear direction) should be zero. The theoretical result of the grain boundary part is: the stress at normal direction is 20MPa and the stress in the separated direction should be zero. The simulation results for each element are shown in Table 6.2.20. According to this table, the accuracy of the Goodman element with angle has been validated.

Table 6.2.20 The simulation stress for each element. Reproduced from [45]

Grain (Unit: MPa)			
Element No.	X Direction	Y Direction	Shear Direction
1	-3.663×10^{-16}	40.000	0.000
2	0.000	40.000	2.086×10^{-16}
Grain Boundary (Unit: MPa)			
Element No.	X Direction	Y Direction	
3	-20.000	20.000	

Validation of Non-linear Creep Part

6.2.5 Validation of the Non-linear Creep Part with quadrilateral element

The information of the FE model, boundary condition, loading, and elastic parameters have been mentioned in section 6.2.2. Therefore, the above information is not repeated here, only the benchmark of the non-linear iteration part is presented. The accuracy of the creep solution is related to the integral of the creep constitutive equation and the elastic stress field of each iteration step. Thus, the benchmark focuses on two parts: the convergence of stress updating and the integral accuracy of constitutive equations.

➤ *The Validation of the Stress Update*

The Bi-grains case study is a kind of uniform condition and the deformation of each element is the same, hence there is no effect of stress redistribution. The sign of the non-linear iterating convergence in the elastic stress of each element is the same. In this case, the non-linear iteration lasts for 14166 steps with the time step of 0.000001 (normalized). In order to visualize the convergence of the stress field, the stress condition of two iterative steps (60000 and 141664) are selected randomly during the process and shown in Table 6.2.21.

Table 6.2.21 The elastic stress field of each element at selected step.

Iteration Step		Element No.	Direction	Elastic Stress (Unit: MPa)
60000	Grain2	1	X	-2.686×10^{-9}
			Y	20.000
			τ	9.387×10^{-10}
		2	X	-3.348×10^{-9}
			Y	19.999
			τ	-1.317×10^{-9}
	Grain1	3	X	-9.670×10^{-10}
			Y	20.000
			τ	8.767×10^{-10}
		4	X	2.489×10^{-9}
			Y	20.000
			τ	-4.068×10^{-10}
	GB1	6	X	-1.006×10^{-9}
			Y	19.999
		6	X	0.000
Y			20.000	
141664	Grain 2	1	X	1.718×10^{-10}
			Y	20.000
			τ	1.026×10^{-10}
		2	X	-2.869×10^{-12}
			Y	20.000
			τ	9.994×10^{-12}

	Grain1	3	X	-9.664×10^{-11}
			Y	19.999
			τ	1.226×10^{-10}
		4	X	2.762×10^{-11}
			Y	20.000
			τ	-3.423×10^{-11}
	GB1	6	X	1.332×10^{-10}
			Y	19.999
		6	X	0.000
			Y	20.000

According to the observation of the data shown in Table 6.2.21, the result shows good agreement with the theoretical result that the applied stress of each element keeps constant during the non-linear iteration period. For the grain part, the maximum final cumulative error in the x direction is 1.718×10^{-10} MPa (NO.1 element), in shear direction is 9.994×10^{-9} MPa (NO.2 element). For the grain boundary part, the maximum final cumulative error in separate direction is 1.332×10^{-10} MPa (NO.6 element) and in normal direction is 0.866×10^{-8} MPa(NO.6 element). In summary, the in-house procedure has shown a good convergence at the preliminary Bi-grains case study.

➤ *The Validation of the integration.*

The set of the constitutive equation for describing the creep evolution of grain boundary has been mentioned before. The creep deprecating state is determined by three key variables: q (the cavity density), β (the damage variable), and a (the average radius of cavities), and the state includes the failure degree and the creep deformation of grain boundaries. Therefore, these three parameters are treated as indicators of the benchmark.

This case simulates the micro creep evolution of copper at 600°C, the grain part is modeled by power-law creep and the grain boundary part is modeled by Vöse's equations. The parameters of these constitutive equations are shown in Table 6.2.22.

In Figure 6.2.5, it shows the three key indicators versus the normalized time. Based on the observation of the results, the rupture time is 0.142 (normalized), the value of the peak point in the β curve is 0.162, and the q is stopped at 0.011. The evolution of cavity density is shown in Figure 6.2.5 (a) and it is clear that the density of cavity increases at the initial stage. When the upper limit 0.017 is reached, it starts to decrease and stops at 0.097 finally. The main reason causes

Table 6.2.22 The material parameters of copper at 600°C [68,82].

Grain	A	$38.8\text{MPa}^{-n}\text{S}^{-m-1}$
	Q	$197\text{KJ} \cdot \text{mol}^{-1}$
	n	4.8
	m	0
GB	$\sqrt[3]{\bar{D}_{\text{gb}}}$	3.9696
	$\bar{\gamma}_s$	0.089
	$\bar{\alpha}_p$	0.24
	\bar{R}	42

cavity density to decrease if the cavity coalescence rate is over the cavity nucleation rate when the cavity density reaches the upper limit. In this constitutive model, it defines that cavity coalescence occurs when the distance is less than the critical value (here are 0.1 times of the initial cavity radius). Although the cavity density is reduced in the final stage, the average radius of the cavities is increasing monotonically, as shown in Figure 6.2.5(c). Therefore, the total area of the cavities increases continuously. The macroscopic phenomenon is that the deformation and damage value (as shown in Figure 6.2.5(b)) of grain boundaries are increasing continuously.

➤ *Error Analysis*

The creep evolution of these three indicators of the constitutive equations has been given. However, it is only mathematical integral and does not involve the FE algorithm. Due to this bi-grains case study belonging to a special

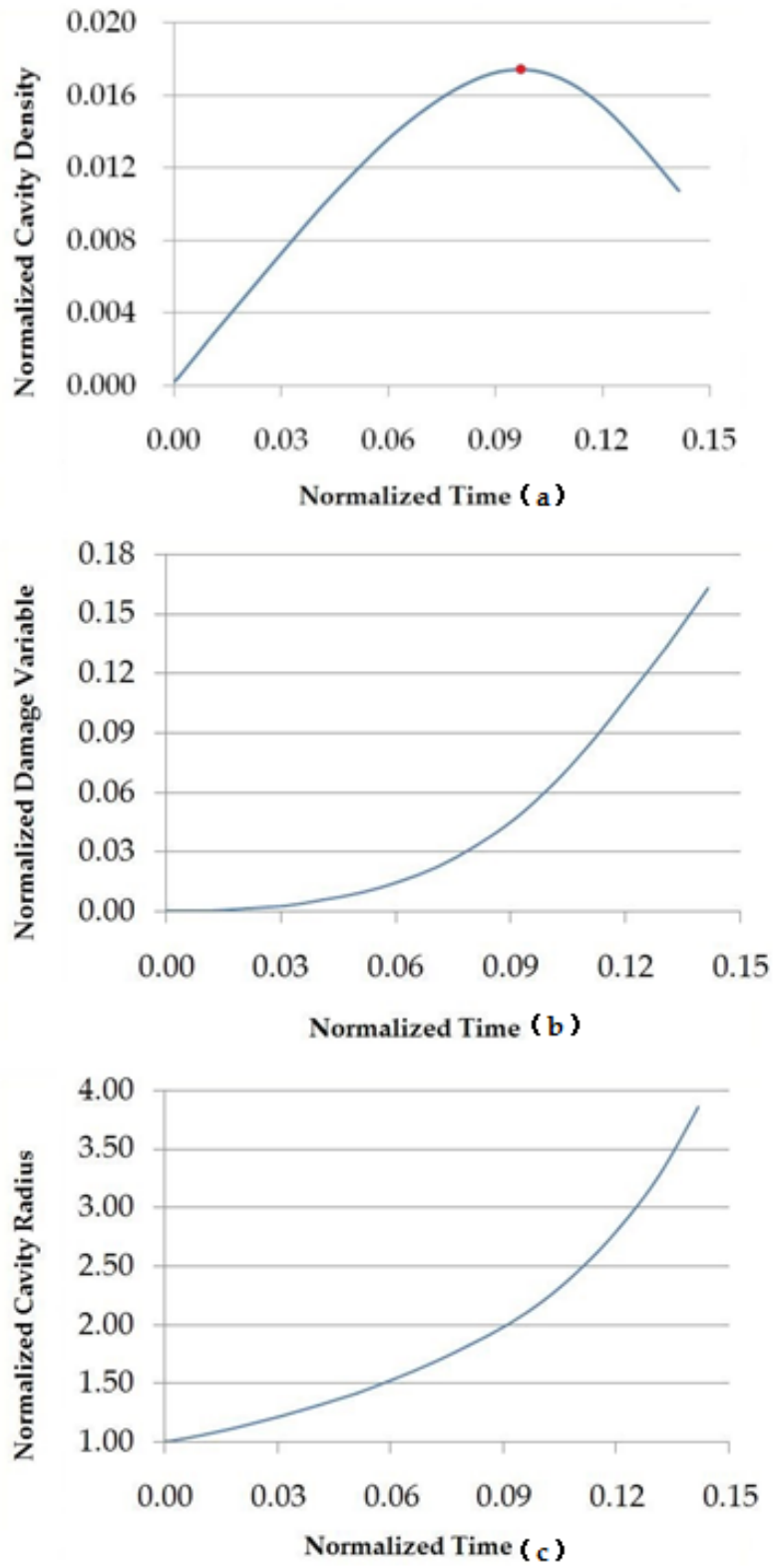


Figure 6.2.5 Three indicators of the creep evolution. (a) the change of normalized cavity density versus the normalized time. (b) the change of normalized damage versus the normalized time. (c) the change of the average cavity radius versus the normalized time.

Table 6.2.23 The percentage error of curves between simulation and reading value. Reproduced from [45]

Name	Reading Value	Simulation Value	Percentage Error
Rupture Time(normalized)	0.15	0.142	6%
Maximum value of the damage variable β	0.166	0.162	1.63%
Final value of cavity density ρ (normalized)	0.010	0.011	6.96%
The change point value of the cavity density ρ (normalized)	0.017	0.017	2.11%
The time point of the cavity density change point(normalized)	0.096	0.097	1.8%

uni-axial loading condition, therefore, it is still meaningful to use the published curve as the benchmark result. The exact value of the curve is not given in the publication, so the simulation results can only be compared with the reading value from curves and the percentage error is shown in Table 6.2.23.

6.2.6 Validation of the Non-linear Creep Part with triangle element

The validation of bi-grains case with triangle element is presented in this section, similar to the previous section 6.2.5, the convergence of the non-linear iteration and the integral of the constitutive equations need to be verified.

➤ *The Validation of the Stress Update*

The setting of the bi-grain FE model under two different meshing (quadrilateral element type and triangle element type) is consistent. It includes the same loading, boundary condition, constitutive equation, material constants and integral time step size. In this case, the non-linear iteration lasted 141664 steps. The elastic stress fields in two randomly selected steps NO.6000 and NO.141664 are shown in Table 6.2.24.

Table 6.2.24 The elastic stress field of each element at selected steps with triangle element.

Iteration Step		Element No.	Direction	Elastic Stress (Unit: MPa)	Theoretical Value (Unit: MPa)
60000	Grain2	1	X	1.332×10^{-13}	0
			Y	20.000	20
			τ	-2.606×10^{-13}	0
		2	X	-3.180×10^{-13}	0
			Y	20.000	20
			τ	1.249×10^{-13}	0
		3	X	4.974×10^{-14}	0
			Y	20.000	20
			τ	2.264×10^{-13}	0
		4	X	-2.682×10^{-13}	0
			Y	20.000	20
			τ	-8.268×10^{-14}	0
	Grain1	6	X	-7.248×10^{-13}	0
			Y	20.000	20
			τ	-1.981×10^{-13}	0
		6	X	6.009×10^{-13}	0
			Y	20.000	20
			τ	-2.716×10^{-13}	0
		7	X	1.699×10^{-14}	0
			Y	20.000	20
			τ	-1.779×10^{-13}	0
		8	X	-6.413×10^{-13}	0
			Y	20.000	20
			τ	-4.097×10^{-14}	0
GB1	9	X	-3.903×10^{-13}	0	
		Y	20.000	20	
141664	Grain2	1	X	-2.461×10^{-12}	0
			Y	20.000	20
			τ	-6.787×10^{-12}	0
		2	X	1.791×10^{-11}	0
			Y	20.000	20
			τ	7.969×10^{-11}	0
		3	X	-2.467×10^{-11}	0
			Y	20.000	20
			τ	-1.198×10^{-11}	0
		4	X	-6.238×10^{-11}	0

			Y	20.000	20	
			τ	-1.062×10^{-10}	0	
			X	-9.640×10^{-12}	0	
	Grain1	6		Y	20.000	20
				τ	-2.869×10^{-12}	0
				X	1.111×10^{-11}	0
		6		Y	20.000	20
				τ	-1.439×10^{-14}	0
				X	1.671×10^{-11}	0
		7		Y	20.000	20
				τ	7.147×10^{-13}	0
				X	1.648×10^{-11}	0
		8		Y	20.000	20
				τ	-2.069×10^{-12}	0
				X	-8.327×10^{-11}	0
GB1		Y	20.000	20		

According to the result shown in Table 6.2.24, it is clear that the in-house procedure with the triangle element has been shown in good convergence in the non-linear iteration.

➤ *The Validation of the integration.*

Since this is a uniform loading case, and the applied external load, material parameters, and time steps are consistent with the previous case, the integral part does not need to be described here again.

6.2.7 Validation of the Non-linear Creep Part of FE model with Angle

The previous two cases are special conditions, which is deformation without the sliding part. Thus, the GB with angle is introduced. In order to make the benchmark logical and efficient, the normal stress is controlled to be consistent with these previous two cases, is 20 MPa. The information of the FE model, boundary condition, loading, and elastic parameters have been

mentioned in section 6.2.4.

➤ *The Validation of the Stress Update*

Similar to the present method in section 6.2.5 and 6.2.6, randomly selected two iteration steps, here are No.100000 and No.141664, to check the convergence of the stress field, as shown in Table 6.2.25. It is clear that the in-house procedure under angle conditions has been shown in good convergence in the non-linear iteration.

➤ *The Validation of the sliding part.*

The constitutive model of the sliding part has been mentioned before. In this case study, the normalized magnitude of the sliding viscosity coefficient $\bar{\eta}_{\text{sliding}} = 0.062$, (the original value is $\eta_{\text{sliding}} = 3.86 \times 10^7 \text{Ns/mm}^3$). The sliding model is linear deformation, in which the sliding rate is positively related to the stress in the separated direction. The normalized theoretical sliding amount is 2.721 and the normalized simulating sliding amount is 2.722, compared these two values, it is clear that the result has shown good agreement (the percentage error is 0.037%).

Table 6.2.25 The elastic stress field at the selected iteration steps. Reproduced from [45]

Iteration step		Element No.	Direction	Elastic Stress Field (Unit: MPa)	Theoretical Value (Unit: MPa)
100000	Grain 1	1	X	-9.623×10^{-9}	0
			Y	40.000	40
			τ	0.000	0
	Grain 2	2	X	-6.369×10^{-10}	0
			Y	40.000	40
			τ	-20.000	0
	GB 1	3	X	-20.000	-20
			Y	20.000	20
	141664	Grain 1	1	X	4.134×10^{-7}
Y				40.000	40
τ				0.000	0
Grain 2		2	X	1.317×10^{-8}	0
			Y	40.000	40
			τ	-1.349×10^{-8}	0
GB 1		3	X	-20.000	-20
			Y	20.000	20

6.2.8 Conclusion

Based on the benchmark of these three conditions, it demonstrates the numerical accuracy and stability of the procedure under the simple stress condition. For the non-linear iteration part, it shows the good convergence during the stress updating. In summary, the primary verification has paved the way for the subsequent polycrystalline case study.

6.3 The Application of the In-house Procedure via the Numerical Investigation of the Polycrystal Case Study.

In this case, it simulates the creep evolution of copper-antimony alloy at GB level, the temperature is 823K. For the GB part, the cavity model of GB has been mentioned before, and the material constants have been mentioned in Table 6.2.22. The initial cavity density $\rho_0 = 10^{-3} \text{mm}^{-2}$. The Newtonian viscous flow is adopted to describe the sliding deformation and the sliding viscosity $\eta_{\text{sliding}} = 3.86 \times 10^7 \text{Ns/mm}^3$ [14]. For the grain part, it is modeled by the power-law constitutive equations, and the parameters have been mentioned in Table 6.2.22.

6.3.1 FE Model Generation

The FE model has been generated by the Neper package [39], the structure is built by the Tessellation Module (-T) of Neper and in a rectangular domain with 1mm length and 1mm width (the instruction is '-square (1,1)'). The initial seed positions identifier variable is set 1 (the instruction is '-id 1'). This model contains 20 grains and 60 grain boundaries (the instruction is '-n 20'). The mesh of the structure and re-mesh to generate the GB is by the Meshing Module (-M) of Neper. The grains are meshed by 909 triangle plane strain elements (the instruction is '-eltype tri') and the grain boundaries are meshed by 162 Goodman elements, as depicted in Figure 6.3.1. (GB is marked by the red line), and the orientation angle of the total Goodman elements is shown in Figure 6.3.2. The total instruction to generate this FE model is shown in Table

6.3.1. A uniformly distributed loading 10MPa is applied on the top surface in the Y direction, the boundary condition of this case study is: the left line and the bottom line of the domain are fixed on the X direction and the Y direction respectively.

Table 6.3.1 The instruction for generate the FE model.

Module	Instruction
Tessellation Module	\$ neper -T -n 20 -id 1 -reg
Meshing Module (Mesh and Re-mesh)	\$ neper -M -n20-id 1.tess - interface continuous

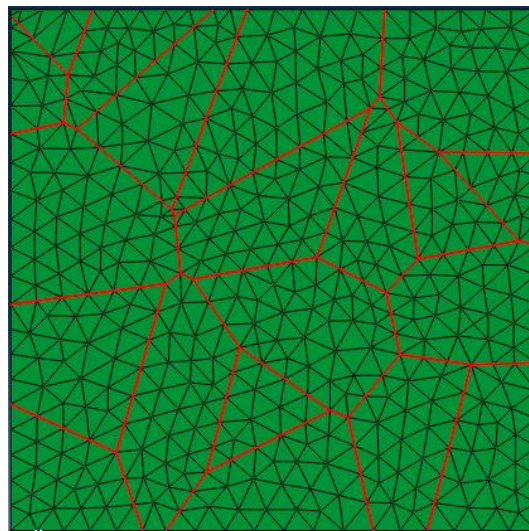


Figure 6.3.1 The FE model for the polycrystalline case study.
Reproduced from [45]

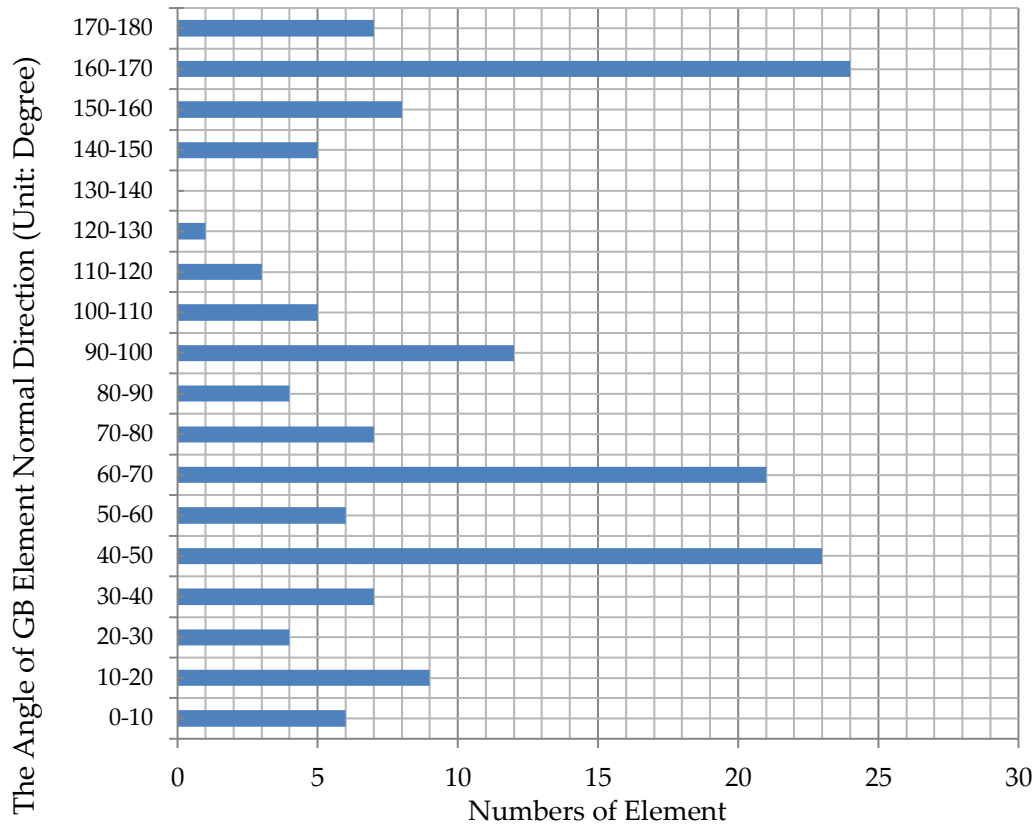


Figure 6.3.2 Orientation angle of the grain boundary element's normal direction.
 Orientation angle of the grain boundary element's normal direction.
 Reproduced from [45]

6.3.2 Failed element, Stress Field and Creep Damage Evolution

The sequence evolution of failed element.

The total simulating time is 78.9h and there are seven Goodman elements that failed. The failed sequence of seven positions is shown in Figure 6.3.3 and the times have been listed in Table 6.3.2.

Table 6.3.2 The element details of sequence at seven boundary elements. Reproduced from [45]

Position	Element NO.	Angle	Time (Unit: hour)	Step
A	48	66.3	23.66	12003387
B	122	76.2	66.66	33246192
C	93	64.0	68.48	34728834
D	94	64.0	68.48	34728866
E	111	66.4	70.69	36848660
F	87	146.3	78.90	39987606
G	88	146.3	78.90	39987617

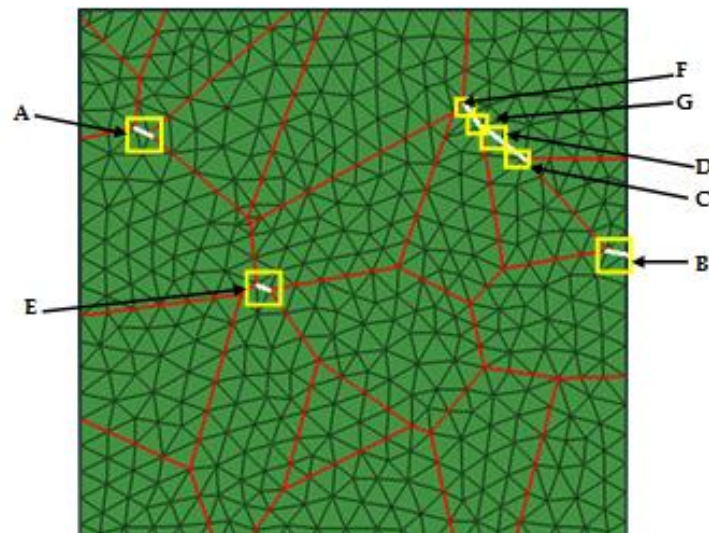
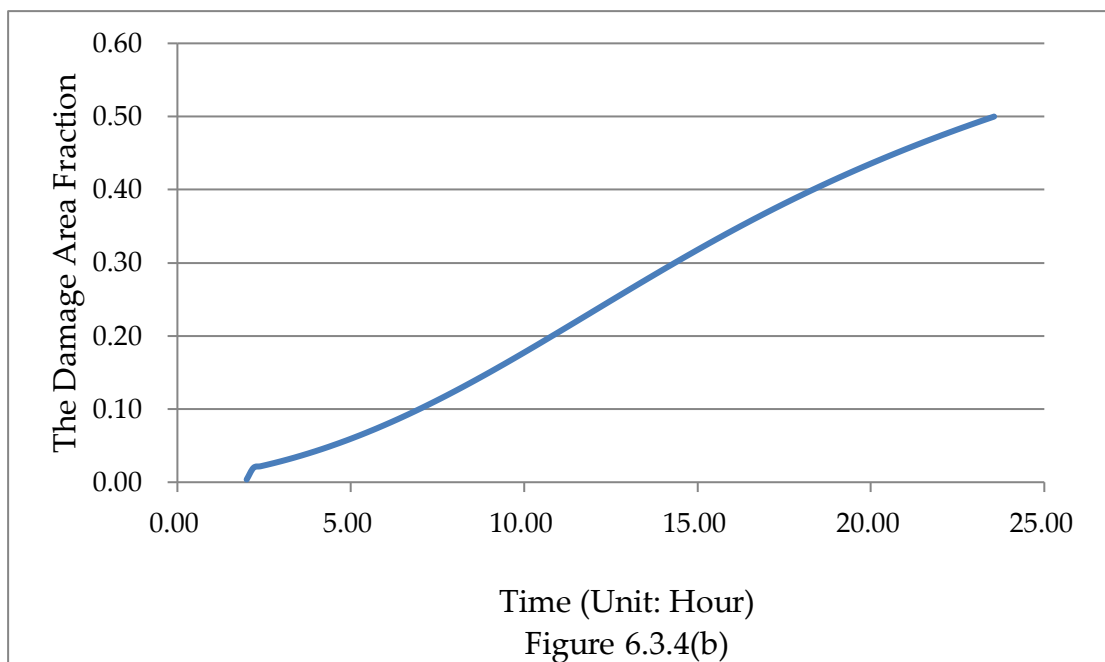
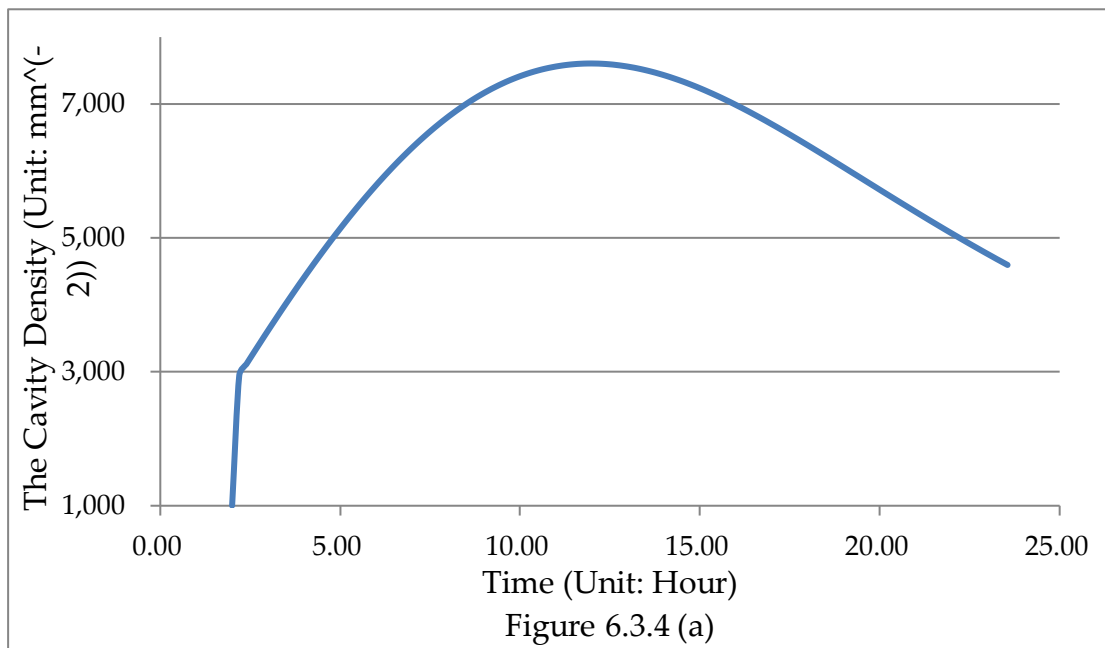


Figure 6.3.3 The sequence evolution of the failure elements. Reproduced from [45]
 The first failure occurs at the position A, the evolution of this position, which include the cavity density, the damage variable, the damage area fraction, and the average radius of the cavities all of which are shown in Figure 6.3.4. The

creep evolutions of seven positions (A, B, C, D, E, F, G), which include these aspects: the stress at normal direction, the cavity nucleation ratio, the evolution of the cavity density, the cavity coalescence rate, and damage area fraction are shown in Figure 6.3.6. The normal stress and damage evolution of all GB elements at different sampling time point are shown in Figure 6.3.6 and Figure 6.3.7, respectively.



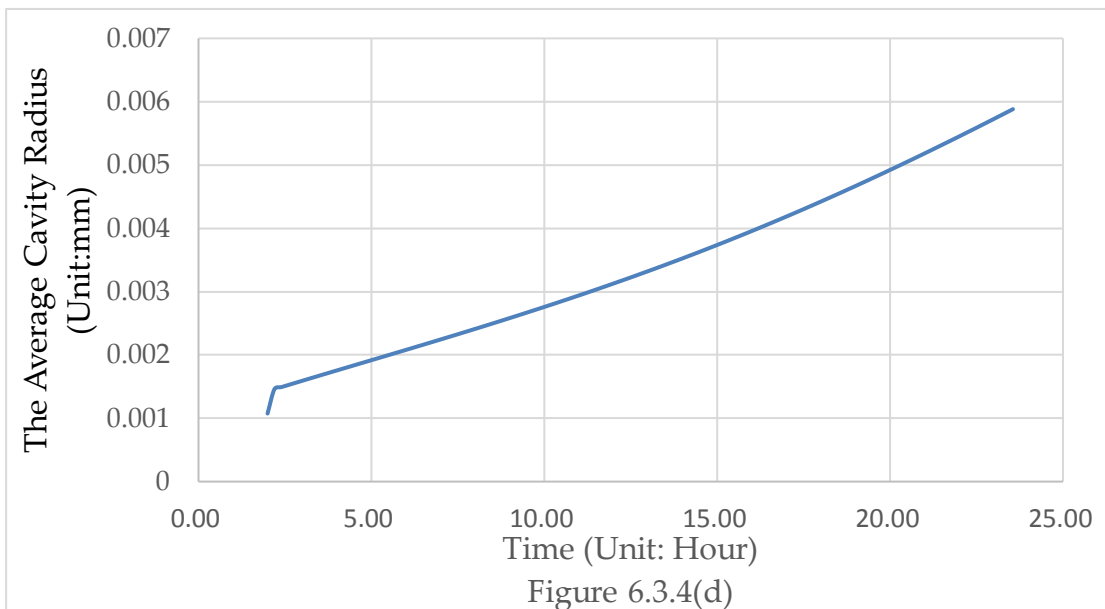
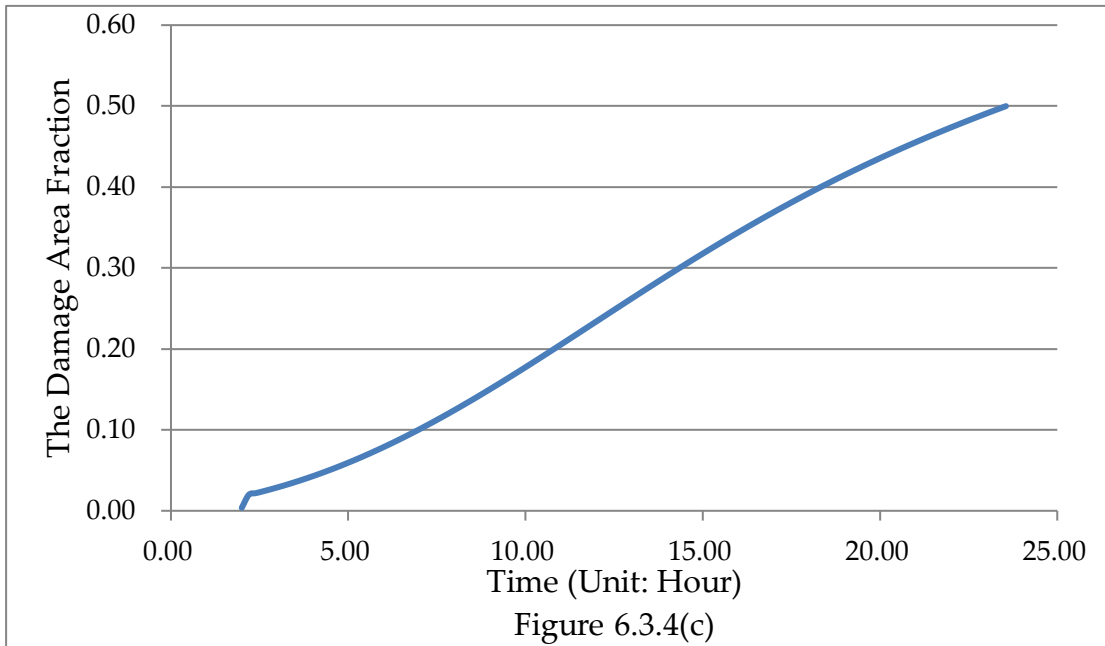


Figure 6.3.4 The creep evolution of Position A. (a) the cavity density versus the time; (b) the damage variable versus the time; (c) the damage area fraction versus the time; (d) the average radius of cavities versus the time. Reproduced from [45]

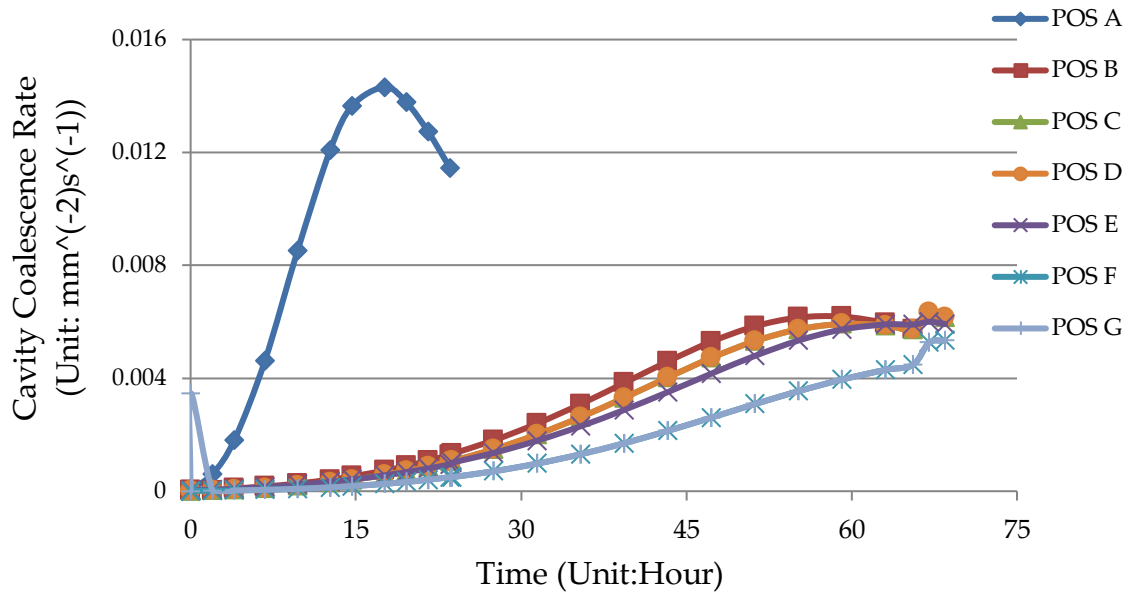


Figure 6.3.5(a)

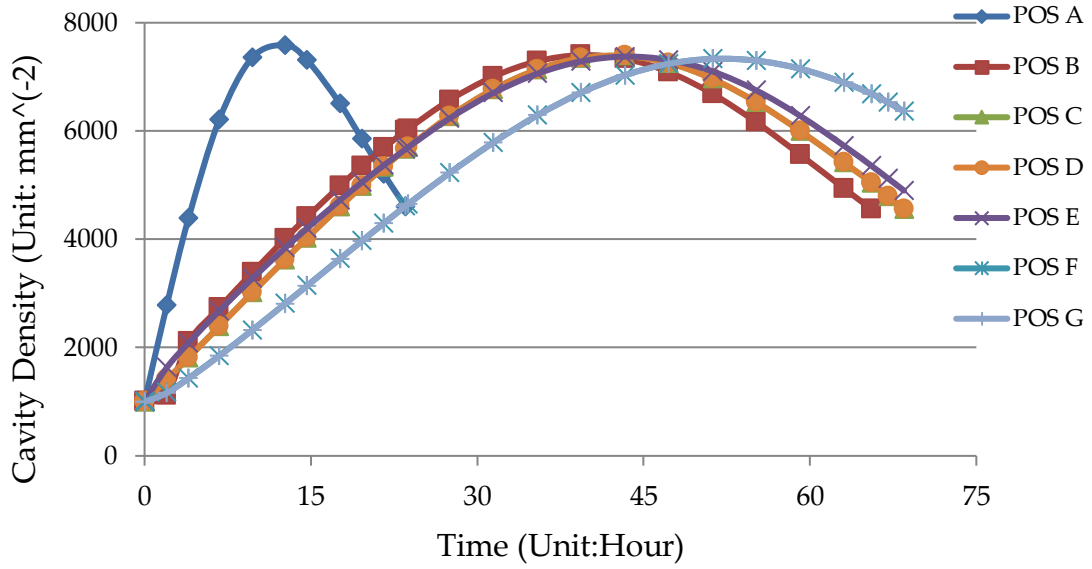


Figure 6.3.5 (b)

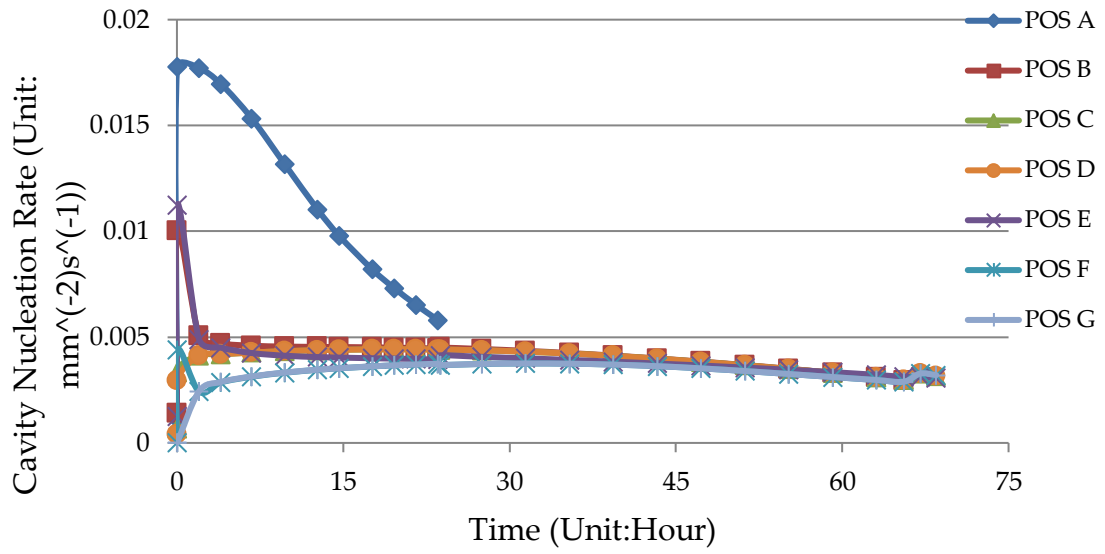


Figure 6.3.5 (c)

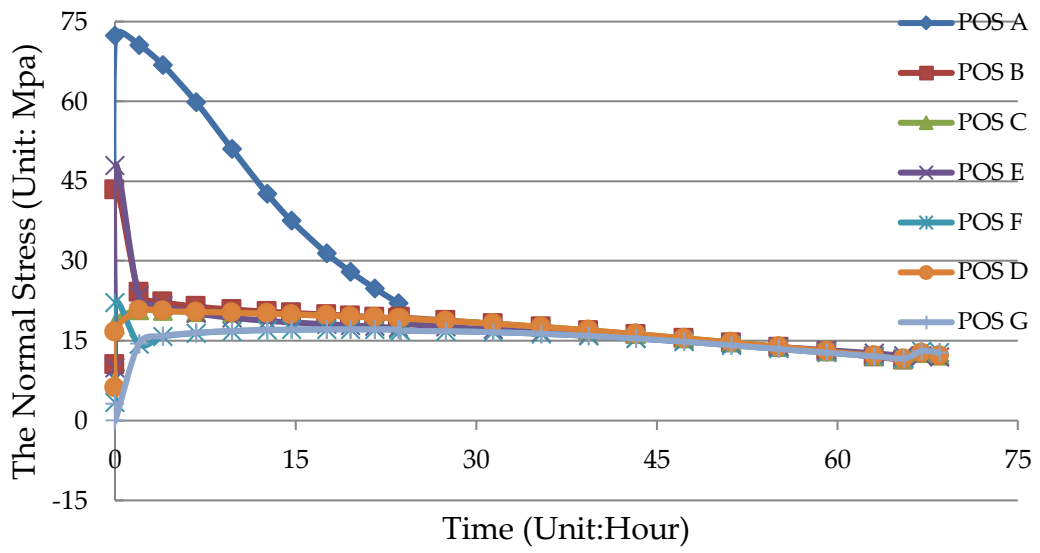


Figure 6.3.5(d)

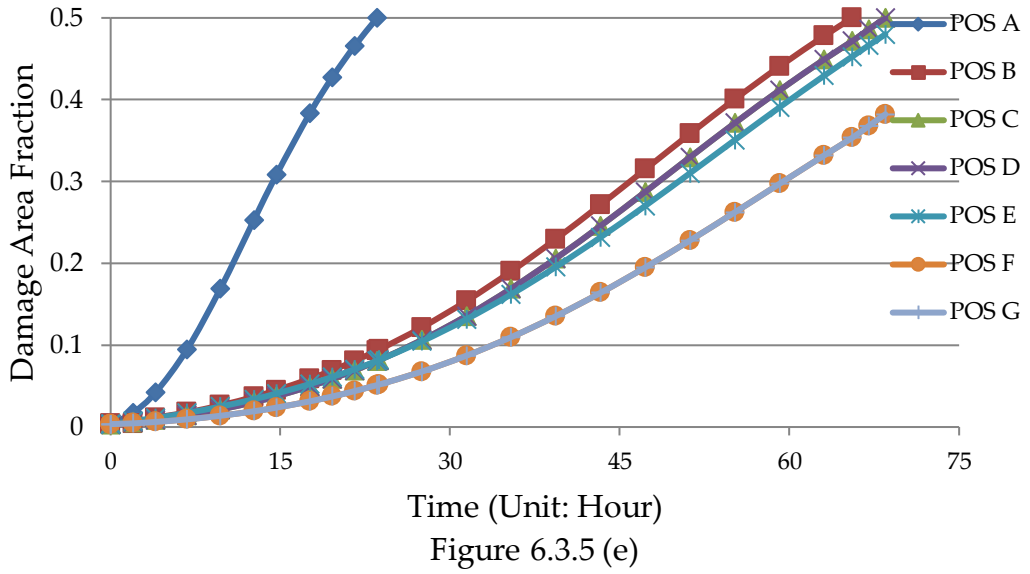
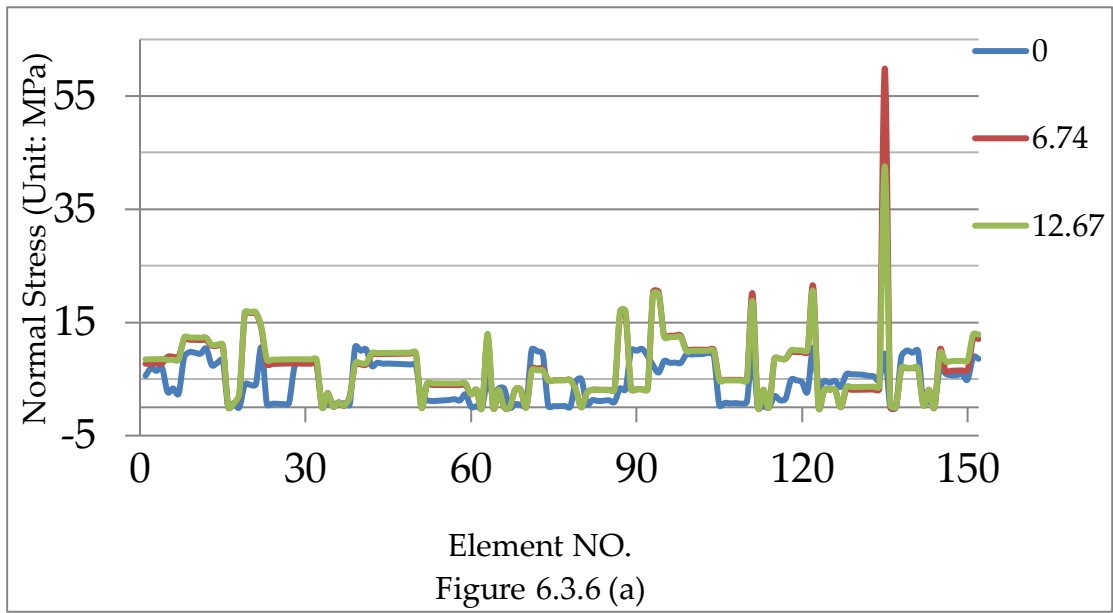
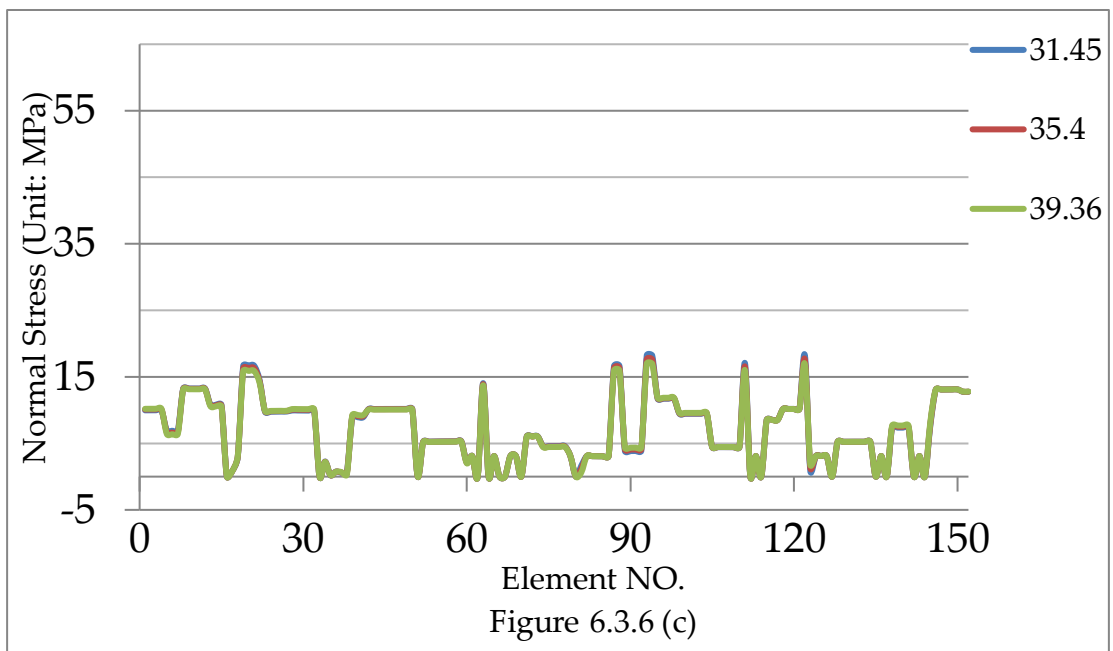
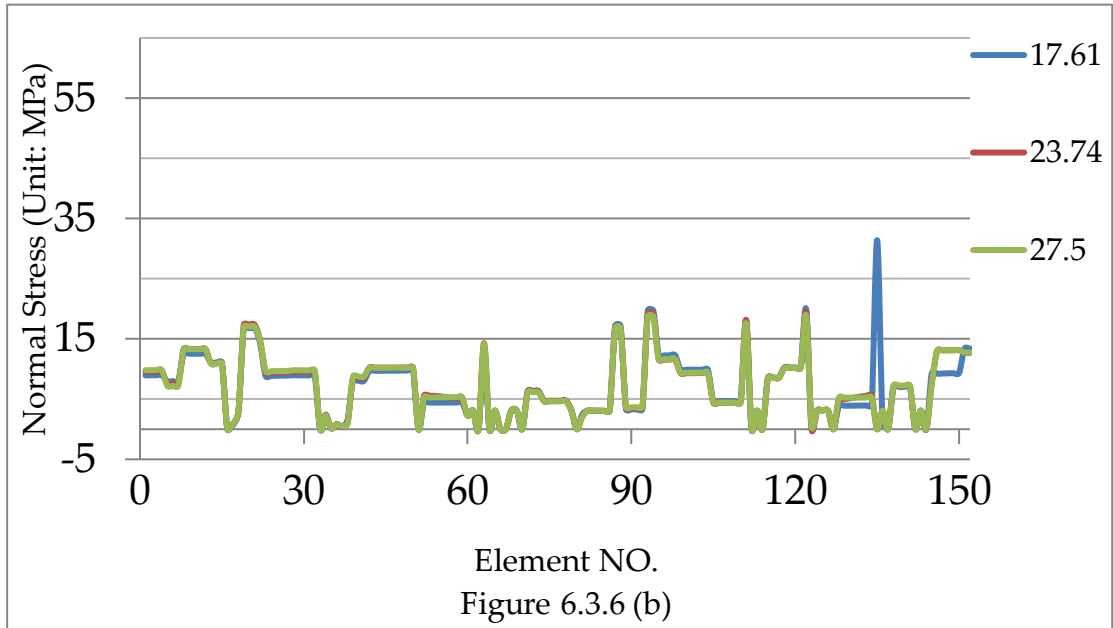
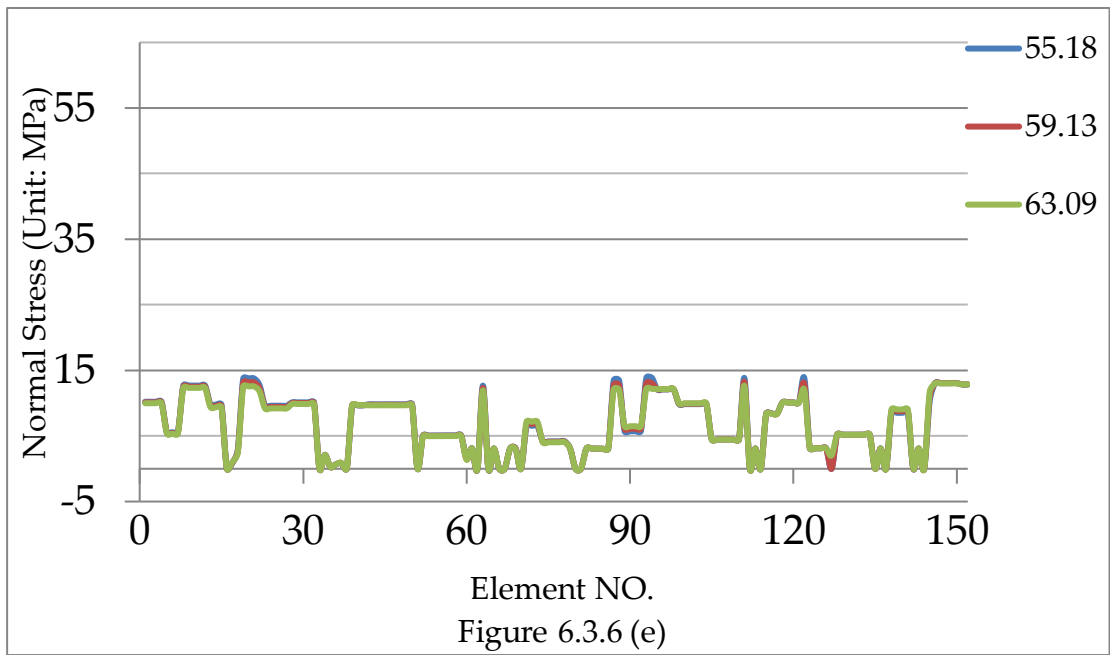
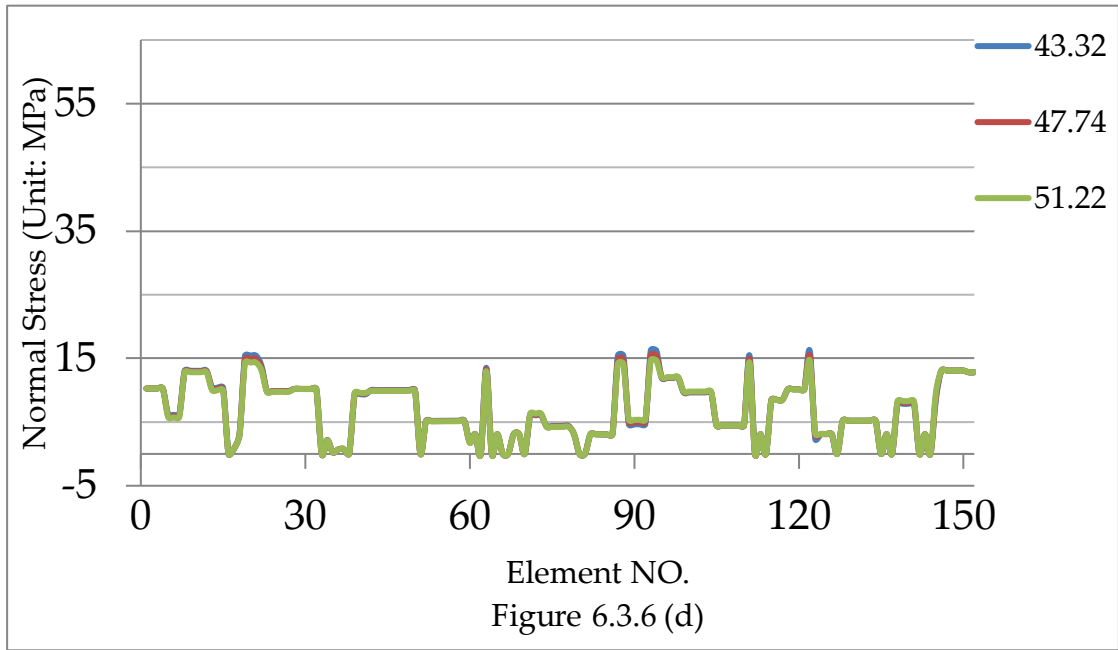


Figure 6.3.5 The creep evolution of seven failed elements. (a) the cavity coalescence Rate versus the time; (b) the cavity density versus the time; (c) the cavity nucleation rate versus the time; (d) the normal stress versus the time; (e) the damage area fraction versus time.







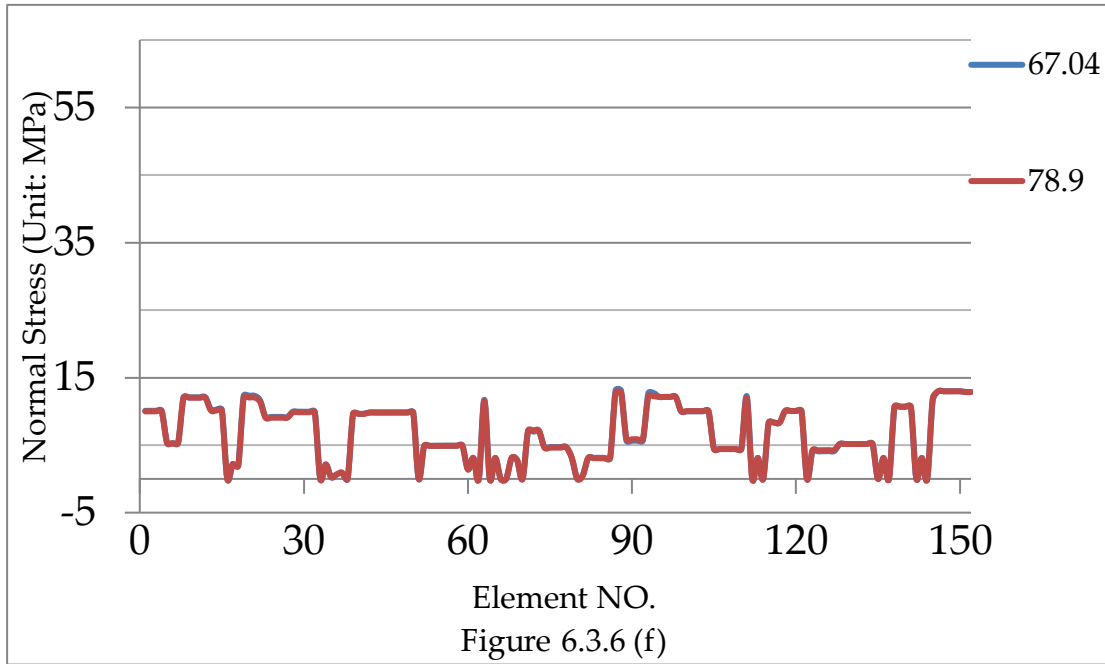


Figure 6.3.6 The normal stress evolution of all Goodman elements. (a)Time Point: 0h, 6.74h, 12.67h; (b)Time Point: 17.61h, 23.74h, 27.6h; (c)Time Point: 31.46h, 36.4h, 39.36h; (d)Time Point: 43.32h, 47.27h, 51.22h; (e)Time Point: 55.18h, 59.13h,63.09h; (f)Time Pont:67.04h, 78.9h.

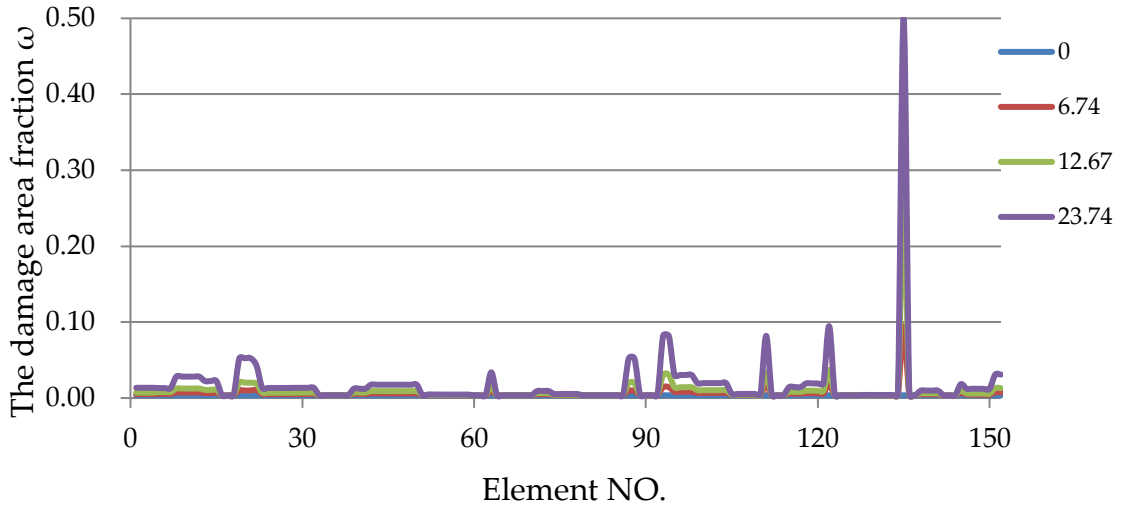


Figure 6.3.7 (a)

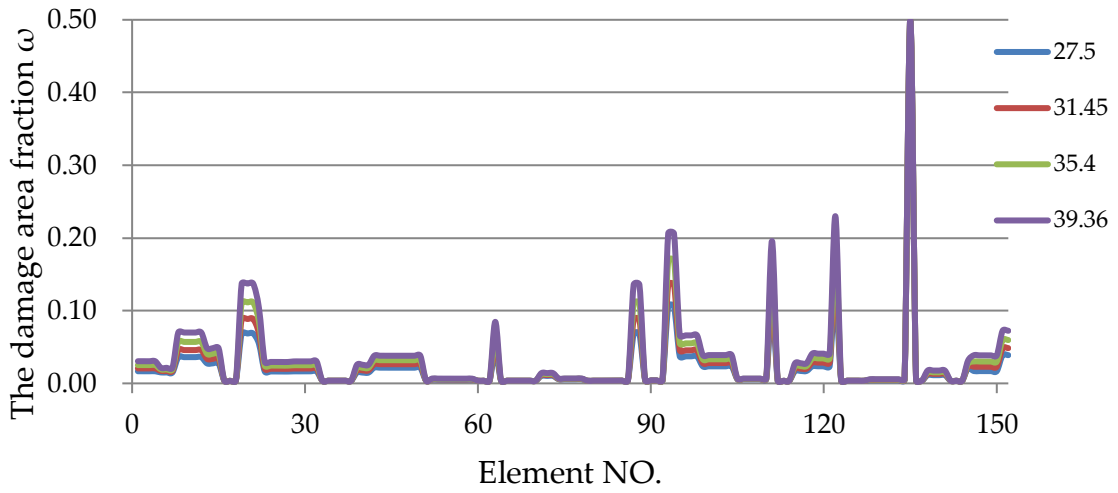


Figure 6.3.7 (b)

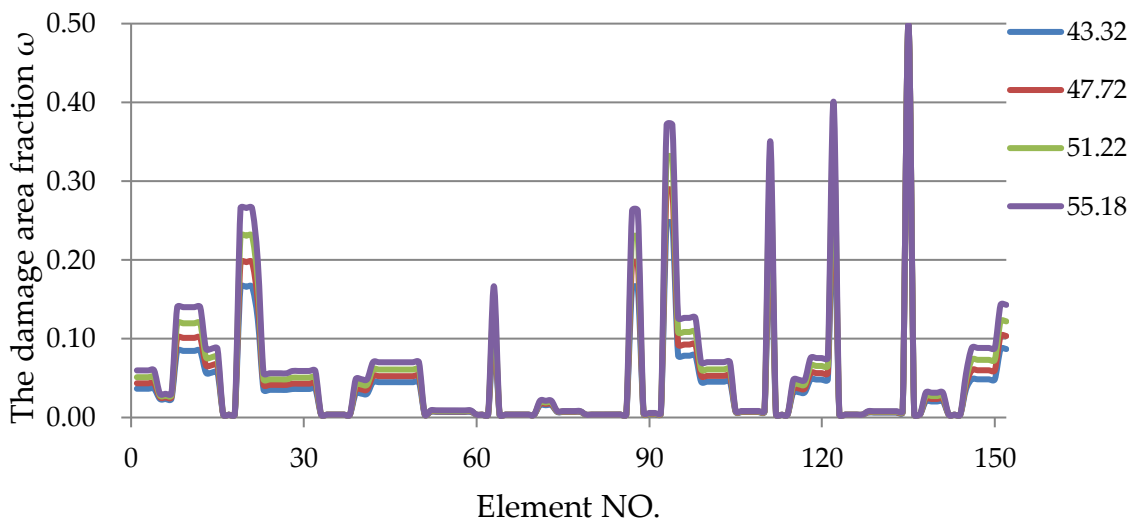


Figure 6.3.7(c)

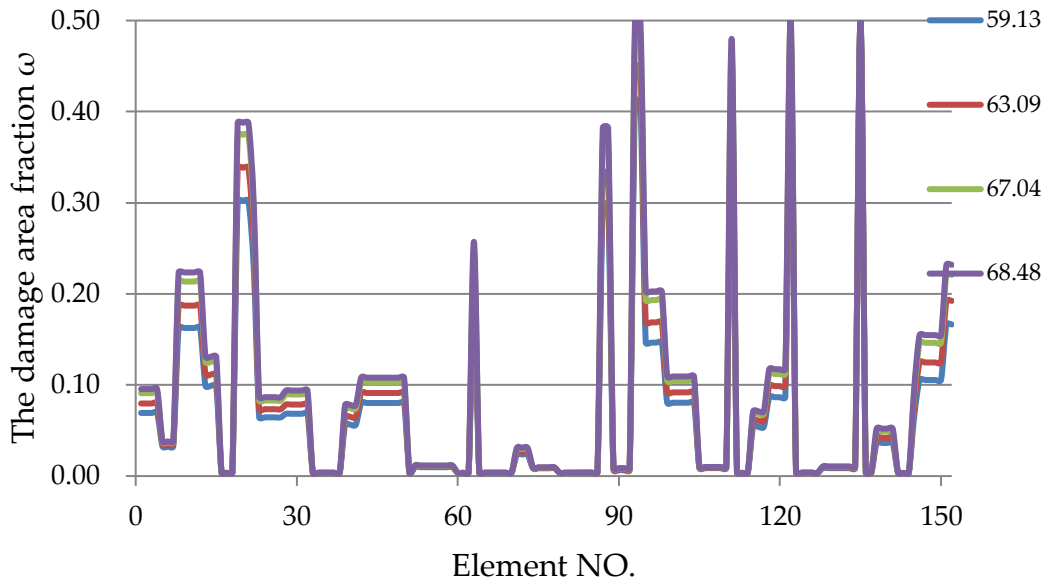


Figure 6.3.7(d)

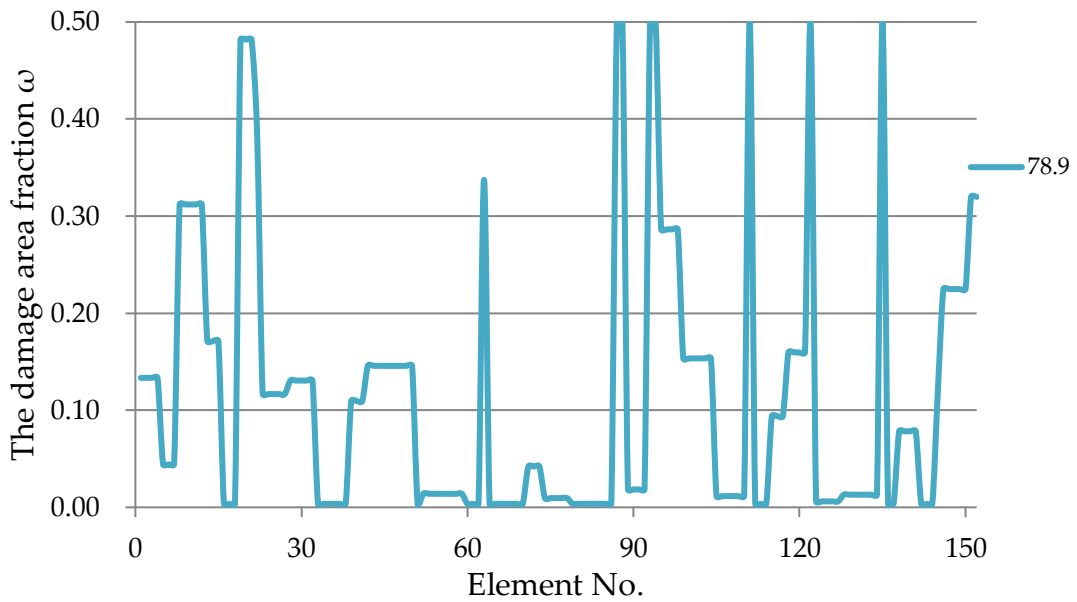


Figure 6.3.7(e)

Figure 6.3.7 The damage evolution of all Goodman elements at the same time point; (a)Time Point: 0h, 6.74h, 12.67h, 23.74h; (b)Time Point: 27.6h, 31.46h, 36.4h, 39.36h; (c)Time Point: 43.32h, 47.27h, 61.22h, 66.18h; (d)Time Point: 69.13h, 63.09h, 67.04h, 68.48h; (e)Time Point: 78.9h. Reproduced from [45]

6.3.3 Discussion

The discussion comes from the publications and done by Q.Xu [45].

According to Figure 6.3.6 (d), it can be found that the highest point of normal stress occurs at position A and it remains relatively high until failure.

Compared with position A, the normal stress of other grain boundaries is lower than it but still higher than the applied stress (10 MPa). Hence, the elements could be subject to higher normal stress, which lead to a higher nucleation rate and growth rate of cavities, this is the main reason for GB failure.

From Figure 6.3.6 (a-f), it presents the stress redistribution that happens during the non-linear creep iteration, which reveals the uneven GB sliding and normal jumping deformation play the main contribution to it; the further stress redistribution will be caused by due the increasing of creep deformation, resulting in a local high-stress reduction. Therefore, the damage evolution occurs in such a way as shown in Figure 6.3.6(e).

From Figure 6.3.6 (a-f) and Figure 6.3.7 (a-e), it clearly reveals that the stress concentration happens on the grain boundary initially, due to the creep evolution and stress redistribution, the concentrated stress was relaxed. However, after failure occurs, the stress of the failed element decreases sharply to zero and has no loading capability. In addition, the grain boundaries with some slant degree have high damage value, however, damage of the elements parallel to the direction of the applied stress is relatively lower. It may present that the deformation in normal direction is greater than the sliding direction and the GB sliding affects the stress redistribution as well. Based on the creep damage evolution of all elements, it can be clearly concluded that the fracture of a component is determined by a

reasonable proportion of failed grain boundaries, and the evolution of creep damage is determined by the mesh size, the GB orientation, and the grain shape are determined together [45].

Chapter 7 Development and Validation of the 3-dimensional Procedure at Grain Boundary Level

7.1 Introduction

The three stress states (plane stress, plane strain, and axisymmetric) in the two-dimension version are idealizing the three-dimensional states at different conditions, hence the more general three-dimensional version has its practical importance and significance. Compared with two-dimensional, the standard 8-nodal 3D hexahedron element is used for FE modeling in the grain part, and the 8-nodal 3D Goodman element is used for modeling in the grain boundary part. The implementation of the two-dimensional Goodman element is simple, in which the element has a lower degree of freedom and the integration can be obtained by algebraic calculation. However, the implementation of the three-dimensional Goodman element is more complicated. It involves creating a new local coordinate for integration and connecting global coordinates, and the element has a higher degree of freedom. In this chapter, the technical details of the implementation of the three-dimensional version are explained in detail.

The structure of this chapter is described in the following:

1. In Section 7.2, it reports the development details of the 3D version in-house procedure. Since the previous creep solver is applied in the version, the modified part is to replace the triangular element/ quadrilateral

element and 2D Goodman elements with hexagonal elements and 3D Goodman elements respectively. In section 7.2.1, it introduces the coding implementation of hexahedron for grain part, and in section 7.2.2, it introduces the coding implementation of 3D Goodman element for grain boundary part.

2. In section 7.3, it reports the validation of the grain part under elastic condition. According to this section, the accuracy and reliability of the program block for implementing the hexahedron element are verified.
3. In section 7.4, it reports the validation of the grain boundary part under elastic condition. According to this section, the developed program block and subroutines for obtaining the stiffness matrix of the 3D Goodman element are verified. The benchmark specifically includes three aspects: the accuracy of the load at normal and separated directions, the accuracy and reliability of the coordinate transmission system when the element's local coordinates and global coordinates are inconsistent, and the accuracy and reliability of the module for recovering the results at Gaussian Point.

7.2 Development of 3D Version

The core of the 3D version's development is to replace the 2D elements of the previous solver with 3D elements. In this procedure, the grain part is modeled by the 3D hexahedron element with 8 nodes, and the grain boundary part is modeled by the 3D Goodman element with 8 nodes.

7.2.1 Implementation of Grain Part

The Finite Element modeling of the grain is based on the traditional 3D hexahedron element and the mathematical background has been mentioned in section 2.6.2. The element stiffness matrix obtaining of this type is in-built in Smith's element library and program block, which can be calculated by four existing standard FE subroutines from the 'main' library [13] and the similar structure of the 2D solid element to assemble these four subroutines. The details of the relationship between the mathematical background and the subroutines are summarized in Table 7.2.1.

Table 7.2.1 List of the standard FE subroutines for the element stiffness matrix of 3D 8-node hexahedron element [13].

Name	Function
deemat	It returns the stress-strain matrix [D] at 3-Dimension condition. (EQ.2.6.21)
shape_fun	It returns the shape functions at the integration point in local coordinates. ([N] matrix, EQ.2.6.24)
shape_der	It returns the derivatives of the shape functions.
beemat	It returns the strain-displacement matrix [B].(EQ.2.6.23)

7.2.2 Implementation of Grain Boundary part

The Finite Element modeling of the grain boundary is based on the 3D eight-nodal Goodman element. The mathematical background of the element' stiffness matrix is mentioned in section 2.8 and the mathematical background of coordinate transmission is mentioned in section 3.6.2. Therefore, in this section it mainly describes the development of the program block to calculate this element's stiffness matrix.

In this program block, there are six subroutines developed: 'Gdeemat', 'TM_TD', 'T_Coord', 'G_Shape_Fun', 'G_Shaper_Der', and 'Gbeemat'. The details of these six subroutines can be found in the Appendix 1. The flowchart of this block is presented in the Figure 7.2.1.

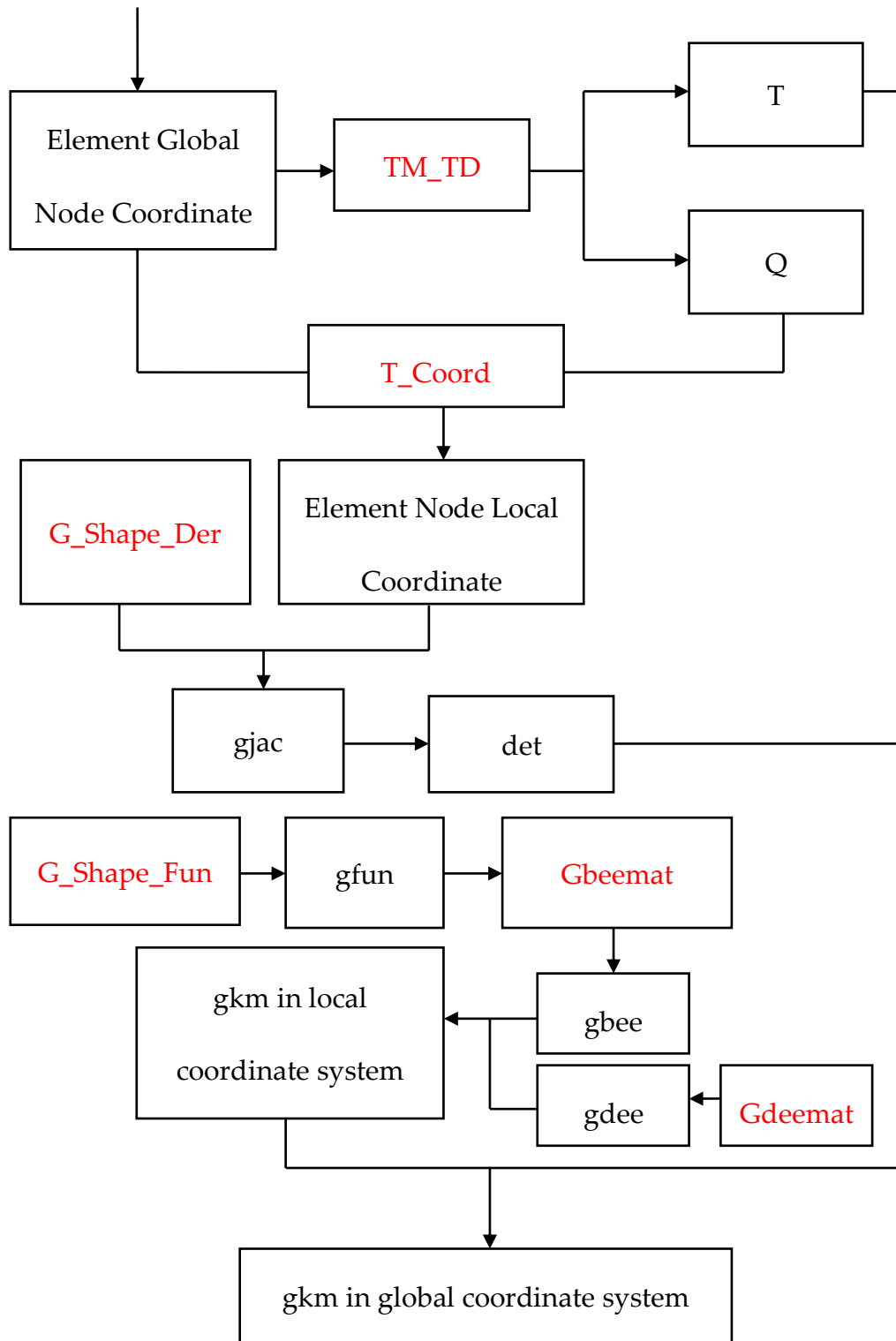


Figure 7.2.1 The flow diagram structure for obtaining the 3D Goodman element stiffness matrix. (subroutines are marked by red)

After the solution of grains' element stiffness matrix, the procedure enters the loop module of the solution of grain boundaries' element stiffness matrix, as shown in Figure 7.2.1. In the beginning, the stiffness constants which are stored in array 'cprop' is imported to the subroutine 'Gdeemat' to obtain the stiffness matrix 'D' in Equation 2.8.10, the 'D' is named 'gdee' in the procedure. The array 'gnum' and 'gcoord' store the element steering vectors and the element nodal coordinates. Based on the subroutine 'TM_TD' and imports the element nodal coordinates into it to solve out the matrix 'Q' and 'T' in Equation 2.8.21, and then applies the subroutine 'T_coord' to obtain the element nodal coordinates in local coordinates system, the mathematical background is in Equation 2.8.21. The first two columns of array 'gcoord' are restored into array 'cgcoord', which is the matrix $[\text{coord}]_{\mu\&v}$ in Equation 2.8.16. After completing the above preparations, the procedure enters the integration part. As mentioned before, the Gauss-Legendre method is employed to do the integral over the surface. For this reason, two subroutines 'G_shape_fun' and 'G_shape_der' are developed to produce the shape function array 'fun' and the derivatives of the shape functions 'gder' in M coordinates systems, respectively. Based on the derivatives 'gder', the Jacobin matrix 'gjac' and its determinates 'det' are solved. Finally, according to Equation 2.8.20, the element stiffness can be obtained.

7.3 Elastic Validation of Grain Part

7.3.1 FE Model

In this 3D version procedure, the 8-node hexahedron element is used to mesh the grain part. The benchmark of the procedure for this element is performed in this section and the FE model is shown in Figure 7.3.1.

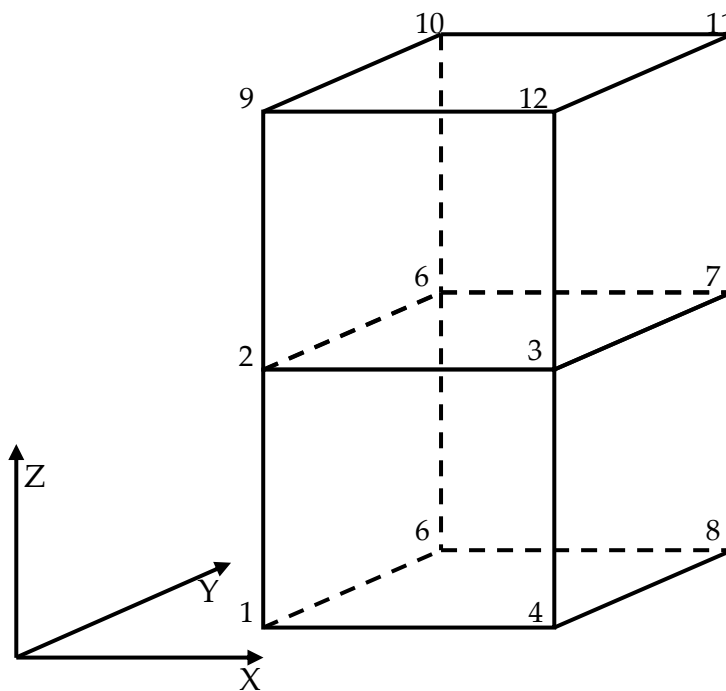


Figure 7.3.1 The schematic figure of 3D FE model.

- *The mesh, loads information, boundary condition and material parameters for this FE model*

In this model, it contains 12 nodes to generate two 8-node hexahedron elements. The coordinates of these nodes are shown in Table 7.3.1 and the topology information is shown in Table 7.3.2.

Table 7.3.1 The coordinates of the node. (unit: mm)

Node NO.	X direction	Y direction	Z direction
1	0	0	0
2	0	0	1
3	1	0	1
4	1	0	0
6	0	1	0
6	0	1	1
7	1	1	1
8	1	1	0
9	0	0	2
10	0	1	2
11	1	1	2
12	1	0	2

Table 7.3.2 The topology information of these two elements.

Element NO.	Topology Information
1	1, 2, 3, 4, 6, 6, 7, 8
2	2, 9, 12, 3, 6, 10, 11, 7

The boundary condition is imposed on the bottom surface such that the displacement components to the Z direction, the front surface such that the displacement components to the Y direction, and the left surface such that the displacement components to the X direction are always zero respectively. The

implementation of the boundary condition is through add the constraint of the node, in this case, the node constraint information is listed in Table 7.3.3.

Table 7.3.3 The constrained node of boundary condition.

The Constraint Direction	Node Number
X direction	1, 4, 6, 8
Y direction	1, 2, 3, 4, 9, 12
Z direction	1, 2, 6, 6, 9, 10

In this benchmark, it contains two sub-cases, the first case is on the top surface to the Z direction and the second one is on the right surface to the X direction.

The equivalent node loading factor information of these two cases is shown in Table 7.3.4.

Table 7.3.4 The equivalent node loading factor information.

Node NO.	X	Y	Z
The No. 1 Case			
9	0.0	0.0	0.26
10	0.0	0.0	0.26
11	0.0	0.0	0.26
12	0.0	0.0	0.26
The No. 2 Case			
3	0.6	0.0	0.0
4	0.26	0.0	0.0
7	0.6	0.0	0.0
8	0.26	0.0	0.0
11	0.26	0.0	0.0
12	0.26	0.0	0.0

7.3.2 Validation of Uni-axial Loading

In this case, the loading is applied on the top surface with 30MPa tension to the Z direction, shown in Figure 7.3.2. The theoretical stress at Gaussian Point is 30 MPa to the Z direction, and in the X, the Y, and three shear directions should be zero. According to the stress-strain constitutive relationship which has been mentioned in section 2.6.2, the strain of these six directions are:

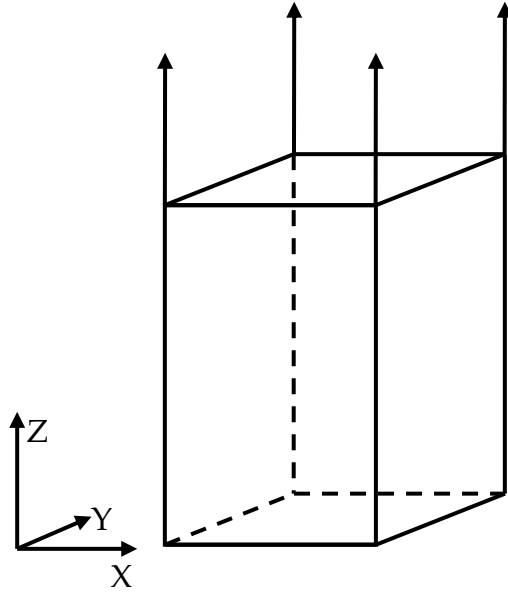


Figure 7.3.2 The schematic figure of tension model.

$$\begin{aligned}
 \varepsilon_x &= \frac{(A + B) \cdot \sigma_x - B \cdot \sigma_y - B \cdot \sigma_z}{(A - B) \cdot (A + 2 \cdot B)} \\
 \varepsilon_y &= \frac{-B \cdot \sigma_x + (A + B) \cdot \sigma_y - B \cdot \sigma_z}{(A - B) \cdot (A + 2 \cdot B)} \\
 \varepsilon_z &= \frac{-B \cdot \sigma_x - B \cdot \sigma_y + (A + B) \cdot \sigma_z}{(A - B) \cdot (A + 2 \cdot B)} \\
 \gamma_{xy} &= \frac{\tau_{xy}}{C} \\
 \gamma_{yz} &= \frac{\tau_{yz}}{C} \\
 \gamma_{zx} &= \frac{\tau_{zx}}{C}
 \end{aligned}
 \tag{7.3.1}$$

where

$$\begin{aligned}
 A &= \frac{E \cdot (1 - \nu)}{(1 + \nu) \cdot (1 - 2 \cdot \nu)} \\
 B &= \frac{E \cdot \nu}{(1 + \nu) \cdot (1 - 2 \cdot \nu)} \\
 C &= \frac{E \cdot \nu}{2 \cdot (1 + \nu)}
 \end{aligned}$$

In which, E is Young's Module and ν is the passion ratio.

In this case, Young's Modulus E and Passion's ratio ν are set to 1×10^6 GPa and 0.3, respectively. Based on the Equation 7.3.1, the theoretical strain in the

X direction and the Y direction is -9.0×10^{-6} , in the Z direction is 3.0×10^{-4} , and in three shear directions ($\gamma_{xy}, \gamma_{yz}, \gamma_{zx}$) should be zero. The simulation stress and strain at Gaussian Point are shown in Table 7.3.5, which is shown in good agreement with the theoretical value and the error is negligible.

Table 7.3.5 The elastic stress field and strain field for the FE model under 30MPa uniform loading of 3D 8-node hexahedron. (At Gauss Point)

Element No.	Direction	Stress (Uni: MPa)	Strain
1	X	-3.663×10^{-14}	-8.999×10^{-6}
	Y	-3.197×10^{-14}	-9.000×10^{-6}
	Z	30.000	3.000×10^{-4}
	XY	0.000	0.000
	YZ	0.000	0.000
	ZX	-2.669×10^{-13}	-6.939×10^{-18}
2	X	-6.716×10^{-13}	-9.000×10^{-6}
	Y	-2.693×10^{-13}	-9.000×10^{-6}
	Z	30.000	3.000×10^{-4}
	XY	-2.106×10^{-13}	-6.476×10^{-18}
	YZ	-6.672×10^{-14}	1.736×10^{-18}
	ZX	-6.672×10^{-14}	-1.736×10^{-18}

7.3.3 Validation of Multi-Axial Loading

In this case, the loading is applied on the top surface and right surface with 30MPa tension to the Z and the X directions, shown in Figure 7.3.3. The theoretical stress at Gaussian Point is 30 MP in the X direction and the Z direction, and the Y and three shear directions should be zero.

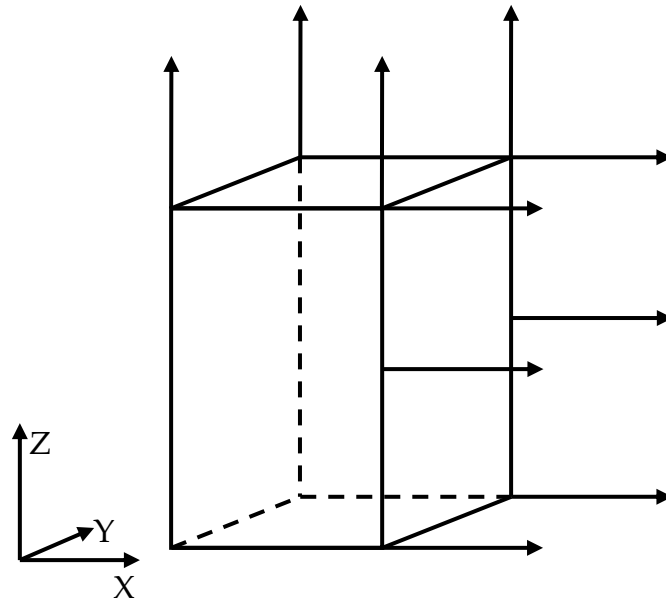


Figure 7.3.3 The schematic figure of Bi-loading model.

According to the Equation 7.3.1, the theoretical strain in the X direction and the Z direction is 2.1×10^{-4} , in the Y direction is -1.8×10^{-4} , and in three shear directions ($\gamma_{xy}, \gamma_{yz}, \gamma_{zx}$) should be zero. The simulation stress and strain at Gaussian Point are shown in Table 7.3.6, which is shown in good agreement with the theoretical value and the error is negligible.

Table 7.3.6 The elastic stress field and strain field for the FE model under Bi-axial loading of 3D 8-node hexahedron. (At Gauss Point)

Element No.	Direction	Stress (Uni: MPa)	Strain
1	X	30.000	2.100×10^{-4}
	Y	-2.132×10^{-14}	-1.800×10^{-4}
	Z	30.000	2.100×10^{-4}
	XY	-8.340×10^{-15}	-2.168×10^{-19}
	YZ	-1.459×10^{-14}	-3.795×10^{-19}
	ZX	-1.043×10^{-15}	2.711×10^{-20}
2	X	30.000	2.100×10^{-4}
	Y	3.552×10^{-15}	-1.800×10^{-5}
	Z	30.000	2.100×10^{-4}
	XY	-2.085×10^{-15}	-5.421×10^{-20}
	YZ	1.876×10^{-15}	4.879×10^{-20}
	ZX	-4.378×10^{-15}	-1.138×10^{-19}

7.4 Elastic Validation of Grain Boundary Part

7.4.1 FE Model

In this 3D version procedure, the 8-node Goodman element is used to mesh the grain boundary part. The benchmark of the procedure for this element is performed in this section and the FE model is shown in Figure 7.4.1.

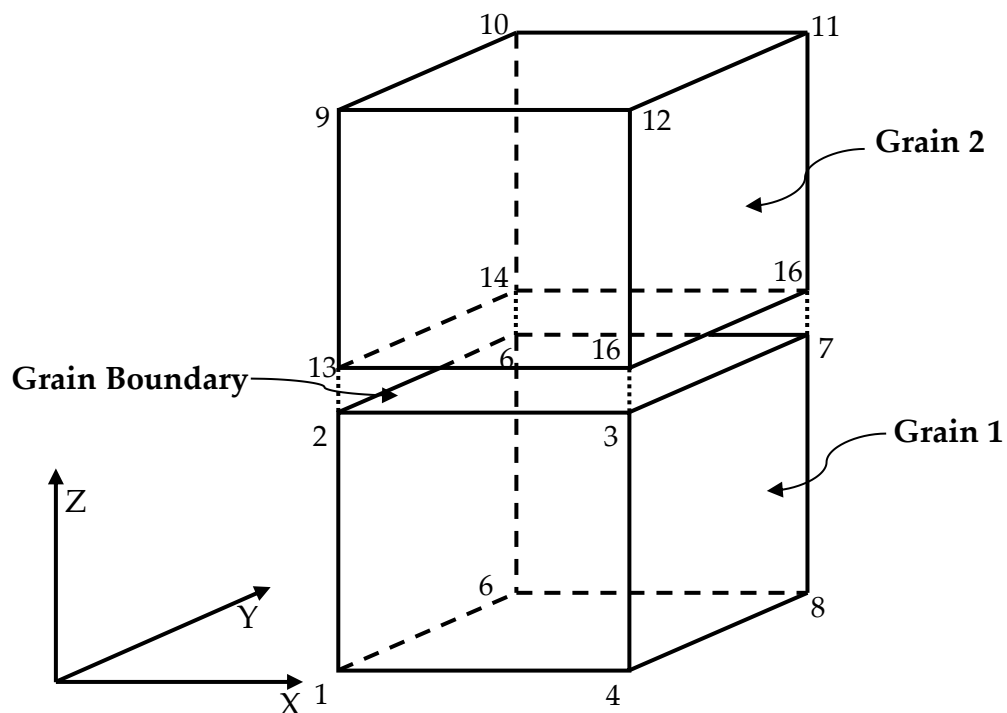


Figure 7.4.1 The schematic figure of Bi-grain case study.

The FE model consists of two parts: two 8-nodal hexahedron elements for grain part and one 8-nodal 3D Goodman element for the grain boundary part. The area integration of grain and grain boundary part is implemented by one Gaussian integration point, respectively.

- *The Geometry information, Boundary Condition and Loading Information and Material Parameters.*

In this FE model, there are a total of 16 nodes to form 3 elements. The coordinates of these 16 nodes and the topology of the elements are stored in the 'DAT' file. The coordination of the node is shown in Table 7.4.1 and the topology information is shown in Table 7.4.2.

Table 7.4.1 The coordinates of the node. (Unit: mm)

Node NO.	X direction	Y direction	Z direction
1	0	0	0
2	0	0	1
3	1	0	1
4	1	0	0
6	0	1	0
6	0	1	1
7	1	1	1
8	1	1	0
9	0	0	2
10	0	1	2
11	1	1	2
12	1	0	2
13	0	0	1
14	0	1	1
16	1	1	1
16	1	0	1

Table 7.4.2 The topology information of these two elements.

Element NO.	Topology Information
Grain 1	1, 2, 3, 4, 6, 6, 7, 8, 16
Grain 2	13, 9, 12, 16, 14, 10, 11, 16
Grain Boundary	13, 14, 16, 16, 2, 6, 7, 3

The boundary condition is imposed on the bottom surface such that the displacement components to the Z direction and the implementation of the boundary condition is through adding the constraint of the node, in this case, the node constraint information is listed in Table 7.4.3.

Table 7.4.3 The constrained node of boundary condition.

The Constraint Direction	Node Number
X direction	1, 2, 6, 6, 9, 10, 13, 14
Y direction	1, 2, 3, 4, 9, 12, 13, 16
Z direction	1, 4, 6, 8

In this benchmark case, a uniform load 30 MPa is applied on the top surface to the Z direction, as shown in Figure 7.4.2. The equivalent node loading factor information is shown in Table 7.4.4.

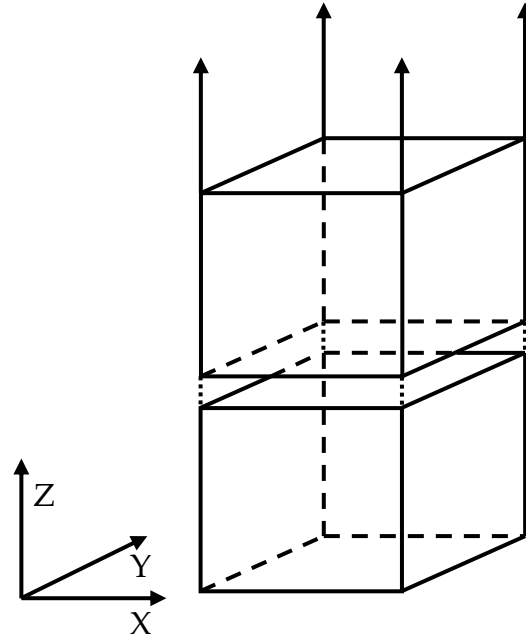


Figure 7.4.2 The schematic figure of tension model of Bi-grains case.

Table 7.3.4 The equivalent node loading factor information.

Node NO.	X	Y	Z
9	0.0	0.0	0.26
10	0.0	0.0	0.26
11	0.0	0.0	0.26
12	0.0	0.0	0.26

The theoretical stress at the Gaussian point of two grains is 30MPa to the Z direction, and in the X, the Y, and three shear directions should be zero. In the grain boundary part, the stress at Gaussian point is 30MPa to the normal direction, and in two separated directions should be zero. According to the stress-jump displacement constitutive relationship, the relative-displacement of three directions can be calculated with the following:

$$\begin{aligned}\Phi_{\mu} &= \frac{F_{\mu}}{K_{\mu}} \\ \Phi_{\nu} &= \frac{F_{\nu}}{K_{\nu}} \\ \Phi_{\omega} &= \frac{F_{\omega}}{K_{\omega}}\end{aligned}\tag{7.4.1}$$

In which, K_{μ} , K_{ν} , and K_{ω} are the stiffness of three directions.

In this case, K_{μ} , K_{ν} , and K_{ω} are set to 1×10^6 , 6×10^6 , and 6×10^6 with unit MPa, respectively. According to the Equation 7.4.1, the jump displacement in normal direction is 3×10^{-4} mm, and in other two directions should be zero. The simulation stress and jump displacement at Gaussian Point is shown in Table 7.4.5, which is shown in good agreement with the theoretical value and the error is negligible.

Table 7.4.5 The elastic stress field and jump displacement/strain field for the bi-crystal FE model. (At Gauss Point)

Element No.	Direction	Stress (Uni: MPa)	Jump Displacement (Unit: mm)
Grain Boundary	Normal	30.000	-3.000×10^{-4}
	Separated	2.386×10^{-11}	4.770×10^{-17}
	Separated	-2.429×10^{-12}	-2.429×10^{-17}
Element No.	Direction	Stress (Uni: MPa)	Strain
Grain 1	X	2.338×10^{-12}	-9.000×10^{-6}
	Y	3.137×10^{-12}	-9.000×10^{-6}
	Z	30.000	3.000×10^{-4}
	XY	4.637×10^{-12}	1.179×10^{-16}
	YZ	-2.402×10^{-12}	-6.246×10^{-17}
	ZX	-3.202×10^{-12}	-8.327×10^{-17}
Grain 2	X	-2.136×10^{-12}	-9.000×10^{-6}
	Y	-8.641×10^{-12}	-9.000×10^{-6}
	Z	30.000	3.000×10^{-4}
	XY	-3.203×10^{-12}	-8.327×10^{-17}
	YZ	-2.136×10^{-12}	6.661×10^{-17}
	ZX	-6.338×10^{-12}	-1.388×10^{-16}

7.4.2 Validation of Separated Direction

In order to add the separated node force to the four nodes on the upper surface of grain boundary uniformly and directly, the upper grain is removed, and the FE model for validating separated direction is shown in Figure 7.4.3, which consists of two parts: one 8-node hexahedron elements for grain part and one 3D Goodman element for the grain boundary part.

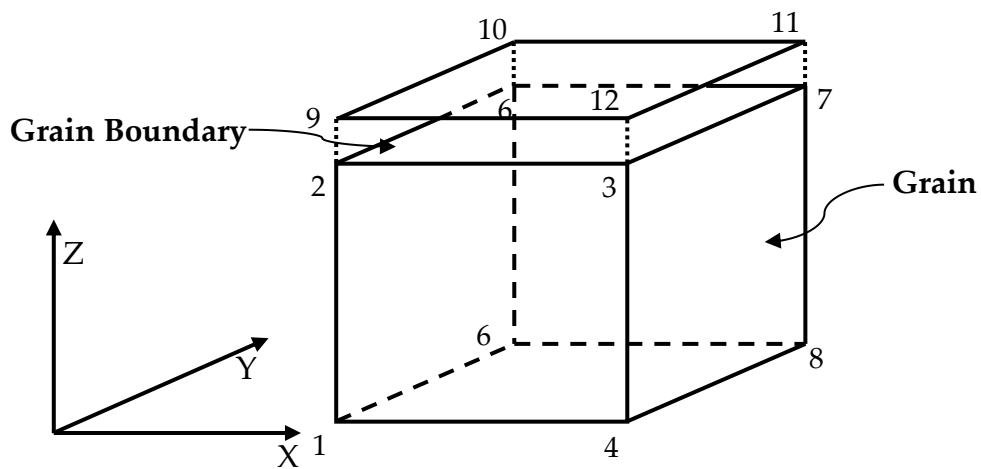


Figure 7.4.3 The schematic figure for validation of Grain boundary at separated direction.

- *The Geometry information, Boundary Condition and Loading Information and Material Parameters.*

In this FE model, there are a total of 12 nodes to form 2 elements. The coordination of the node is shown in Table 7.4.6 and the topology information is shown in Table 7.4.7.

Table 7.4.6 The coordinates of the node. (Unit: mm)

Node NO.	X direction	Y direction	Z direction
1	0	0	0
2	0	0	1
3	1	0	1
4	1	0	0
6	0	1	0
6	0	1	1
7	1	1	1
8	1	1	0
9	0	0	1
10	0	1	1
11	1	1	1
12	1	0	1

Table 7.4.7 The topology information of these two elements.

Element NO.	Topology Information
Grain	1, 2, 3, 4, 6, 6, 7, 8, 16
Grain Boundary	9, 10, 11, 12, 2, 6, 7, 3

The boundary condition is imposed on the bottom surface such that the displacement components to the Z direction, the node constraint information is list in Table 7.4.8.

Table 7.4.8 The constrained node of boundary condition.

The Constraint Direction	Node Number
X direction	1
Y direction	1
Z direction	1, 4, 6, 8

In this benchmark case, a uniform loads 30MPa is applied on the top surface to the X direction, as shown in Figure 7.4.4.

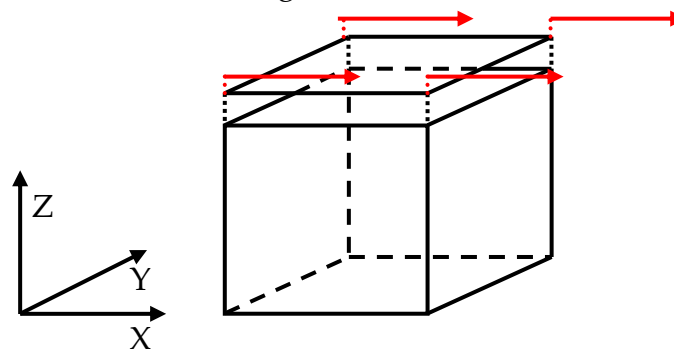


Figure 7.4.4 The schematic figure of tension model for validating separated direction.

The equivalent node loading factor information is shown in Table 7.4.9.

Table 7.4.9 The equivalent node loading factor information.

Node NO.	X	Y	Z
9	0.26	0.0	0.0
10	0.26	0.0	0.0
11	0.26	0.0	0.0
12	0.26	0.0	0.0

In this case, only one Gaussian point does not satisfy the accuracy requirements for the grain boundary part. Here, the integral of the grain boundary area is implemented by nine Gaussian Integration Points. The theoretical stress at these nine Gaussian points are 30MPa to the separated

direction, whereas in normal directions and other separated directions it should be zero. According to Equation 7.4.1, the jump-displacement in one separated direction is 6.0×10^{-4} mm, and in other two directions should be zero. The simulation stress and jump displacement at nine Gaussian Points are shown in Table 7.4.10, which is shown in good agreement.

Table 7.4.10 The elastic stress field and jump-displacement for the grain boundary part. (At Gauss Point)

GP NO.	Stress (Unit: MPa)		
Direction	Normal	Separated	Separated
1	-1.689×10^{-14}	29.999	3.706×10^{-16}
2	-8.082×10^{-16}	30.000	-1.069×10^{-16}
3	3.388×10^{-16}	29.999	2.966×10^{-16}
4	-4.066×10^{-16}	30.000	8.682×10^{-16}
6	1.366×10^{-16}	29.999	6.082×10^{-16}
6	8.132×10^{-16}	30.000	1.694×10^{-16}
7	3.388×10^{-16}	29.999	1.779×10^{-14}
8	2.711×10^{-16}	30.000	2.033×10^{-20}
9	6.421×10^{-16}	29.999	0.000
	Jump Displacement (Unit: mm)		
1	-1.169×10^{-19}	6.999×10^{-6}	7.412×10^{-21}
2	-6.082×10^{-20}	6.000×10^{-6}	-2.118×10^{-21}
3	3.388×10^{-20}	6.999×10^{-6}	6.929×10^{-21}
4	-4.066×10^{-20}	6.000×10^{-6}	1.736×10^{-20}
6	-1.366×10^{-20}	6.999×10^{-6}	1.016×10^{-20}
6	8.132×10^{-20}	6.000×10^{-6}	3.388×10^{-21}
7	3.388×10^{-20}	6.999×10^{-6}	3.668×10^{-20}
8	2.711×10^{-20}	6.000×10^{-6}	2.033×10^{-20}

9	6.421×10^{-20}	6.999×10^{-6}	0.000
---	-------------------------	------------------------	-------

7.4.3 Validation of Rotation Status

In this section, the main task is to verify the accuracy and reliability of the rotating system. The design of the case study is from Q.Xu's idea, the FE model is obtained by rotating the previous model (Figure 7.4.4) by 45° counterclockwise around the Y-axis, as shown in Figure 7.4.5. The model with the red broken line is the original model before the rotation.

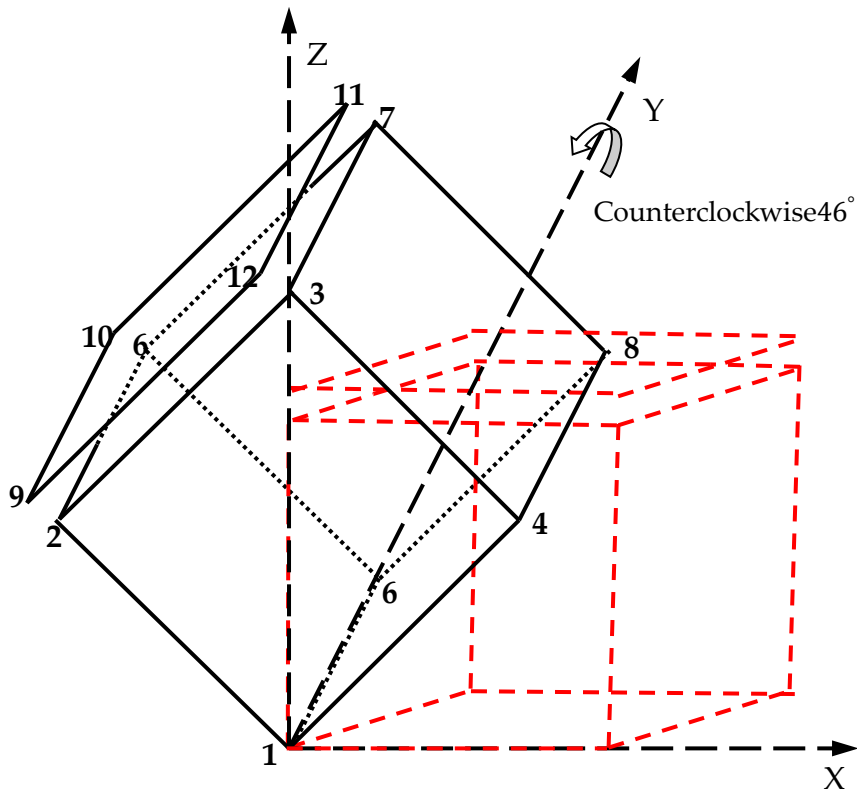


Figure 7.4.5 The schematic figure for validation the rotation system.

- *The Geometry information, Boundary Condition and Loading Information and Material Parameters.*

In this FE model, there are a total of 12 nodes to form 2 elements. The coordinates of the node in the global system is shown in Table 7.4.11 and the topology information is shown in Table 7.4.7.

Table 7.4.11 The coordinates of the node. (Unit: mm)

Node NO.	X direction	Y direction	Z direction
1	0	0	0
2	$-\frac{\sqrt{2}}{2}$	0	$\frac{\sqrt{2}}{2}$
3	0	0	$\sqrt{2}$
4	$\frac{\sqrt{2}}{2}$	0	$\frac{\sqrt{2}}{2}$
6	0	1	0
6	$-\frac{\sqrt{2}}{2}$	1	$\frac{\sqrt{2}}{2}$
7	0	1	1
8	$\frac{\sqrt{2}}{2}$	1	$\frac{\sqrt{2}}{2}$
9	$-\sqrt{2}$	0	$\sqrt{2}$
10	$-\sqrt{2}$	1	$\sqrt{2}$
11	$-\frac{\sqrt{2}}{2}$	1	$\frac{2 + \sqrt{2}}{2}$
12	$-\frac{\sqrt{2}}{2}$	0	$\frac{2 + \sqrt{2}}{2}$

The boundary condition is imposed on the bottom surface such that the displacement components to the X and the Z directions, in this case, the node constraint information is listed in Table 7.4.12.

Table 7.4.12 The constrained node of boundary condition.

The Constraint Direction	Node Number
X direction	1,4, 6, 8
Y direction	1
Z direction	1, 4, 6, 8

In this benchmark case, a uniform loads 30MPa is applied on the top surface to the normal direction, as shown in Figure 7.4.6.

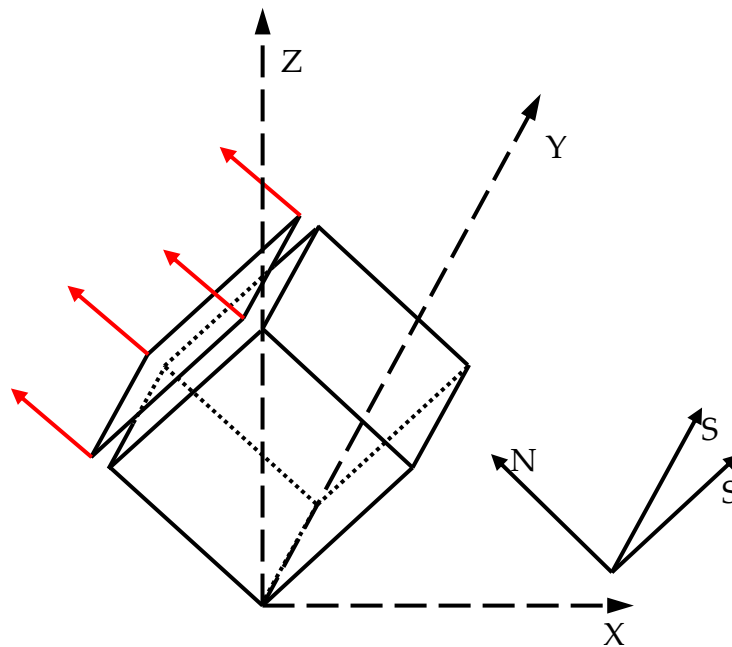


Figure 7.4.6 The schematic figure of tension model for validating rotating system.

The equivalent node loading factor information is shown in Table 7.4.13.

Table 7.4.13 The equivalent node loading factor information.

Node NO.	X	Y	Z
9	$-\frac{\sqrt{2}}{8}$	0.0	$\frac{\sqrt{2}}{8}$
10	$-\frac{\sqrt{2}}{8}$	0.0	$\frac{\sqrt{2}}{8}$
11	$-\frac{\sqrt{2}}{8}$	0.0	$\frac{\sqrt{2}}{8}$
12	$-\frac{\sqrt{2}}{8}$	0.0	$\frac{\sqrt{2}}{8}$

In this case, as the same with the previous case, nine Gaussian Points are used to implement the integration of grain boundary region. The theoretical stress at these nine Gaussian points is 30MPa to the normal direction, and in two separated directions should be zero. According to Equation 7.4.1, the jump-displacement in normal direction is 3.0×10^{-4} mm, and in the other two directions should be zero. The simulation stress and jump displacement at nine Gaussian Points are shown in Table 7.4.14, which is shown in good agreement.

Table 7.4.14 The elastic stress field and jump-displacement for the grain boundary part. (At Gauss Point)

GP NO.	Stress (Unit: MPa)		
Direction	Normal	Separated	Separated
1	29.999	3.642×10^{-14}	-7.417×10^{-16}
2	29.999	1.779×10^{-14}	-2.118×10^{-21}
3	29.999	-6.929×10^{-16}	6.929×10^{-16}
4	29.999	8.893×10^{-16}	-2.076×10^{-16}
6	29.999	-6.776×10^{-16}	-7.623×10^{-16}
6	29.999	-6.776×10^{-16}	3.388×10^{-16}
7	29.999	-1.948×10^{-14}	6.776×10^{-16}
8	29.999	-1.694×10^{-20}	-2.641×10^{-16}
9	29.999	-4.066×10^{-20}	-6.421×10^{-20}
	Jump Displacement (Unit: mm)		
1	2.999×10^{-4}	7.284×10^{-20}	-7.417×10^{-20}
2	2.999×10^{-4}	3.668×10^{-20}	-2.118×10^{-21}
3	2.999×10^{-4}	-1.186×10^{-20}	6.929×10^{-20}
4	2.999×10^{-4}	1.779×10^{-20}	-2.076×10^{-20}
6	2.999×10^{-4}	-1.366×10^{-20}	-7.623×10^{-21}
6	2.999×10^{-4}	-1.366×10^{-20}	3.388×10^{-21}
7	2.999×10^{-4}	-3.896×10^{-20}	6.776×10^{-21}
8	2.999×10^{-4}	-3.388×10^{-20}	-2.641×10^{-20}

9	2.999×10^{-4}	-4.066×10^{-20}	-6.421×10^{-20}
---	------------------------	--------------------------	--------------------------

7.5 Validation of the Non-linear Creep

The benchmark strategy for the 3D version is similar to the previous 2D version, the verification of the accuracy and stability relies on the Bi-grains case study. In this section, the main task is to report the verification of the procedure in the normal direction.

The information of the FE model, boundary condition, loading, and elastic parameters have been mentioned in section 7.4.1. The benchmark concentrates on the integral accuracy of the creep constitutive equation and the stress convergence of the non-linear iteration.

7.5.1 The Validation of the Stress Update

The sign of the non-linear iterating convergence is the elastic stress of each element that are the same. In this case, the non-linear iteration lasts for 141387 steps with the time step of 0.000001 (normalized). In order to visualize the convergence of the stress field, the stress condition of two iterative steps (60000 and 141386) are selected randomly during the process and shown in Table 7.5.1.

Table 7.5.1 The elastic stress field of each element at selected step.

Iteration Step		Element No.	Direction	Elastic Stress (Unit: MPa)
60000	Grain	1	X	2.694×10^{-11}
			Y	1.886×10^{-11}
			Z	19.999
			XY	4.897×10^{-12}
			YZ	-1.896×10^{-12}
			ZX	5.773×10^{-12}
	Grain	2	X	8.673×10^{-12}
			Y	1.501×10^{-11}
			Z	19.999
			XY	4.465×10^{-12}
			YZ	3.911×10^{-12}
			ZX	1.400×10^{-11}
GB	3	X	1.387×10^{-11}	
		Y	0.000	
		Z	19.999	
141386	Grain	1	X	-4.987×10^{-11}
			Y	1.189×10^{-10}
			Z	20.000
			XY	1.036×10^{-11}

			YZ	-7.344×10^{-11}	
			ZX	-5.121×10^{-12}	
		2	X	1.228×10^{-10}	
			Y	9.535×10^{-10}	
			Z	20.000	
			XY	-3.955×10^{-10}	
			YZ	-1.532×10^{-10}	
			ZX	2.952×10^{-12}	
		GB	3	X	-6.661×10^{-10}
				Y	2.359×10^{-10}
Z	20.000				

According to the stress field shown in Table 7.5.1 the result shows good agreement with the theoretical results that the applied stress of each element keeps constant during the non-linear iteration period. For the grain part, the maximum final cumulative error in the Y direction is 9.535×10^{-10} MPa (NO.2element). For the grain boundary part, the maximum final cumulative error in separate direction is -6.661×10^{-10} MPa (NO.3 element). In summary, the in-house procedure has shown a good convergence under the uni-axial loading.

7.5.2 The Validation of the Integration

In this procedure, the built-in constitutive equations for describing the creep evolution of grain boundary part has been mentioned in section 2.9.2. In this

mathematical model, the creep evolution is determined by three key variables: ρ (the cavity density), β (the damage variable), and a (the average radius of cavities), Therefore, these three parameters are treated as indicators of the benchmark.

The case setting is similar in section 6.2, which simulates the creep evolution of Copper at 600°C and the loading is on the top surface with 20 MPa. The trend of three key indicators versus the normalized time is shown in Figure 7.5.1 to Figure 7.5.3. According to the analysis in section 6.2, it is clear that the trend of the three indicators is as expected.

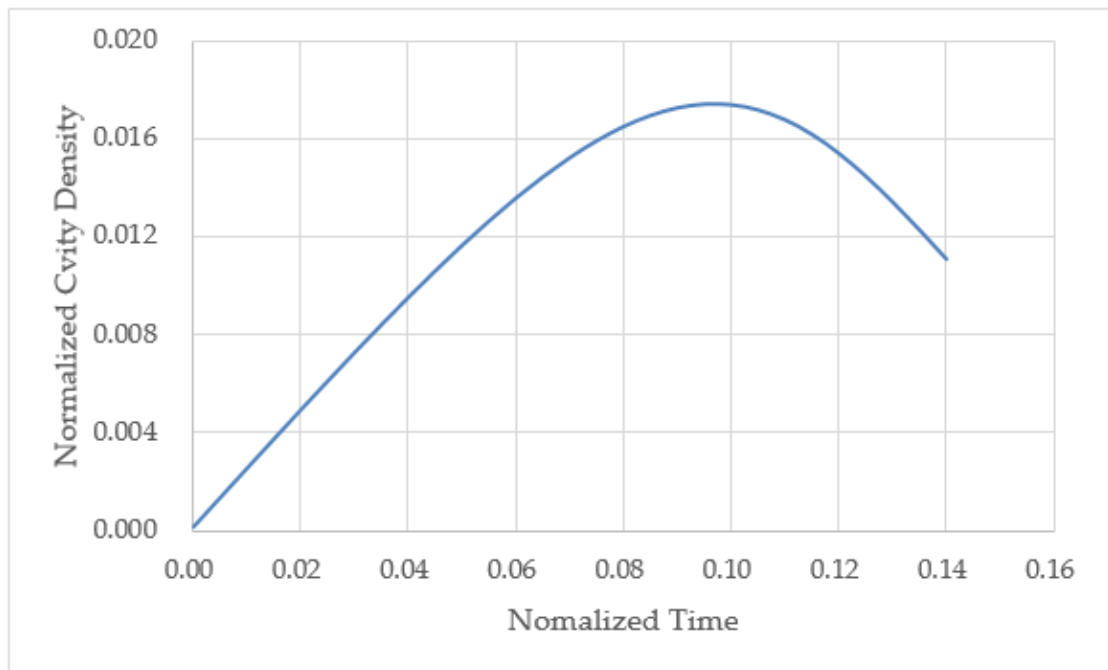


Figure 7.5.1 The trend of the normalized cavity density $\bar{\rho}$ versus normalized time.

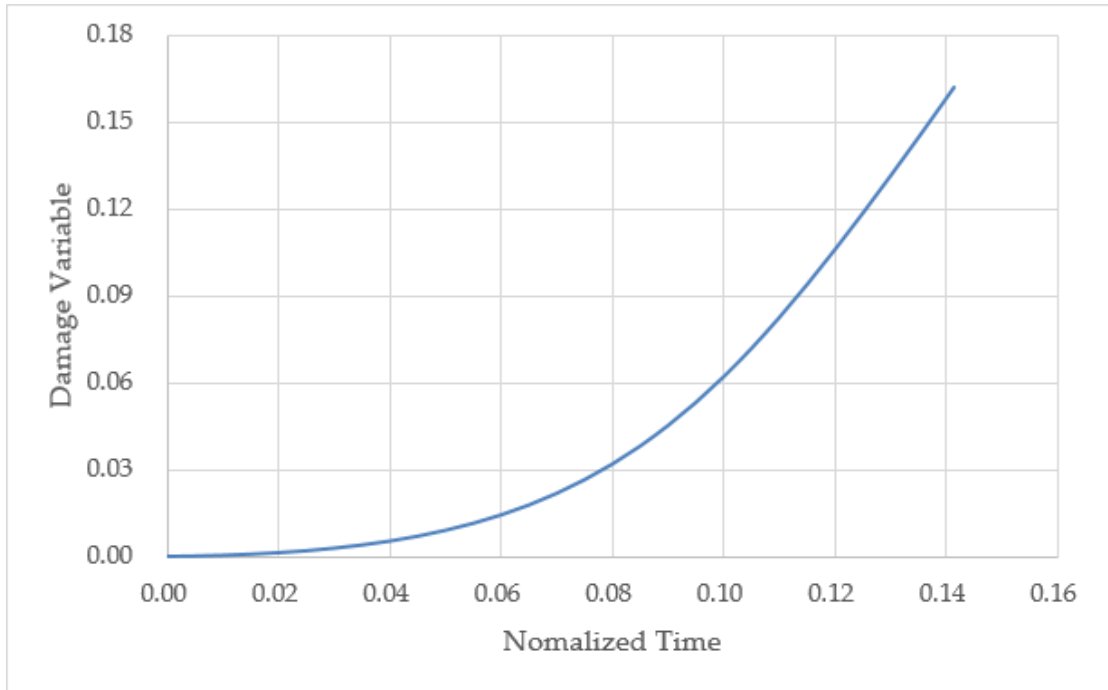


Figure 7.5.2 The trend of the damage variable β versus normalized time.

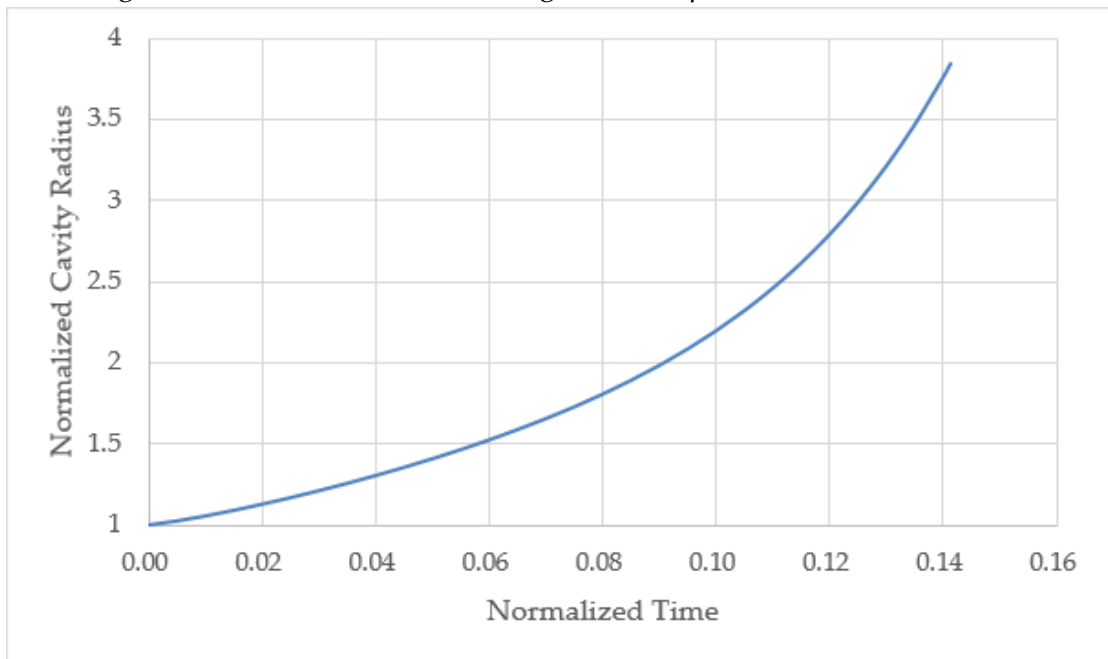


Figure 7.5.3 The trend of the normalized cavity radius \bar{a} versus normalized time.

7.5.3 Error Analysis

As mentioned in section 6.2, the simulation of the constitutive equations under uniaxial loading conditions is the benchmark result [68]. The

percentage error between the reference result and the simulation is shown in

Table 7.5.2.

Table 7.5.2 The percentage error of curves between simulation and reading value.

Name	Reading Value	Simulation Value	Percentage Error
Rupture Time(normalized)	0.15	0.1413	5.8%
Maximum value of the damage variable β	0.166	0.1620	2.41%
Final value of cavity density ρ (normalized)	0.010	0.0107	7.00%
The change point value of the cavity density ρ (normalized)	0.017	0.0174	2.35%
The time point of the cavity density change point(normalized)	0.096	0.0968	0.83%

7.5.4 Conclusion

Based on the above benchmark process, it demonstrates the numerical accuracy and stability of the 3D version procedure. For the non-linear iteration part, it shows the good convergence during the stress updating. In summary, the preliminary verification is shown to be in good agreement with theoretical results.

Chapter 8 Conclusions and Future Work

The contribution of this project is to develop an in-house procedure for Finite Element simulation of creep evolution at grain boundary level. In this chapter, detailed achievements and future work have been summarized below.

8.1 Contribution and Conclusion

An In-house procedure is developed to do the Finite Element Analysis of the creep evolution at grain boundary level. My main contribution is to develop a subroutine library and modify the P61 program to obtain this computational capability. Specifically,

1. The following five sets of constitutive equations have been built into the modified program, including: Kachanov-Rabatnov, Kachanov-Rabatnov-Hayhurst, Kachanov-Rabatnov-Hayhurst-Xu, Vöse's cavitation equations, and Newtonian Viscous Flow equation.
2. The benchmark of the in-house procedure, including:
 - a) the elastic benchmark of the plane stress element, plane strain element, axisymmetric element, 3-dimensional element, 2-dimensional 4-node Goodman element, and 3-dimensional 8-node Goodman element.
 - b) the creep benchmark of a simple quadrilateral FE model, notched bar FE model, and 2D Bi-grains FE model, and 3D Bi-grains FE model.
3. The application of the in-house procedure in a polycrystalline case study.

The in-house procedure was used to simulate the creep evolution of the copper polycrystalline structure at the grain boundary level, and the results were published [45].

8.2 Future Work

In this section, a detailed description of the possible future work based on the current work is summarized below.

1. A 3D polycrystalline case study should be subjected. The 3D version procedure has been developed to achieve the mechanical behavior of the 3-dimensional Goodman element and have passed the preliminary tests.
2. Develop parallel computing capability for the procedure. Compared with the 2-dimensional case, the project size of the 3-dimensional case increases sharply, therefore, it is necessary to make the procedure have the parallel computing capability to improve the computational efficiency to reduce the computational consumption and time.
3. Apply the procedure to conduct a parametric study. Based on this research, to obtain the relative importance of various parameters in micro creep modeling.

Reference

- [1] Ling, X., Tu, S. and Gong, J. (1999). Damage Mechanics Considerations for Life Extension of High-Temperature Components. *Journal of Pressure Vessel Technology*, 122(2), pp.174-179.
- [2] Becker, A., Hyde, T. and Xia, L. (1994). Numerical analysis of creep in components. *The Journal of Strain Analysis for Engineering Design*, 29(3), pp.185-192.
- [3] Hayhurst, D., Dimmer, P. and Morrison, C. (1984). Development of Continuum Damage in the Creep Rupture of Notched Bars. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 311(1516), pp.103-129.
- [4] Kattan P.I., Voyiadjis G.Z. (2002) DNA Commands. In: *Damage Mechanics with Finite Elements*. Springer, Berlin, Heidelberg.
- [5] Ivt-abaqusdoc.ivt.ntnu.no. (2019). Abaqus 6.14 Documentation. [online] Available at: <http://ivt-abaqusdoc.ivt.ntnu.no:2080/teXis/search/?query=wetting&submit.x=0&submit.y=0&group=bk&CDB=v6.14> [Accessed 15 Sep. 2019].
- [6] Kachanov, L. (1958). On rupture time under condition of creep. *IzvestiaAkademiiNauk USSR, Otd. Techn. Nauk, Moskva*, 8, 26-31
- [7] Murakami, S. (2012.02.03), *Continuum Damage Mechanics: A Continuum Mechanics Approach to the Analysis of Damage and Fracture*, Springer, ISBN 9400726651.
- [8] Rabotnov, Y. N. (1969). *Creep problems in structural members*.

- [9] Yu, C., Huang, C., Chen, C., Gao, Y. and Hsueh, C. (2012). Effects of grain boundary heterogeneities on creep fracture studied by rate-dependent cohesive model. *Engineering Fracture Mechanics*, 93, pp.48-64.
- [10] Vöse, M., Otto, F., Fedelich, B. and Eggeler, G. (2014). Micromechanical investigations and modelling of a Copper–Antimony-Alloy under creep conditions. *Mechanics of Materials*, 69(1), pp.41-62.
- [11] Liu, D. (2015). The development of finite element software for creep damage mechanics. Doctor of Philosophy. The university of Huddersfield.
- [12] Richard E. Goodman, Robert L. Taylor, Tor L. Brekke (1964). A Model for the Mechanics of Jointed Rock. *Journal of the Soil Mechanics and Foundations Division*, 1968, Vol. 94, Issue 3, Pg. 637-660
- [13] Smith, I., Griffiths, D. and Margetts, I. (2014). PROGRAMMING THE FINITE ELEMENT METHOD. 5th ed. Chichester, West Sussex: John Wiley & Sons Ltd.
- [14] Ashby, M. (1972). Boundary defects, and atomistic aspects of boundary sliding and diffusional creep. *Surface Science*, 31, pp.498-542.
- [15] Betten, J. (2005). Creep mechanics (2nd edition). Berlin; New York, Springer.
- [16] Hult, J. A. (1966). Creep in engineering structures: Blaisdell Pub. Co.
- [17] Finnie, I., and Heller, W. R. (1959). Creep of Engineering Materials
- [18] Es-Souni, M. (2000). Primary, secondary and anelastic creep of a high temperature near α -Ti alloy Ti6242Si. *Materials Characterization*, 45(2), pp.153-164.

- [19]Frost, H. J., and Ashby, M. F. (1982). Deformation mechanism maps: the plasticity and creep of metals and ceramics
- [20]J.J. Skrzypek, in: R.B. Hetnarski (1993), Plasticity and Creep: Theory, Examples, and Problems, CRC Press.
- [21]Nabarro, F. (1967). Steady-state diffusional creep. Philosophical Magazine, 16(140), pp.231-237.
- [22]Shibli, A. and Holdsworth, S. (2008). Creep and fracture in high-temperature components—Design and life assessment issues. International Journal of Pressure Vessels and Piping, 85(1-2), p.1.
- [23]Abe, F., Kern, T.-U., and Viswanathan, R. (2008). Creep-resistant steels:Elsevier.
- [24]Parker, J. (1995). Creep behavior of low alloy steel weldment. International journal of pressure vessels and piping, 63(1), 55-62. Goldhoff and Woodford, 1972)
- [25]Gorash, Y. (2008.). Development of a creep-damage model for non-isothermal long-term strength analysis of high-temperature components operating in a wide stress range.
- [26]Riedel. and Baumann. (1987). Fracture at High Temperatures. [Place of publication not identified]: Springer Berlin Heidelberg.
- [27]R. Goldhoff and D. Woodford (1972), The Evaluation of Creep Damage in a Cr-Mo-V Steel, in Testing for Prediction of Material Performance in Structures and Components, ed. R. Shane (West Conshohocken, PA: ASTM International), 89-106.
- [28]Ashby, M. and Brown, L. (1983). Perspectives in Creep Fracture. Kent: Elsevier Science
- [29]Penny, R. K., Marriott, D. L., &Boresi, A. P. (1995). Design for creep.

- [30] C.L. Bauer, "Polygonization of Rock Salt", *Trans. Metall. Soc. of AIME* 223 [4] (1965) 846-847
- [31] Rollason, E. C. (1973). *Metallurgy for engineers*: Edward Arnold London
- [32] Passchier, C. (2014). *Microtectonics*. [Place of publication not identified]: Springer.
- [33] H., Courtney, Thomas (1990). *Mechanical Behavior of Materials :Solutions Manual to Accompany*. New York: McGraw-Hill,Inc. ISBN 0070132666. OCLC 258076725.
- [34] Goretta, K., Cruse, T., Koritala, R., Routbort, J., Mélendez-Martínez, J. and de Arellano-López, A. (2001). Compressive creep of polycrystalline ZrSiO₄. *Journal of the European Ceramic Society*, 21(8), pp.1055-1060.
- [35] Aurenhammer, F. (1991). Voronoi diagrams---a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3), pp.345-405.
- [36] Barber, C., Dobkin, D. and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4), pp.469-483.
- [37] Barber, C., Dobkin, D. and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4), pp.469-483.
- [38] Rycroft, C. (2009). VORO++: A three-dimensional Voronoi cell library in C++. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 19(4), p.041111.
- [39] Quey, R., Dawson, P. and Barbe, F. (2011). Large-scale 3D random polycrystals for the finite element method: Generation, meshing and remeshing. *Computer Methods in Applied Mechanics and Engineering*, 200(17-20), pp.1729-1745.

- [40] NIST. (2019). Object Oriented Finite Elements (OOF). [online] Available at: <https://www.nist.gov/programs-projects/object-oriented-finite-elements-oof> [Accessed 15 Sep. 2019].
- [41] van der Giessen, E. and Tvergaard, V. (1991). A creep rupture model accounting for cavitation at sliding grain boundaries. *International Journal of Fracture*, 48(3), pp.153-178.
- [42] Onck, P. and van der Giessen, E. (1997). Microstructurally-based modelling of intergranular creep fracture using grain elements. *Mechanics of Materials*, 26(2), pp.109-126.
- [43] Barenblatt, G. (1959). Equilibrium cracks formed during brittle fracture rectilinear cracks in plane plates. *Journal of Applied Mathematics and Mechanics*, 23(4), pp.1009-1029.
- [44] Me'ric, L., Poubanne, P. and Cailletaud, G. (1991). Single Crystal Modeling for Structural Calculations: Part 1 – Model Presentation. *Journal of Engineering Materials and Technology*, 113(1), pp.162-170.
- [45] Xu, Q., Tu, J. and Lu, Z. (2019). Development of the FE In-House Procedure for Creep Damage Simulation at Grain Boundary Level. *Metals*, 9(6), p.656.
- [46] Шамровский, А. and Богданова, Е. (2014). Solution of contact problems of elasticity theory using a discrete finite-size element. *Eastern-European Journal of Enterprise Technologies*, 3(7(69)), p.41.
- [47] Luo, P., Liu, G., Wang, W. and Xu, C. (2016). The Viscoelastic boundary simulation based on Non-thickness Goodman element. *Journal of Fuzhou University (Natural Science Edition)*, [online] 44(104-109), pp.105-107. Available at: http://xueshu.baidu.com/usercenter/paper/show?paperid=32abe97b88ff3edf2ca1a8a0d5491884&site=xueshu_se [Accessed 9 Sep. 2019].
- [48] Kozák, Vladislav. (2008). Cohesive Zone Modelling. 1048. 10.1063/1.2990924.

- [49] G.I. Barenblatt (1962). The mathematical theory of equilibrium cracks in brittle fracture. *Advances in Applied Mechanics*. 7. pp. 55–129.
- [50] Donald S. Dugdale (1960). "Yielding of steel sheets containing slits". *Journal of the Mechanics and Physics of Solids*. 8 (2): 100–104.
- [51] Needleman, A. (1987). A Continuum Model for Void Nucleation by Inclusion Debonding. *Journal of Applied Mechanics*, 54(3), pp.525-531.
- [52] Petersson, P. E. (1981). Crack growth and development of fracture zones in plain concrete and similar materials. Technical report, Division of Building Materials, Lund Institute of Technology.
- [53] Carpinteri, A. (1986). Mechanical damage and crack growth in slits. MartinusNijhoff Kluwer.
- [54] Harder, N. (1997). Size effect caused by brittleness in fracturemechanics. *Engineering Fracture Mechanics*, 56(1), pp.3-8.
- [55] Park, K. and Paulino, G. (2011). Cohesive Zone Models: A Critical Review of Traction-Separation Relationships Across Fracture Surfaces. *Applied Mechanics Reviews*, 64(6).
- [56] Zienkiewicz, O., Taylor, R. and Fox, D. (2014). The finite element method for solid and structural mechanics. Oxford:Butterworth-Heinemann.
- [57] Osgerby, S. and Dyson, B. (1993). A methodology for modelling tertiary creep behaviour of engineering alloys under oxidising conditions. *International Journal of Pressure Vessels and Piping*, 55(2), pp.333-341.
- [58] Perrin, I. and Hayhurst, D. (1996). Creep constitutive equations for a 0.5Cr–0.5Mo–0.25V ferritic steel in the temperature range 600–675°C. *The Journal of Strain Analysis for Engineering Design*, 31(4), pp.299-314.

- [59] Xu, Q. (2001). Creep damage constitutive equations for multi-axial states of stress for 0.5Cr0.5Mo0.25V ferritic steel at 590°C. *Theoretical and Applied Fracture Mechanics*, 36(2), pp.99-107.
- [60] Dyson, B. (1983). Continuous cavity nucleation and creep fracture. *Scripta Metallurgica*, 17(1), pp.31-37.
- [61] Dyson, B. (1976). Constraints on diffusional cavity growth rates. *Metal Science*, 10(10), pp.349-353.
- [62] Watanabe, T. (1993). Grain boundary design and control for high temperature materials. *Materials Science and Engineering: A*, 166(1-2), pp.11-28.
- [63] Tvergaard, V. (1984). On the creep constrained diffusive cavitation of grain boundary facets. *Journal of the Mechanics and Physics of Solids*, 32(5), pp.373-393.
- [64] Rice, J. (1981). Constraints on the diffusive cavitation of isolated grain boundary facets in creeping polycrystals. *Acta Metallurgica*, 29(4), pp.675-681
- [65] Cocks, A. and Ashby, M. (1982). Creep fracture by coupled power-law creep and diffusion under multiaxial stress. *Metal Science*, 16(10), pp.465-474.
- [66] Yousefiani, A., Mohamed, F. and Earthman, J. (2000). Creep rupture mechanisms in annealed and overheated 7075 Al under multiaxial stress states. *Metallurgical and Materials Transactions A*, 31(11), pp.2807-2821.
- [67] Kassnera, M.E.; Hayesb, T.A.(2003). Creep cavitation in metals. *Int. J. Plast.* 19, 1715–1748
- [68] Vose, M.; Fedelich, B. (2012); Owen, J. A simplified model for creep induced grain boundary cavitation validated by multiple cavity growth simulations. *Comput. Mater. Sci.* 58, 201–213.
- [69] Wang, M. (1999). Three-Dimensional Finite Element Analysis of Creep Continuum Damage Growth and Failure in Weldments. Ph.D. UMIT.

- [70]Hall, F. (1990). DEVELOPMENT OF CONTINUUM DAMAGE MECHANICS MODELS TO PREDICT THE CREEP DEFORMATION AND FAILURE OF HIGH TEMPERATURE STRUCTURES. Ph.D. The University of Sheffield.
- [71]Hyde, T., Becker, A., Sun, W. and Williams, J. (2006). Finite-element creep damage analyses of P91 pipes. *International Journal of Pressure Vessels and Piping*, 83(11-12), pp.853-863.
- [72]Ralph, P., & Wand, Y. (2009). A proposal for a formal definition of the design concept Design requirements engineering: A ten-year perspective (pp. 103-136): Springer
- [73]IRONS, B. (1966). Engineering applications of numerical integration in stiffness methods. *AIAA Journal*, 4(11), pp.2035-2037.
- [74]Ling, X., Tu, S.-T., & Gong, J.-M. (2000). Application of Runge–Kutta–Merson algorithm for creep damage analysis. *International journal of pressure vessels and piping*, 77(5), 243-248.
- [75]Hayhurst, D., & Henderson, J. (1977). Creep stress redistribution in notched bars. *International journal of mechanical sciences*, 19(3), 133-146.
- [76]Watkins, D., 2010. *Fundamentals Of Matrix Computations*. Hoboken, N.J.: Wiley, p.60.
- [77]Barrett, R., (1994). *Templates For The Solution Of Linear Systems*. Philadelphia, Pa.: Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), p.Skyline Storage (SKS).
- [78]George, A. and Liu, J., (1981). *Computer Solution Of Large Sparse Positive Definite Systems*. Englewood Cliffs, N.J.: Prentice-Hall.

- [79] Hayhurst, D. ,(1973). Stress Redistribution and Rupture Due to Creep in a Uniformly Stretched Thin Plate Containing a Circular Hole. *Journal of Applied Mechanics*, 40(1), pp.244-250.
- [80] Cormeau IC., (1975). Numerical stability in quasi-static elasto-viscoplasticity. *Int J Numer Methods Eng* 9(1),109–127.
- [81] D.R.Hayhurst, D.A.Miller,(1997). The use of creep constitutive damage mechanics to predict damage evolution and failure in weld vessels, in: *IMEch Seminar: Remanent Life Prediction*, London.
- [82] Li, G., Thomas, B. and Stubbins, J. (2000). Modeling creep and fatigue of copper alloys. *Metallurgical and Materials Transactions A*, 31(10), pp.2491-2502.
- [83] Kachanov, L. (1968). On rupture time under condition of creep. *Izvestia Akademi Nauk USSR, Otd. Techn. Nauk, Moskwa*, 8, 26-31.
- [84] Hayhurst, D. (1972). Creep rupture under multi-axial states of stress. *Journal of the Mechanics and Physics of Solids*, 20(6), pp.381-382.
- [85] Hayhurst, D., Dimmer, P. and Chernuka, M. (1976). Estimates of the creep rupture lifetime of structures using the finite element method. *Journal of the Mechanics and Physics of Solids*, 23(4-6), pp.336-360.
- [86] Machiels, L., & Deville, M. (1997). Fortran 90: an entry to object-oriented programming for the solution of partial differential equations. *ACM Transactions on Mathematical Software (TOMS)*, 23(1), 32-49.
- [87] Akin, J. (1999). Object oriented programming via Fortran 90. *Engineering Computations*, 16(1), 26-48.
- [88] Timoshenko, S. (1962). *Theory of elasticity* (2nd ed.). Auckland: McGraw-Hill.

[89] Taylor, C. and Hughes, T., (1981). Finite Element Programming Of The Navier-Stokes Equations. Swansea: Pineridge Press.

Appendix I the description of library 'Math'

The subroutine library 'Math' is developed by JIADA TU, which contains 21 subroutines. The description of these subroutines is shown in Table AP-1.

Table AP-1 The description of the subroutine library 'Math'.

Name	Description
mpstress	This subroutine is used to generate the principle stress, (1st, 2nd, 3th).
KRHQ	This subroutine is used to implement the Q.Xu's creep constitutive equation for 0.6Cr0.6Mo0.26V ferritic steel at 690 °C. The integral method applies the Euler
KR	This subroutine is used to implement the creep constitutive equation of Katchanov-Robotnov. The material of this constitutive equation is: Bar267 at 660 °C
psigma	This subroutine is used to output the deviator stress tensor
execute	This subroutine is used to determine if the array has changed, if yes, the logical variable 'reform' set true, if not, set false.
KRH	This subroutine is used to implement the

	Hayhurst's creep constitutive equation for 0.6Cr0.6Mo0.26V ferritic steel at 690 °C. The integral method apply the Euler
RKKRH	This subroutine is used to implement the Hayhurst's creep constitutive equation for 0.6Cr0.6Mo0.26V ferritic steel at 690 °C. The integral method applies the 4th Runge-Kutta Method.
new_km	This subroutine returns the stiffness matrix of a 2D Goodman element. It uses to simulate the grain boundary. The element is made up of two linear elements and four nodal points.
beeg	This subroutine returns the [B] matrix for a 2D Goodman element.
deeg	This subroutine returns the [D] matrix for Goodman element
element_inf	This subroutine returns the element information: length and the angle from a 2D Goodman element.
Loc_Gol	This subroutine returns the angle transform matrix from a 2D Goodman element.
P_L	This subroutine is used to implement the creep constitutive equation of power law creep model. The

	material of this constitutive equation is: pure copper at 600 °C
gbvm	This subroutine is used to implement the creep constitutive equation of Markus Vose creep model for grain boundary.
G_shape_fun	This subroutine returns the values at the gauss point of a shape functions of single surface of a 3D Goodman element.
Gsample	This subroutine returns the local coordinates and weighting coefficients of the integrating points.
Gbeemat	This subroutine returns the [B] matrix of a 3D Goodman element.
Gdeemat	This subroutine returns the [D] matrix of a 3D Goodman element.
G_shape_der	This subroutine returns derivatives of shape functions of a 3D Goodman element.
TM_TD	This subroutine returns the transmission matrix of a 3D Goodman element.
T_coord	This subroutine returns the local node coordinates of a 3D Goodman element

Appendix II Source code of Library 'Math'

Module math

contains

SUBROUTINE

mpstress(theta, stress1, theta11, sstress, lode_theta, dsbar, sigm)

!-----

!This subroutine is used to generate the max principle stress, (1st,

! 2nd, 3th principle. This subroutine is written by researcher j.tu

! at University of Huddersfield

! 29/04/2017

!-----

!-----

! Input: (lode_theta:lode_angle;

! dsbar: von_mises_stress;

! sigm: mean_stress)

! Output: (theta(:):principal_stress_array;

! stress1 :First_principal_stress;

! theta11 :maximun_principal_stress)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

INTEGER::i

doubleprecision,INTENT(IN)::lode_theta,dsbar,sigm

doubleprecision,INTENT(OUT),OPTIONAL::theta(:),stress1,sstress,theta11

doubleprecision::flow1,flow2,flow3,two=2.0_iwp,theta1,theta2, &

theta3,pi=3.1415926,D3=3.0_iwp,loca(1),m,v

!-----To determine the max principle stress-----

flow1=sin(lode_theta-((two*pi)/(d3)))

flow2=sin(lode_theta)

flow3=sin(lode_theta+((two*pi)/(d3)))

theta1=sigm+(((two)/(d3))*flow1*dsbar)

theta2=sigm+(((two)/(d3))*flow2*dsbar)

theta3=sigm+(((two)/(d3))*flow3*dsbar)

theta(1)=theta1

theta(2)=theta2

theta(3)=theta3

! loca=maxloc(abs(theta))

! i=loca(1)

! m=theta(i)

! v=maxval(abs(theta))

```

!      theta11=sign(v,m)

      theta11=maxval(theta)

!      theta11=theta(1)

      sstress=(theta1**2+theta2**2+theta3**2)**(0.5)

      stress1=theta11-sigm

```

```

RETURN

```

```

      END subroutine mpstress

```

```

      SUBROUTINE KRHQ(orb,iav,cprop,theta,theta11,dsbar)

```

```

!-----

```

```

!This subroutine is used to implement the creep constitutive equation
!of Hayhurt's for 0.5Cr0.5Mo0.25V ferritic steel at 590 °C 。 The
!integral method apply the EULAR.

```

```

!                                                                 17/05/2018

```

```

!-----

```

```

!-----

```

```

! Input: (iav(:)  :Creep_status_parameters_array

```

```

!      cprop(:):creep_constitutive_equation_paramaters_array

```

```

!      theta(:):principal_stress_array

```

```

!          theta11 :maximun_principal_stress

!          dsbar   :von_mises_stress)

! Output: (orv(:) :output_creep_rates_array)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

Doubleprecision,intent(in)::iav(:),cprop(:),theta(:),theta11,dsbar

Doubleprecision,INTENT(OUT),OPTIONAL::orv(:)

Doubleprecision::A,B,C,h,hplus,kc,v,flow1,flow2,flow3,flow4,      &
flow5,flow6,flow7,d3=3.0_iwp,zero=0.0_iwp,esp,harder,voul,damage,  &
espt,ehard,erave,edate,a1,b1,p,q,stress1,stress2,stress3,mdsbar,  &
flow8,flow9,flow10,flow11,flow12,flow13,sdsbar,dsbar1,flow14

      esp=iav(1)

      harder=iav(2)

      voul=iav(3)

      damage=iav(4)

      A=cprop(1)

      B=cprop(2)

      C=cprop(3)

```


$h=cprop(4)$

$hplus=cprop(5)$

$kc=cprop(6)$

$a1=cprop(7)$

$b1=cprop(8)$

$p=cprop(9)$

$q=cprop(10)$

$stress1=theta(1)$

$stress2=theta(2)$

$stress3=theta(3)$

$mdsbar=(stress1+stress2+stress3)/d3$

$sdsbar=sqrt(stress1**2+stress2**2+stress3**2)$

$dsbar1=theta11-mdsbar$

$orv=zero$

!-----The strain rate Part-----

$flow1=B*dsbar*(1-harder)$

$flow2=((1-voul)*(1-damage))$

$flow3=((flow1)/(flow2))$

$$\text{espt} = A * (\sinh(\text{flow3}))$$

!-----The Harder rate Part-----

$$\text{flow4} = (1 - (\text{harder})) / (\text{hplus})$$

$$\text{flow5} = (\text{h}) / (\text{dsbar})$$

$$\text{ehard} = \text{flow4} * \text{flow5} * \text{espt}$$

!-----The Evoulation rate Part-----

$$\text{flow6} = \text{kc} / \text{d3}$$

$$\text{erave} = \text{flow6} * ((1 - \text{voul}) ** (4))$$

!-----The Damage rate Part-----

if($\theta_{11} > \text{zero}$)then

$$\text{flow7} = (1 - (\theta_{11}) / (\text{dsbar})) * \text{p}$$

$$\text{flow8} = (0.5 - (3 * \text{mdsbar}) / (2 * \text{dsbar})) * \text{q}$$

$$\text{flow9} = (\exp(\text{flow7} + \text{flow8})) ** (-1)$$

$$\text{flow10} = C * \text{espt} * \text{flow9}$$

$$\text{flow11} = (((3 * \text{mdsbar}) / (\text{sdsbar})) - 1) * \text{b1}$$

$$\text{flow12} = \exp(\text{flow11})$$

$$\text{flow13} = ((2 * \text{dsbar}) / (3 * \text{dsbar1})) ** \text{a1}$$

$$\text{flow14} = \text{flow12} * \text{flow13}$$

$$\text{edate} = \text{flow10} * \text{flow14}$$

else if($\theta_{11} \leq \text{zero}$)then

edate=zero

end if

orv(1)=espt

orv(2)=ehard

orv(3)=erave

orv(4)=edate

RETURN

END subroutine KRHQ

SUBROUTINE KR(espt,edate,damage,theta11,dsbar,t)

!-----

!This subroutine is used to implement the creep constitutive equation

!of Katchanov-Robotnov. The material of this constitutive equation is:

! Bar257 at 650 Celsius

!

10/01/2018

!-----

!-----

```

! Input: (theta11 :maximun_principal_stress;
!       damage  :creep_damage_value;
!       dsbar   :von_mises_stress;
!       t       :temperature_variable)
!       dsbar   :von_mises_stress)

! Output:(espt :creep_strain_rate;
!         edate:creep_damage_rate)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

Doubleprecision,INTENT(IN)::damage,theta11,dsbar,t

Doubleprecision,INTENT(OUT),OPTIONAL::espt,edate

Doubleprecision::a,q,x,dx,m,flow1,flow2,flow3,flow4,flow5,flow6,   &
flow7,flow8,n,xm

!-----the parameters of the constitutive equation-----

a=6.599e-16

q=4.5

x=0.3

dx=5.767

m=5.998e-14

```

n=6.108

xm=0

!-----

flow1=((1-damage)**(n))

flow2=(flow1)**(-1)

flow3=a*(dsbar**(n))

flow4=((t)**(xm))

espt=flow2*flow3*flow4

flow5=((1-damage)**(q))*(1+q)

flow6=((x*theta11)+((1-x)*dsbar))

flow7=(flow6**(dx))

edate=m*((flow7)/(flow5))

RETURN

END subroutine KR

SUBROUTINE psigma(stress,pstress)

```
!-----  
! This subroutine is used to output the Partial stress  
! component form  
!  
! 04/05/2017
```

```
!-----  
!-----  
! Input : ( stress(:) :stress_tenor)  
! Output: (pstress(:) :Stress_partial_tensor)  
!-----
```

```
IMPLICIT NONE
```

```
INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)
```

```
doubleprecision,INTENT(IN)::stress(:)
```

```
doubleprecision,INTENT(OUT)::pstress(:)
```

```
doubleprecision::sx,sy,sz,txy,sgm,s1,s2,s3,s4,s5,s6, &
```

```
three=3.0_iwp
```

```
INTEGER::nst
```

```
nst=UBOUND(stress,1)
```

```
SELECT CASE(nst)
```

```
CASE(3)
```

```
sx=stress(1)
```

```
sy=stress(2)
```

txy=stress(3)

sigm=(sx+sy)/three

pstress(1)=sx-sigm

pstress(2)=sy-sigm

pstress(3)=txy

CASE(4)

sx=stress(1)

sy=stress(2)

txy=stress(3)

sz=stress(4)

sigm=(sx+sy+sz)/three

pstress(1)=sx-sigm

pstress(2)=sy-sigm

pstress(3)=txy

pstress(4)=sz-sigm

!-----

! sigm=(sx+sy)/three

! pstress(1)=sx-sigm

! pstress(2)=sy-sigm

! pstress(3)=txy

! pstress(4)=-sigm

!-----

CASE(6)

s1=stress(1)

s2=stress(2)

s3=stress(3)

s4=stress(4)

s5=stress(5)

s6=stress(6)

sigm=(s1+s2+s3)/three

pstress(1)=s1-sigm

pstress(2)=s2-sigm

pstress(3)=s3-sigm

pstress(4)=s4

pstress(5)=s5

pstress(6)=s6

CASE DEFAULT

WRITE(*,*)"wrong size for nst in invar"


```

END SELECT

RETURN

    END SUBROUTINE psigma

    SUBROUTINE execute(olds,fal,reform)

!-----

! This subroutine sets reform to .FALSE. if relative change in when the

! fal array changed.

!-----

!-----

! Input : ( old(:) the_array_stores_the_initial_status_of_all_elements;

!         fal(:) the_array_stores_the_end_status_of_all_elements)

! Output: (reform: trigger_variable_for_removal_failed_element)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

INTEGER,INTENT(IN)::olds(:),fal(:)

LOGICAL,INTENT(OUT)::reform

reform=.true.

reform=((MAXVAL(ABS(fal-olds)))>0)

RETURN

```

END SUBROUTINE execute

SUBROUTINE KRH(orsv,iav,cprop,theta11,dsbar)

!-----

!This subroutine is used to implement the creep constitutive equation
!of Hayhurst's for 0.5Cr0.5Mo0.25V ferritic steel at 590 °C 。 The
!integral method apply the EULAR.

! 17/05/2018

!-----

!-----

! Input : (iav(:) : Creep_status_parameters_array;

! cprop(:): Creep_status_parameters_array;

! theta11 : maximun_principal_stress;

! dsbar : von_mises_stress)

! Output: (orsv : output_creep_rates_array)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

Doubleprecision,intent(in)::iav(:),cprop(:),theta11,dsbar

Doubleprecision,INTENT(OUT),OPTIONAL::orsv(:)

Doubleprecision::A,B,C,h,hplus,kc,v,flow1,flow2,flow3,flow4, &

flow5,flow6,flow7,d3=3.0_iwp,zero=0.0_iwp,esp,harder,voul,damage, &
espt,ehard,erave,edate

esp=iav(1)

harder=iav(2)

voul=iav(3)

damage=iav(4)

A=cprop(1)

B=cprop(2)

C=cprop(3)

h=cprop(4)

hplus=cprop(5)

kc=cprop(6)

v=cprop(7)

orv=zero

!-----The strain rate Part-----

flow1=B*dsbar*(1-harder)

flow2=((1-voul)*(1-damage))

$$\text{flow3} = (\text{flow1}) / (\text{flow2})$$

$$\text{espt} = A * (\sinh(\text{flow3}))$$

!-----The Harder rate Part-----

$$\text{flow4} = (1 - (\text{harder})) / (\text{hplus})$$

$$\text{flow5} = (\text{h}) / (\text{dsbar})$$

$$\text{ehard} = \text{flow4} * \text{flow5} * \text{espt}$$

!-----The Evoulation rate Part-----

$$\text{flow6} = \text{kc} / \text{d3}$$

$$\text{erave} = \text{flow6} * ((1 - \text{voul}) ** (4))$$

!-----The Damage rate Part-----

if($\theta_{11} > \text{zero}$)then

$$\text{flow7} = ((\theta_{11}) / (\text{dsbar})) ** (\text{v})$$

$$\text{edate} = C * \text{espt} * \text{flow7}$$

else if($\theta_{11} \leq \text{zero}$)then

$$\text{edate} = \text{zero}$$

end if

$$\text{orv}(1) = \text{espt}$$

$$\text{orv}(2) = \text{ehard}$$

$$\text{orv}(3) = \text{erave}$$

$$\text{orv}(4) = \text{edate}$$

RETURN

END subroutine KRH

SUBROUTINE RKKRH(orsv,iav,cprop,theta11,dsbar,dt)

!-----

!This subroutine is used to implement the integration method for the
!constitutive equations by 4th Rounge Kutta Method.

! 17/05/2018

!-----

!-----

! Input : (iav(:) : Creep_status_parameters_array;

! cprop(:): Creep_status_parameters_array;

! theta11 : maximun_principal_stress;

! dsbar : von_mises_stress

! dt : the_time_step_for_integration)

! Output: (orsv : output_creep_rates_array)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

Doubleprecision,intent(in)::iav(:),cprop(:),theta11,dsbar,dt

Doubleprecision,INTENT(OUT),OPTIONAL::orv(:)

Doubleprecision::zero=0.0_iwp,D3=3.0_iwp,A,B,C,damage,ehard,ehard1,

&

ehard2,ehard3,ehard4,ehard5,erave1,erave2,erave3,erave4,erave5,erave, &

esp,espt,flow1,flow2,flow3,flow4,flow5,flow6,H,harder,hplus,kc,v, &

voul,flow7,edate

esp=iav(1)

harder=iav(2)

voul=iav(3)

damage=iav(4)

!-----input the paramaters of the constitutive equation-----

A=cprop(1)

B=cprop(2)

C=cprop(3)

h=cprop(4)

hplus=cprop(5)

kc=cprop(6)

v=cprop(7)

!-----

orv=zero

!-----obtin the strain rate-----

$$\text{flow1} = B * \text{dsbar} * (1 - \text{harder})$$

$$\text{flow2} = ((1 - \text{voul}) * (1 - \text{damage}))$$

$$\text{flow3} = (\text{flow1}) / (\text{flow2})$$

$$\text{espt} = A * (\sinh(\text{flow3}))$$

!-----obtain the harder rate-----

!-----for ehard1-----

$$\text{flow4} = (1 - (\text{harder})) / (\text{hplus})$$

$$\text{flow5} = (\text{h}) / (\text{dsbar})$$

$$\text{ehard1} = \text{flow4} * \text{flow5} * \text{espt}$$

!-----for ehard 2-----

$$\text{harder} = \text{iav}(2)$$

$$\text{harder} = \text{harder} + \text{ehard1} * \text{dt} / 3$$

$$\text{flow4} = (1 - (\text{harder})) / (\text{hplus})$$

$$\text{flow5} = (\text{h}) / (\text{dsbar})$$

$$\text{ehard2} = \text{flow4} * \text{flow5} * \text{espt}$$

!-----for ehard 3-----

$$\text{harder} = \text{iav}(2)$$

$$\text{harder} = \text{harder} + (\text{ehard1} * \text{dt}) / (6) + (\text{ehard2} * \text{dt}) / (6)$$

$$\text{flow4} = (1 - (\text{harder})) / (\text{hplus})$$

$$\text{flow5}=(h)/(dsbar)$$

$$\text{ehard3}=\text{flow4}*\text{flow5}*espt$$

!-----for ehard 4-----

$$\text{harder}=\text{iav}(2)$$

$$\text{harder}=\text{harder}+(\text{ehard1}*dt)/(8)+(\text{ehard3}*dt^3)/(8)$$

$$\text{flow4}=(1-(\text{harder})/(\text{hplus}))$$

$$\text{flow5}=(h)/(dsbar)$$

$$\text{ehard4}=\text{flow4}*\text{flow5}*espt$$

!-----for ehard 5-----

$$\text{harder}=\text{iav}(2)$$

$$\text{harder}=\text{harder}+(\text{ehard1}*dt)/(2)-(\text{ehard3}*dt^3)/(2)+2*\text{ehard4}*dt$$

$$\text{flow4}=(1-(\text{harder})/(\text{hplus}))$$

$$\text{flow5}=(h)/(dsbar)$$

$$\text{ehard5}=\text{flow4}*\text{flow5}*espt$$

!-----for final ehard-----

$$\text{ehard}=(\text{ehard1}/6)+((2*\text{ehard4})/3)+(\text{ehard5}/6)$$

!-----

!-----

!-----

!-----obtain the Evoulation rate-----

!-----for erave1-----

$$\text{flow6}=\text{kc}/\text{d3}$$

$$\text{erave1}=\text{flow6}*((1-\text{voul})**(4))$$

!----- for erave2-----

$$\text{voul}=\text{iav}(3)$$

$$\text{voul}=\text{voul}+(\text{erave1}*\text{dt}/3)$$

$$\text{flow6}=\text{kc}/\text{d3}$$

$$\text{erave2}=\text{flow6}*((1-\text{voul})**(4))$$

!----- for erave3-----

$$\text{voul}=\text{iav}(3)$$

$$\text{voul}=\text{voul}+(\text{erave1}*\text{dt}/6)+(\text{erave2}*\text{dt}/6)$$

$$\text{flow6}=\text{kc}/\text{d3}$$

$$\text{erave3}=\text{flow6}*((1-\text{voul})**(4))$$

!----- for erave4-----

$$\text{voul}=\text{iav}(3)$$

$$\text{voul}=\text{voul}+(\text{erave1}*\text{dt}/8)+(\text{erave3}^3*\text{dt}/8)$$

$$\text{flow6}=\text{kc}/\text{d3}$$

$$\text{erave4}=\text{flow6}*((1-\text{voul})**(4))$$

!----- for erave5-----

$$\text{voul}=\text{iav}(3)$$

$$\text{voul}=\text{voul}+(\text{erave1}*\text{dt}/2)-(\text{erave3}^3*\text{dt}/2)+(2*\text{erave4}*\text{dt})$$

$$\text{flow6}=\text{kc}/\text{d3}$$

$$\text{erave5}=\text{flow6}*((1-\text{voul})**(4))$$

!----- the final erave-----

$$\text{erave}=(\text{erave1}/6)+((2*\text{erave4})/3)+(\text{erave5}/6)$$

!-----obtin the damage rate-----

if(theta11>zero)then

$$\text{flow7}=(\text{theta11}/(\text{dsbar}))^{**}(\text{v})$$

$$\text{edate}=\text{C}*\text{espt}*\text{flow7}$$

else if(theta11<=zero)then

$$\text{edate}=\text{zero}$$

end if

$$\text{orv}(1)=\text{espt}$$

$$\text{orv}(2)=\text{ehard}$$

$$\text{orv}(3)=\text{erave}$$

$$\text{orv}(4)=\text{edate}$$

END SUBROUTINE RKKRH

```

SUBROUTINE new_km(km,kcoh)

!-----

! This subroutine forms the stiffness matrix of a joint element.

! It use to simulate the grain bounday.

! The element is made up of two linear elements and four nodal point.

! The mathmatical background is based on these two articles.

![1]Goodman RE, Taylor RL, Brekke TI, A model for the mechanics of
!jointed rock, J. Soil Mech, Found. Div, ASCE, 1968, 94 (SM3):637~659.

![2] Шамровский, А. and Богданова, Е. (2014). Solution of contact
!problems of elasticity theory using a discrete finite-size element.
! Eastern-European Journal of Enterprise Technologies, 3(7(69), p.41.i

! The code is writied by J.Tu. Research of Huddersfield University.

!
!
!-----
!-----
!-----
!-----
! Input : (kcoh(:): Goodman_element_elastic_parameters_array)
! Output: (km(:) : output_Goodman_element_stiffness_matrix)
!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

REAL(iwp),INTENT(IN):: kcoh(:)

REAL(iwp),INTENT(OUT)::km(:,:)

```

REAL(iwp)::one=1.0_iwp,six=6.0_iwp,ZERO=0.0_iwp,two=2.0_iwp,KS,KN,A

KN=KCOH(1)

KS=KCOH(2)

A=ONE/SIX

KM=ZERO

KM(1,1)=TWO*KS

KM(1,2)=ZERO

KM(1,3)=(TWO*KS)*(-1)

KM(1,4)=ZERO

KM(1,5)=KS*(-1)

KM(1,6)=ZERO

KM(1,7)=KS

KM(1,8)=ZERO

KM(2,1)=ZERO

KM(2,2)=TWO*KN

KM(2,3)=ZERO

KM(2,4)=(TWO*KN)*(-1)

KM(2,5)=ZERO

$$\text{KM}(2,6)=\text{KN}*(-1)$$

$$\text{KM}(2,7)=\text{ZERO}$$

$$\text{KM}(2,8)=\text{KN}$$

$$\text{KM}(3,1)=(\text{TWO}*\text{KS})*(-1)$$

$$\text{KM}(3,2)=\text{ZERO}$$

$$\text{KM}(3,3)=\text{TWO}*\text{KS}$$

$$\text{KM}(3,4)=\text{ZERO}$$

$$\text{KM}(3,5)=\text{KS}$$

$$\text{KM}(3,6)=\text{ZERO}$$

$$\text{KM}(3,7)=\text{KS}*(-1)$$

$$\text{KM}(3,8)=\text{ZERO}$$

$$\text{KM}(4,1)=\text{ZERO}$$

$$\text{KM}(4,2)=(\text{TWO}*\text{KN})*(-1)$$

$$\text{KM}(4,3)=\text{ZERO}$$

$$\text{KM}(4,4)=\text{TWO}*\text{KN}$$

$$\text{KM}(4,5)=\text{ZERO}$$

$$\text{KM}(4,6)=\text{KN}$$

$$\text{KM}(4,7)=\text{ZERO}$$

$$\text{KM}(4,8)=\text{KN}*(-1)$$

$$\text{KM}(5,1)=\text{KS}*(-1)$$

$$\text{KM}(5,2)=\text{ZERO}$$

$$\text{KM}(5,3)=\text{KS}$$

$$\text{KM}(5,4)=\text{ZERO}$$

$$\text{KM}(5,5)=\text{TWO}*\text{KS}$$

$$\text{KM}(5,6)=\text{ZERO}$$

$$\text{KM}(5,7)=(\text{TWO}*\text{KS})*(-1)$$

$$\text{KM}(5,8)=\text{ZERO}$$

$$\text{KM}(6,1)=\text{ZERO}$$

$$\text{KM}(6,2)=\text{KN}*(-1)$$

$$\text{KM}(6,3)=\text{ZERO}$$

$$\text{KM}(6,4)=\text{KN}$$

$$\text{KM}(6,5)=\text{ZERO}$$

$$\text{KM}(6,6)=\text{TWO}*\text{KN}$$

$$\text{KM}(6,7)=\text{ZERO}$$

$$\text{KM}(6,8)=(\text{TWO}*\text{KN})*(-1)$$

$$\text{KM}(7,1)=\text{KS}$$

$$\text{KM}(7,2)=\text{ZERO}$$

$$\text{KM}(7,3)=\text{KS}*(-1)$$

$$\text{KM}(7,4)=\text{ZERO}$$

$$\text{KM}(7,5)=(\text{TWO}*\text{KS})*(-1)$$

$$\text{KM}(7,6)=\text{ZERO}$$

$$\text{KM}(7,7)=\text{TWO}*\text{KS}$$

$$\text{KM}(7,8)=\text{ZERO}$$

$$\text{KM}(8,1)=\text{ZERO}$$

KM(8,2)=KN

KM(8,3)=ZERO

KM(8,4)=KN*(-1)

KM(8,5)=ZERO

KM(8,6)=(TWO*KN)*(-1)

KM(8,7)=ZERO

KM(8,8)=TWO*KN

KM=A*KM

RETURN

END SUBROUTINE new_km

SUBROUTINE beeg(bee)

!-----

! This subroutine forms the bee matrix for goodman element

!-----

!-----

! Output: (bee(:) : output_Goodman_element_B_matrix)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

REAL(iwp),INTENT(OUT)::bee(:,:)

REAL::zero=0.0_iwp,one=1.0_iwp

bee=zero

bee(1,1)=-one

bee(2,2)=-one

bee(1,3)=one

bee(2,4)=one

bee(1,5)=one

bee(2,6)=one

bee(1,7)=-one

bee(2,8)=-one

bee=0.5*bee

RETURN

END SUBROUTINE beeg


```

SUBROUTINE deeg(dee,kcoh)

!-----
! This subroutine returns the elastic dee matrix for goodman element
!-----
!-----
! Input : (kcoh(:) : Goodman_element_elastic_parameters_array)
! Output: (dee(:) : output_Goodman_element_dee_matrix)
!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

REAL(iwp),INTENT(IN)::kcoh(:)

REAL(iwp),INTENT(OUT)::dee(:,:)

REAL(iwp)::KN,KS,zero=0.0_iwp

dee=zero

KN=KCOH(1)

KS=KCOH(2)

dee(1,1)=KS

dee(2,2)=KN

RETURN

```

END SUBROUTINE deeg

SUBROUTINE element_inf(length,angle,coord)

!-----

! This subroutine forms the length and the angle for
! the goodman element

! Research:J.Tu

! The University of Huddersfield

! 06/09/2018

!-----

!-----

! Input : (coord(:) : Goodman_element_node_coordinate_array)

! Output: (length : 2d_Goodman_element_length;

! angle(:) : array_stores_the_sine&cosine_of_local_to_global_

! _coordinate_system_angle)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

REAL(iwp),INTENT(IN)::coord(:,:)

REAL(iwp),INTENT(OUT)::length,angle(:)

REAL::zero=0.0_iwp,one=1.0_iwp,two=2.0_iwp,x1,x2,x3,x4,y1,&

y2,y3,y4,angle1,angle2,length1,length2,a1,b1,a2,b2

!-----inport the coordinate for each node-----

x1=coord(1,1)

y1=coord(1,2)

x2=coord(2,1)

y2=coord(2,2)

x3=coord(3,1)

y3=coord(3,2)

x4=coord(4,1)

y4=coord(4,2)

!-----for upper surface-----

length1=sqrt((x2-x3)**(two)+(y2-y3)**(two))

a1=(y3-y2)/length1

b1=(x3-x2)/length1

!-----for bottom surface-----

length1=sqrt((x1-x4)**(two)+(y1-y4)**(two))

a2=(y4-y1)/length2

b2=(x4-x1)/length2

length=length1

angle(1)=a1

angle(2)=b1

RETURN

END SUBROUTINE element_inf

SUBROUTINE Loc_Gol(angle_t,angle)

!-----

! This subroutine is used to obtain the angle transform

! matrix for goodman element.

! Research:J.Tu

! The University of Huddersfield

! 08/09/2018

!-----

!-----

! Input : (angle(:) : array_stores_the_sine&cosine_of_local_to_global_

! _coordinate_system_angle)

! Output: (angle(:) : local_global_coordinate_transformation_matrix)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

```

REAL(iwp),INTENT(IN)::angle(:)

REAL(iwp),INTENT(OUT)::angle_t(:,:)

INTEGER::i

REAL::zero=0.0_iwp,one=1.0_iwp,two=2.0_iwp,sina,cosa

!----inport the trigonometric value for each element-----

sina=angle(1)

cosa=angle(2)

!-----

angle_t=zero

Do i=1,8

    angle_t(i,i)=cosa

End do

angle_t(1,2)=sina

angle_t(2,1)=-sina

angle_t(3,4)=sina

angle_t(4,3)=-sina

angle_t(5,6)=sina

angle_t(6,5)=-sina

angle_t(7,8)=sina

angle_t(8,7)=-sina

```

RETURN

END SUBROUTINE Loc_Gol

SUBROUTINE P_L(espt,cprop,dsbar,t)

!-----

!This subroutine is used to implement the creep constitutive equation
!of power law creep model. The material of this constitutive equation
!is: pure copper at 500 Celsius

! 15/10/2018

!-----

!-----

! Input: (cprop(:):creep_constitutive_equation_paramaters_array

! theta11 :maximun_principal_stress

! dsbar : von_mises_stress

! t :temperature_variable)

! Output: (espt(:) :output_creep_strain_rate)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

Doubleprecision,INTENT(IN)::cprop(:),dsbar,t

Doubleprecision,INTENT(OUT),OPTIONAL::espt

Doubleprecision::A,Q,N,M,Tem,R,K,flow1,flow2,flow3,flow4,flow5

!-----the parameters of the constitutive equation-----

A=cprop(1)

Q=CPROP(2)

N=CPROP(3)

M=CPROP(4)

Tem=CPROP(5)

R=8.314

K=Tem+273.15

!-----the power law constitutive equation-----

!The constitutive equation is form the publication:

!G. LI, B.G. THOMAS, and J.F. STUBBINS (10/2000)

!'Modeling Creep and Fatigue of Copper Alloys',

!METALLURGICAL AND MATERIALS TRANSACTIONS A, 31(10), pp. 2491

- 2502

!-----

```
flow1=(-Q)/(R*K)

flow2=exp(flow1)

flow3=A*flow2

flow4=(dsbar**(n))

flow5=((t**(m)))

espt=flow3*flow4*flow5
```

RETURN

END subroutine P_L

SUBROUTINE gbvm(dp,db,w,gcprop,p,b)

!-----

!This subroutine is used to implement the creep constitutive equation
!of Markus Vose creep model for grain boundary. The reference for this
!model is :

! Vöse, M., Fedelich, B. and Owen, J. (2012). A simplified model for
! creep induced grain boundary cavitation validated by multiple cavity
! growth simulations. Computational Materials Science, 58, pp.201-213.

!

18/10/2018


```

!-----
!-----
! Input: (gcprop(:):grain_boundary_creep_constitutive_equation
!
!         _paramaters_array;
!
!         p   :input_grain_boundary_creep_cavity_density;
!
!         b   :input_grain_boundary_creep_damage_variable)
!Output: (  dp   :output_grain_boundary_creep_cavity_density_rate;
!
!         db   :output_grain_boundary_creep_damage_variable_rate;
!
!         w    :output_grain_boundary_creep_damage)
!-----

```

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

Doubleprecision,INTENT(IN)::gcprop(:),p,b

Doubleprecision,INTENT(OUT),OPTIONAL::dp,db,w

Doubleprecision::a1,dgb1,rs,xp,r,theta,x1,x2,x3,x4,theta1,flow1, &

flow2,flow3,flow4,flow5,flow6,flow7,y,xa,up,stip0,qw0,pi=3.1415926, &

one=1.0_iwp,two=2.0_iwp,d3=3.0_iwp,flow8,flow9,htheta,flow10,flow11, &

flow12,flow13,flow14,f,dup,dgb,da,a

!-----input the creep parameters for the constitutive equation-----

a1=gcprop(1)

dgb1=gcprop(2)

rs=gcprop(3)

xp=gcprop(4)

r=gcprop(5)

theta=gcprop(6)

x1=gcprop(7)

x2=gcprop(8)

x3=gcprop(9)

x4=gcprop(10)

!----- prepare for simulation-----

theta1=(theta*pi)/180

flow1=((1+cos(theta1))**(-1))-(cos(theta1))/2

htheta=flow1/sin(theta1)

up=(4*htheta*(a1**3))/(3*(r**2))

dgb=dgb1**3

!-----initialization for the simulation-----

!-----the constitutive equation part-----

!-----equation 60-----

$$\text{flow1}=(3*b)/(4*h\theta*\pi)$$

$$a=(1/(\text{sqrt}(p)))*(\text{flow1}^{(1/d3)})$$

$$w=((9*\pi*(b^2))/(16*(h\theta^2)))^{(1/d3)}$$

!-----equation 59-----

$$\text{stip0}=(2*r_s*\sin(\theta_1))/a$$

$$q_w0=(-2)*d\log(w*x^2)-((3-w*x^2)*(1-w*x^2))$$

$$\text{flow2}=1-\text{stip0}*(1-x^2*w)$$

$$\text{flow3}=(a^2)*q_w0$$

$$da=(x^2*dgb*\text{flow2})/(h\theta*\text{flow3})$$

!-----equation 58-----

$$x_a=x^3*8*\pi*(p^2)*a*da$$

!-----equation 62-----

$$\text{flow4}=(b*(xp-xa))/(\text{sqrt}(p**3))$$

$$\text{flow10}=36*h\theta*\pi$$

$$\text{flow11}=b**2$$

$$\text{flow12}=\text{flow10}*\text{flow11}$$

$$\text{flow13}=\text{one}/d3$$

$$\text{flow5}=(\text{flow12})**(\text{flow13})*da$$

$$\text{dup}=\text{flow4}+\text{flow5}$$

!-----debug to here-----

!-----equation 61-----

$$\text{flow14}=-\text{one}$$

$$\text{flow6}=x4**2*\pi*dgb*p*((\text{dup})**(\text{flow14}))$$

$$\text{flow7}=(2*rs*\sin(\theta1))-\text{stip0}$$

$$y=\exp(\text{flow6}*\text{flow7})$$

$$f=((y-1)*w)/(1-w)$$

!-----equation 57-----

dp=xp*(1-f)-xa

flow8=(3*b*(xp-xa))/(2*p)

flow9=(sqrt(p))*((36*theta*(b**2)*pi)**(flow13))*da

db=flow8+flow9

RETURN

END subroutine gbvm

SUBROUTINE G_shape_fun(fun,points,i)

!-----

! This subroutine computes the values of the shape functions

! of single surface of 3D Goodman element.

! to local coordinates

! Developed by Researcher:J.TU

! The University of Huddersfield

! 21/09/2019

!-----

!-----

! Input: (i : Gaussian_point_number;

```

!           point(:,:) : array_stores_sampling_function)
! Output: ( fun(:) :array_stores_shape_function)
!-----
IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

INTEGER,INTENT(in)::i

REAL(iwp),INTENT(IN)::points(:,)

REAL(iwp),INTENT(OUT)::fun(:)

REAL(iwp)::eta,xi,etam,etap,xim,xip,zetam,zetap,c1,c2,c3

REAL(iwp)::t1,t2,t3,t4,t5,t6,t7,t8,t9

INTEGER::l,ndim,nod

REAL,PARAMETER::pt125=0.125_iwp,pt25=0.25_iwp,pt5=0.5_iwp,pt75=0.75_i
wp, &

one=1.0_iwp,two=2.0_iwp,d3=3.0_iwp,d4=4.0_iwp,d8=8.0_iwp,d9=9.0_iwp,
&

d16=16.0_iwp,d27=27.0_iwp,d32=32.0_iwp,d64=64.0_iwp,d128=128.0_iwp

nod=UBOUND(fun,1)

```

c1=points(i,1)

c2=points(i,2)

c3=one-c1-c2

xi=points(i,1)

eta=points(i,2)

etam=pt25*(one-eta)

etap=pt25*(one+eta)

xim=pt25*(one-xi)

xip=pt25*(one+xi)

SELECT CASE(nod)

CASE(3)

fun = (/c1,c3,c2/)

CASE(6)

fun(1)=(two*c1-one)*c1

fun(2)=d4*c3*c1

fun(3)=(two*c3-one)*c3

fun(4)=d4*c2*c3

fun(5)=(two*c2-one)*c2

fun(6)=d4*c1*c2

CASE(10)

$$\text{fun}(1) = ((d3*c1-one)*(d3*c1-two)*c1)/two$$

$$\text{fun}(2) = -(d9*(d3*c1-one)*(c1+c2-one)*c1)/two$$

$$\text{fun}(3) = (d9*(d3*c1+d3*c2-two)*(c1+c2-one)*c1)/two$$

$$\text{fun}(4) = -((d3*c1+d3*c2-one)*(d3*c1+d3*c2-two)*(c1+c2-one))/two$$

$$\text{fun}(5) = (d9*(d3*c1+d3*c2-two)*(c1+c2-one)*c2)/two$$

$$\text{fun}(6) = -(d9*(c1+c2-one)*(d3*c2-one)*c2)/two$$

$$\text{fun}(7) = ((d3*c2-one)*(d3*c2-two)*c2)/two$$

$$\text{fun}(8) = (d9*(d3*c2-one)*c1*c2)/two$$

$$\text{fun}(9) = (d9*(d3*c1-one)*c1*c2)/two$$

$$\text{fun}(10) = -d27*((c2-one)+c1)*c1*c2$$

CASE(15)

$$t1 = c1-pt25$$

$$t2 = c1-pt5$$

$$t3 = c1-pt75$$

$$t4 = c2-pt25$$

$$t5 = c2-pt5$$

$$t6 = c2-pt75$$

$$t7 = c3-pt25$$

$$t8 = c3-pt5$$

$$t9 = c3-pt75$$

$$\text{fun}(1) = d32/d3*c1*t1*t2*t3$$

$$\text{fun}(2)=d128/d3*c3*c1*t1*t2$$

$$\text{fun}(3)=d64*c3*c1*t1*t7$$

$$\text{fun}(4)=d128/d3*c3*c1*t7*t8$$

$$\text{fun}(5)=d32/d3*c3*t7*t8*t9$$

$$\text{fun}(6)=d128/d3*c2*c3*t7*t8$$

$$\text{fun}(7)=d64*c2*c3*t4*t7$$

$$\text{fun}(8)=d128/d3*c2*c3*t4*t5$$

$$\text{fun}(9)=d32/d3*c2*t4*t5*t6$$

$$\text{fun}(10)=d128/d3*c1*c2*t4*t5$$

$$\text{fun}(11)=d64*c1*c2*t1*t4$$

$$\text{fun}(12)=d128/d3*c1*c2*t1*t2$$

$$\text{fun}(13)=d128*c1*c2*t1*c3$$

$$\text{fun}(15)=d128*c1*c2*c3*t4$$

$$\text{fun}(14)=d128*c1*c2*c3*t7$$

CASE(4)

$$\text{fun}=(/d4*xim*etam,d4*xim*etap,d4*xip*etap,d4*xip*etam/)$$

CASE(5)

$$\begin{aligned} \text{fun}=(/d4*xim*etam-pt25*(one-xi**2)*(one-eta**2), & \& \\ d4*xim*etap-pt25*(one-xi**2)*(one-eta**2), & \& \\ d4*xip*etap-pt25*(one-xi**2)*(one-eta**2), & \& \\ d4*xip*etam-pt25*(one-xi**2)*(one-eta**2), & \& \\ (one-xi**2)*(one-eta**2)/) \end{aligned}$$

CASE(8)

```
fun=(/d4*etam*xim*(-xi-eta-one),d32*etam*xim*etap,  
&  
d4*etap*xim*(-xi+eta-one),d32*xim*xip*etap,  
&  
d4*etap*xip*(xi+eta-one), d32*etap*xip*etam,  
&  
d4*xip*etam*(xi-eta-one), d32*xim*xip*etam/)
```

CASE(9)

```
etam=eta-one  
etap=eta+one  
xim=xi-one  
xip=xi+one  
fun=(/pt25*xi*xim*eta*etam,-pt5*xi*xim*etap*etam,  
&  
pt25*xi*xim*eta*etap,-pt5*xip*xim*eta*etap,  
&  
pt25*xi*xip*eta*etap,-pt5*xi*xip*etap*etam,  
&  
pt25*xi*xip*eta*etam,-pt5*xip*xim*eta*etam,  
&  
xip*xim*etap*etam/)
```

CASE DEFAULT

WRITE(*,*)"wrong number of nodes in shape_fun"

END SELECT

RETURN

END SUBROUTINE G_shape_fun

SUBROUTINE Gsample(element,s,wt)

!-----

! This subroutine returns the local coordinates and weighting coefficients
! of the integrating points.

!-----

!-----

! Input:(element: the_shape_of_3D_Goodman_single_surface;

! s(:,:) : the_coordinates_of_Gaussian_point_at_local_coordinate

! _system)

!Output:(wt(:) : the_wights_of_Gaussion_points_in_Gauss_Legendre

! _quadrilateral_integration)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

```

REAL(iwp),INTENT(OUT)::s(:,:)
REAL(iwp),INTENT(OUT),OPTIONAL::wt(:)
CHARACTER(*),INTENT(IN)::element
INTEGER::nip
REAL(iwp)::root3,r15,w(3),v(9),b,c
root3=1.0_iwp/SQRT(3.0_iwp)
r15=0.2_iwp*SQRT(15.0_iwp)
nip=UBOUND(s,1)
w=(/5.0_iwp/9.0_iwp,8.0_iwp/9.0_iwp,5.0_iwp/9.0_iwp/)
v=(/5.0_iwp/9.0_iwp*w,8.0_iwp/9.0_iwp*w,5.0_iwp/9.0_iwp*w/)
SELECT CASE(element)
CASE('T')
SELECT CASE(nip)
CASE(1)
s(1,1)= 0.3333333333333333_iwp
s(1,2)= 0.3333333333333333_iwp
wt(1) = 0.5000000000000000_iwp
CASE(3)
s(1,1)= 0.5000000000000000_iwp
s(1,2)= 0.5000000000000000_iwp
s(2,1)= 0.5000000000000000_iwp
s(2,2)= 0.0000000000000000_iwp

```

s(3,1)= 0.0000000000000000_iwp
s(3,2)= 0.5000000000000000_iwp
wt(1:3)=0.3333333333333333_iwp
wt=0.5_iwp*wt

CASE(4)

s(1,1)= 0.6_iwp
s(1,2)= 0.2_iwp
s(2,1)= 0.2_iwp
s(2,2)= 0.6_iwp
s(3,1)= 0.2_iwp
s(3,2)= 0.2_iwp
s(4,1)= 0.3333333333333333_iwp
s(4,2)= 0.3333333333333333_iwp
wt(1:3)= 0.5208333333333333_iwp
wt(4)= -0.5625_iwp
wt=0.5_iwp*wt

CASE(6)

s(1,1)= 0.816847572980459_iwp
s(1,2)= 0.091576213509771_iwp
s(2,1)= 0.091576213509771_iwp
s(2,2)= 0.816847572980459_iwp
s(3,1)= 0.091576213509771_iwp

s(3,2)= 0.091576213509771_iwp
s(4,1)= 0.108103018168070_iwp
s(4,2)= 0.445948490915965_iwp
s(5,1)= 0.445948490915965_iwp
s(5,2)= 0.108103018168070_iwp
s(6,1)= 0.445948490915965_iwp
s(6,2)= 0.445948490915965_iwp
wt(1:3)=0.109951743655322_iwp
wt(4:6)=0.223381589678011_iwp
wt=0.5_iwp*wt

CASE(7)

s(1,1)= 0.333333333333333_iwp
s(1,2)= 0.333333333333333_iwp
s(2,1)= 0.797426985353087_iwp
s(2,2)= 0.101286507323456_iwp
s(3,1)= 0.101286507323456_iwp
s(3,2)= 0.797426985353087_iwp
s(4,1)= 0.101286507323456_iwp
s(4,2)= 0.101286507323456_iwp
s(5,1)= 0.470142064105115_iwp
s(5,2)= 0.059715871789770_iwp
s(6,1)= 0.059715871789770_iwp

s(6,2)= 0.470142064105115_iwp
s(7,1)= 0.470142064105115_iwp
s(7,2)= 0.470142064105115_iwp
wt(1) = 0.2250000000000000_iwp
wt(2:4)=0.125939180544827_iwp
wt(5:7)=0.132394152788506_iwp
wt=0.5_iwp*wt

CASE(12)

s(1,1)= 0.873821971016996_iwp
s(1,2)= 0.063089014491502_iwp
s(2,1)= 0.063089014491502_iwp
s(2,2)= 0.873821971016996_iwp
s(3,1)= 0.063089014491502_iwp
s(3,2)= 0.063089014491502_iwp
s(4,1)= 0.501426509658179_iwp
s(4,2)= 0.249286745170910_iwp
s(5,1)= 0.249286745170910_iwp
s(5,2)= 0.501426509658179_iwp
s(6,1)= 0.249286745170910_iwp
s(6,2)= 0.249286745170910_iwp
s(7,1) =0.053145049844817_iwp
s(7,2) =0.310352451033784_iwp

s(8,1)=0.310352451033784_iwp
s(8,2)=0.053145049844817_iwp
s(9,1)=0.053145049844817_iwp
s(9,2)=0.636502499121398_iwp
s(10,1)=0.310352451033784_iwp
s(10,2)=0.636502499121398_iwp
s(11,1)=0.636502499121398_iwp
s(11,2)=0.053145049844817_iwp
s(12,1)=0.636502499121398_iwp
s(12,2)=0.310352451033784_iwp
wt(1:3)=0.050844906370207_iwp
wt(4:6)=0.116786275726379_iwp
wt(7:12)=0.082851075618374_iwp
wt=0.5_iwp*wt

CASE(16)

s(1,1)=0.3333333333333333_iwp
s(1,2)=0.3333333333333333_iwp
s(2,1)=0.658861384496478_iwp
s(2,2)=0.170569307751761_iwp
s(3,1)=0.170569307751761_iwp
s(3,2)=0.658861384496478_iwp
s(4,1)=0.170569307751761_iwp

s(4,2)=0.170569307751761_iwp
s(5,1)=0.898905543365938_iwp
s(5,2)=0.050547228317031_iwp
s(6,1)=0.050547228317031_iwp
s(6,2)=0.898905543365938_iwp
s(7,1)=0.050547228317031_iwp
s(7,2)=0.050547228317031_iwp
s(8,1)=0.081414823414554_iwp
s(8,2)=0.459292588292723_iwp
s(9,1)=0.459292588292723_iwp
s(9,2)=0.081414823414554_iwp
s(10,1)=0.459292588292723_iwp
s(10,2)=0.459292588292723_iwp
s(11,1)=0.008394777409958_iwp
s(11,2)=0.263112829634638_iwp
s(12,1)=0.008394777409958_iwp
s(12,2)=0.728492392955404_iwp
s(13,1)=0.263112829634638_iwp
s(13,2)=0.008394777409958_iwp
s(14,1)=0.263112829634638_iwp
s(14,2)=0.728492392955404_iwp
s(15,1)=0.728492392955404_iwp

s(15,2)=0.008394777409958_iwp
s(16,1)=0.728492392955404_iwp
s(16,2)=0.263112829634638_iwp
wt(1)=0.144315607677787_iwp
wt(2:4)=0.103217370534718_iwp
wt(5:7)=0.032458497623198_iwp
wt(8:10)=0.095091634267284_iwp
wt(11:16)=0.027230314174435_iwp

wt=0.5_iwp*wt

CASE DEFAULT

WRITE(*,*)"wrong number of integrating points for a triangle"

END SELECT

CASE('Q')

SELECT CASE(nip)

CASE(1)

s(1,1)=0.0_iwp

s(1,2)=0.0_iwp

wt(1)=4.0_iwp

CASE(4)

s(1,1)=-root3

s(1,2)= root3

s(2,1)= root3

$$s(2,2)= \text{root3}$$

$$s(3,1)=-\text{root3}$$

$$s(3,2)=-\text{root3}$$

$$s(4,1)= \text{root3}$$

$$s(4,2)=-\text{root3}$$

$$\text{wt}=1.0_iwp$$

CASE(9)

$$s(1:7:3,1)=-r15$$

$$s(2:8:3,1)=0.0_iwp$$

$$s(3:9:3,1)=r15$$

$$s(1:3,2) =r15$$

$$s(4:6,2) =0.0_iwp$$

$$s(7:9,2) =-r15$$

$$\text{wt}= v$$

CASE(16)

$$s(1:13:4,1)=-0.861136311594053_iwp$$

$$s(2:14:4,1)=-0.339981043584856_iwp$$

$$s(3:15:4,1)= 0.339981043584856_iwp$$

$$s(4:16:4,1)= 0.861136311594053_iwp$$

$$s(1:4,2) = 0.861136311594053_iwp$$

$$s(5:8,2) = 0.339981043584856_iwp$$

$$s(9:12,2) =-0.339981043584856_iwp$$

s(13:16,2)=-0.861136311594053_iwp
wt(1) = 0.121002993285602_iwp
wt(4) = wt(1)
wt(13) = wt(1)
wt(16) = wt(1)
wt(2) = 0.226851851851852_iwp
wt(3) = wt(2)
wt(5) = wt(2)
wt(8) = wt(2)
wt(9) = wt(2)
wt(12) = wt(2)
wt(14) = wt(2)
wt(15) = wt(2)
wt(6) = 0.425293303010694_iwp
wt(7) = wt(6)
wt(10) = wt(6)
wt(11) = wt(6)

CASE(25)

s(1:21:5,1)= 0.906179845938664_iwp
s(2:22:5,1)= 0.538469310105683_iwp
s(3:23:5,1)= 0.0_iwp
s(4:24:5,1)=-0.538469310105683_iwp

s(5:25:5,1)=-0.906179845938664_iwp
s(1: 5,2) = 0.906179845938664_iwp
s(6:10,2) = 0.538469310105683_iwp
s(11:15,2) = 0.0_iwp
s(16:20,2) =-0.538469310105683_iwp
s(21:25,2) =-0.906179845938664_iwp
wt(1) =0.056134348862429_iwp
wt(2) =0.113400000000000_iwp
wt(3) =0.134785072387521_iwp
wt(4) =0.113400000000000_iwp
wt(5) =0.056134348862429_iwp
wt(6) =0.113400000000000_iwp
wt(7) =0.229085404223991_iwp
wt(8) =0.272286532550750_iwp
wt(9) =0.229085404223991_iwp
wt(10)=0.113400000000000_iwp
wt(11)=0.134785072387521_iwp
wt(12)=0.272286532550750_iwp
wt(13)=0.323634567901235_iwp
wt(14)=0.272286532550750_iwp
wt(15)=0.134785072387521_iwp
wt(16)=0.113400000000000_iwp

wt(17)=0.229085404223991_iwp

wt(18)=0.272286532550750_iwp

wt(19)=0.229085404223991_iwp

wt(20)=0.113400000000000_iwp

wt(21)=0.056134348862429_iwp

wt(22)=0.113400000000000_iwp

wt(23)=0.134785072387521_iwp

wt(24)=0.113400000000000_iwp

wt(25)=0.056134348862429_iwp

CASE DEFAULT

WRITE(*,*)"wrong number of integrating points for a quadrilateral"

END SELECT

CASE DEFAULT

WRITE(*,*)"not a valid element type"

END SELECT

RETURN

END SUBROUTINE Gsample

SUBROUTINE Gbeemat(bee,fun)

!-----

! This subroutine forms the bee matrix for 3D Goodman element.

! Developer:

J.TU

! The University of

Huddersfield

!

21/09/2019

!-----

!-----

! Input:(fun(:): the_shape_function_of_3D_Goodman_single_surface)

!Output:(bee(:): the_bee_matrix_of_3D_Goodman)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

REAL(iwp),INTENT(IN)::fun(:)

REAL(iwp),INTENT(OUT)::bee(:,:)

INTEGER::i,h

REAL::N1,N2,N3,N4,N5,N6,N7,N8

N1=fun(1)

N2=fun(2)

N3=fun(3)

N4=fun(4)

N5=-N1

N6=-N2

N7=-N3

N8=-N4

bee=0.0_iwp

do i=1,3

h=i

bee(i,h)=N1

end do

do i=1,3

h=i+3

bee(i,h)=N2

end do

do i=1,3


```
h=i+6  
bee(i,h)=N3  
end do
```

```
do i=1,3  
h=i+9  
bee(i,h)=N4  
end do
```

```
do i=1,3  
h=i+12  
bee(i,h)=N5  
end do
```

```
do i=1,3  
h=i+15  
bee(i,h)=N6  
end do
```

```
do i=1,3  
h=i+18  
bee(i,h)=N7
```

end do

do i=1,3

h=i+21

bee(i,h)=N8

end do

RETURN

END SUBROUTINE Gbeemat

SUBROUTINE Gdeemat(dee,kcoh)

!-----

! This subroutine returns the elastic dee matrix for 3D Goodman Element,

! it is a kind of cohesive element.

! developer:

J.Tu

! The University of

Huddersfield

!

22/09/2019

!-----

!-----
! Input:(kcoh: 3D_Goodman_element_elastic_parameters_array)

!Output:(dee(:,,:): the_dee_matrix_of_3D_Goodman)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

REAL(iwp),INTENT(IN)::kcoh(:)

REAL(iwp),INTENT(OUT)::dee(:,,:)

REAL(iwp)::KN,KS1,KS2,zero=0.0_iwp

dee=zero

KN=KCOH(1)

KS1=KCOH(2)

KS2=KCOH(3)

dee(1,1)=KS1

dee(2,2)=KS2

dee(3,3)=KN

RETURN

END SUBROUTINE Gdeemat

SUBROUTINE G_shape_der(der,points,i)

```

!-----
!   This subroutine produces derivatives of shape functions of 3D Goodman
!   element.
!
!                                     Developed by
Researcher:J.TU
!
!                                     The University of
Huddersfield
!
21/09/2019
!-----
!-----
! Input:(   i   : Gaussian_point_number;
!         points(:,): array_stores_sampling_function)
!Output:(der(:,): the_array_stores_the shape_function derivatives_of
!         _3D_Goodman)
!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

INTEGER,INTENT(IN)::i

REAL(iwp),INTENT(IN)::points(:,)

REAL(iwp),INTENT(OUT)::der(:,)

REAL(iwp)::eta,xi,zeta,etam,etap,xim,xip,c1,c2,c3

```

REAL(iwp)::t1,t2,t3,t4,t5,t6,t7,t8,t9,x2p1,x2m1,e2p1,e2m1,zetam,zetap

REAL,PARAMETER::zero=0.0_iwp,pt125=0.125_iwp,pt25=0.25_iwp,pt5=0.5_i

wp, &

pt75=0.75_iwp,one=1.0_iwp,two=2.0_iwp,d3=3.0_iwp,d4=4.0_iwp,d5=5.0_iwp,

&

d6=6.0_iwp,d8=8.0_iwp,d9=9.0_iwp,d10=10.0_iwp,d11=11.0_iwp,

&

d12=12.0_iwp,d16=16.0_iwp,d18=18.0_iwp,d27=27.0_iwp,d32=32.0_iwp,

&

d36=36.0_iwp,d54=54.0_iwp,d64=64.0_iwp,d128=128.0_iwp

INTEGER::xii(20),etai(20),zetai(20),l,ndim,nod

nod=UBOUND(der,2)

xi=points(i,1)

eta=points(i,2)

c1=xi

c2=eta

c3=one-c1-c2

etam=pt25*(one-eta)

etap=pt25*(one+eta)

xim= pt25*(one-xi)

xip= pt25*(one+xi)

x2p1=two*xi+one

x2m1=two*xi-one

e2p1=two*eta+one

e2m1=two*eta-one

SELECT CASE(nod)

CASE(3)

der(1,1)=one

der(1,3)=zero

der(1,2)=-one

der(2,1)=zero

der(2,3)=one

der(2,2)=-one

CASE(6)

der(1,1)=d4*c1-one

der(1,6)=d4*c2

der(1,5)=zero

der(1,4)=-d4*c2

der(1,3)=-(d4*c3-one)

der(1,2)=d4*(c3-c1)

$$\text{der}(2,1)=\text{zero}$$

$$\text{der}(2,6)=d4*c1$$

$$\text{der}(2,5)=d4*c2-\text{one}$$

$$\text{der}(2,4)=d4*(c3-c2)$$

$$\text{der}(2,3)=-(d4*c3-\text{one})$$

$$\text{der}(2,2)=-d4*c1$$

CASE(10)

$$\text{der}(1,1)=(d27*c1**2-d18*c1+\text{two})/\text{two}$$

$$\text{der}(1,9)=(d9*(d6*c1-\text{one})*c2)/\text{two}$$

$$\text{der}(1,8)=(d9*(d3*c2-\text{one})*c2)/\text{two}$$

$$\text{der}(1,7)=\text{zero}$$

$$\text{der}(1,6)=-(d9*(d3*c2-\text{one})*c2)/\text{two}$$

$$\text{der}(1,5)=(d9*(d6*c1+d6*c2-d5)*c2)/\text{two}$$

$$\text{der}(1,4)=-(d27*c1**2+d54*c1*c2-d36*c1+d27*c2**2-d36*c2+d11)/\text{two}$$

$$\text{der}(1,3)=(d9*(d9*c1**2+d12*c1*c2-d10*c1+d3*c2**2-d5*c2+\text{two}))/\text{two}$$

$$\text{der}(1,2)=-(d9*(d9*c1**2+d6*c1*c2-d8*c1-c2+\text{one}))/\text{two}$$

$$\text{der}(1,10)=-d27*(((c2-\text{one})+c1)+c1)*c2$$

$$\text{der}(2,1)=\text{zero}$$

$$\text{der}(2,9)=(d9*(d3*c1-\text{one})*c1)/\text{two}$$

$$\text{der}(2,8)=(d9*(d6*c2-\text{one})*c1)/\text{two}$$

$$\text{der}(2,7)=(d27*c2**2-d18*c2+\text{two})/\text{two}$$

$$\text{der}(2,6)=-(d9*((c1+c2-\text{one})*(d6*c2-\text{one})+(d3*c2-\text{one})*c2))/\text{two}$$

$$\text{der}(2,5) = (d9*(d3*c1**2+d12*c1*c2-d5*c1+d9*c2**2-d10*c2+two))/two$$

$$\text{der}(2,4) = -(d27*c1**2+d54*c1*c2-d36*c1+d27*c2**2-d36*c2+d11)/two$$

$$\text{der}(2,3) = (d9*(d6*c1+d6*c2-d5)*c1)/two$$

$$\text{der}(2,2) = -(d9*(d3*c1-one)*c1)/two$$

$$\text{der}(2,10) = -d27*(((c2-one)+c1)+c2)*c1$$

CASE(15)

$$t1 = c1 - pt25$$

$$t2 = c1 - pt5$$

$$t3 = c1 - pt75$$

$$t4 = c2 - pt25$$

$$t5 = c2 - pt5$$

$$t6 = c2 - pt75$$

$$t7 = c3 - pt25$$

$$t8 = c3 - pt5$$

$$t9 = c3 - pt75$$

$$\text{der}(1,1) = d32/d3*(t2*t3*(t1+c1)+c1*t1*(t3+t2))$$

$$\text{der}(1,12) = d128/d3*c2*(t2*(t1+c1)+c1*t1)$$

$$\text{der}(1,11) = d64*c2*t4*(t1+c1)$$

$$\text{der}(1,10) = d128/d3*c2*t4*t5$$

$$\text{der}(1,9) = \text{zero}$$

$$\text{der}(1,8) = -d128/d3*c2*t4*t5$$

$$\text{der}(1,7) = -d64*c2*t4*(t7+c3)$$

$$\text{der}(1,6)=-d128/d3*c2*(t8*(t7+c3)+c3*t7)$$

$$\text{der}(1,5)=-d32/d3*(t8*t9*(t7+c3)+c3*t7*(t8+t9))$$

$$\text{der}(1,4)=d128/d3*(c3*t7*t8-c1*(t8*(t7+c3)+c3*t7))$$

$$\text{der}(1,3)=d64*(c3*t7*(t1+c1)-c1*t1*(t7+c3))$$

$$\text{der}(1,2)=d128/d3*(c3*(t2*(t1+c1)+c1*t1)-c1*t1*t2)$$

$$\text{der}(1,13)=d128*c2*(c3*(t1+c1)-c1*t1)$$

$$\text{der}(1,15)=d128*c2*t4*(c3-c1)$$

$$\text{der}(1,14)=d128*c2*(c3*t7-c1*(t7+c3))$$

$$\text{der}(2,1)=\text{zero}$$

$$\text{der}(2,12)=d128/d3*c1*t1*t2$$

$$\text{der}(2,11)=d64*c1*t1*(t4+c2)$$

$$\text{der}(2,10)=d128/d3*c1*(t5*(t4+c2)+c2*t4)$$

$$\text{der}(2,9)=d32/d3*(t5*t6*(t4+c2)+c2*t4*(t6+t5))$$

$$\text{der}(2,8)=d128/d3*((c3*(t5*(t4+c2)+c2*t4))-c2*t4*t5)$$

$$\text{der}(2,7)=d64*(c3*t7*(t4+c2)-c2*t4*(t7+c3))$$

$$\text{der}(2,6)=d128/d3*(c3*t7*t8-c2*(t8*(t7+c3)+c3*t7))$$

$$\text{der}(2,5)=-d32/d3*(t8*t9*(t7+c3)+c3*t7*(t8+t9))$$

$$\text{der}(2,4)=-d128/d3*c1*(t8*(t7+c3)+c3*t7)$$

$$\text{der}(2,3)=-d64*c1*t1*(t7+c3)$$

$$\text{der}(2,2)=-d128/d3*c1*t1*t2$$

$$\text{der}(2,13)=d128*c1*t1*(c3-c2)$$

$$\text{der}(2,15)=d128*c1*(c3*(t4+c2)-c2*t4)$$

$$\text{der}(2,14)=d128*c1*(c3*t7-c2*(c3+t7))$$

CASE (4)

$$\text{der}(1,1)=-\text{etam}$$

$$\text{der}(1,2)=-\text{etap}$$

$$\text{der}(1,3)=\text{etap}$$

$$\text{der}(1,4)=\text{etam}$$

$$\text{der}(2,1)=-\text{xim}$$

$$\text{der}(2,2)=\text{xim}$$

$$\text{der}(2,3)=\text{xip}$$

$$\text{der}(2,4)=-\text{xip}$$

CASE(5)

$$\text{der}(1,1)=-\text{etam}+\text{pt5}*\text{xi}*(\text{one}-\text{eta}^{**2})$$

$$\text{der}(1,2)=-\text{etap}+\text{pt5}*\text{xi}*(\text{one}-\text{eta}^{**2})$$

$$\text{der}(1,3)=\text{etap}+\text{pt5}*\text{xi}*(\text{one}-\text{eta}^{**2})$$

$$\text{der}(1,4)=\text{etam}+\text{pt5}*\text{xi}*(\text{one}-\text{eta}^{**2})$$

$$\text{der}(1,5)=-\text{two}*\text{xi}*(\text{one}-\text{eta}^{**2})$$

$$\text{der}(2,1)=-\text{xim}+\text{pt5}*\text{eta}*(\text{one}-\text{xi}^{**2})$$

$$\text{der}(2,2)=\text{xim}+\text{pt5}*\text{eta}*(\text{one}-\text{xi}^{**2})$$

$$\text{der}(2,3)=\text{xip}+\text{pt5}*\text{eta}*(\text{one}-\text{xi}^{**2})$$

$$\text{der}(2,4)=-\text{xip}+\text{pt5}*\text{eta}*(\text{one}-\text{xi}^{**2})$$

$$\text{der}(2,5)=-\text{two}*\text{eta}*(\text{one}-\text{xi}^{**2})$$

CASE(8)

$$\text{der}(1,1)=\text{etam}*(\text{two}*\text{xi}+\text{eta})$$

$$\text{der}(1,2)=-\text{d8}*\text{etam}*\text{etap}$$

$$\text{der}(1,3)=\text{etap}*(\text{two}*\text{xi}-\text{eta})$$

$$\text{der}(1,4)=-\text{d4}*\text{etap}*\text{xi}$$

$$\text{der}(1,5)=\text{etap}*(\text{two}*\text{xi}+\text{eta})$$

$$\text{der}(1,6)=\text{d8}*\text{etap}*\text{etam}$$

$$\text{der}(1,7)=\text{etam}*(\text{two}*\text{xi}-\text{eta})$$

$$\text{der}(1,8)=-\text{d4}*\text{etam}*\text{xi}$$

$$\text{der}(2,1)=\text{xim}*(\text{xi}+\text{two}*\text{eta})$$

$$\text{der}(2,2)=-\text{d4}*\text{xim}*\text{eta}$$

$$\text{der}(2,3)=\text{xim}*(\text{two}*\text{eta}-\text{xi})$$

$$\text{der}(2,4)=\text{d8}*\text{xim}*\text{xip}$$

$$\text{der}(2,5)=\text{xip}*(\text{xi}+\text{two}*\text{eta})$$

$$\text{der}(2,6)=-\text{d4}*\text{xip}*\text{eta}$$

$$\text{der}(2,7)=\text{xip}*(\text{two}*\text{eta}-\text{xi})$$

$$\text{der}(2,8)=-\text{d8}*\text{xim}*\text{xip}$$

CASE(9)

$$\text{etam}=\text{eta}-\text{one}$$

$$\text{etap}=\text{eta}+\text{one}$$

$$\text{xim}=\text{xi}-\text{one}$$

$$\text{xip}=\text{xi}+\text{one}$$

$$\text{der}(1,1)=\text{pt25}*\text{x2m1}*\text{eta}*\text{etam}$$

der(1,2)=-pt5*x2m1*etap*etam

der(1,3)=pt25*x2m1*eta*etap

der(1,4)=-xi*eta*etap

der(1,5)=pt25*x2p1*eta*etap

der(1,6)=-pt5*x2p1*etap*etam

der(1,7)=pt25*x2p1*eta*etam

der(1,8)=-xi*eta*etam

der(1,9)=two*xi*etap*etam

der(2,1)=pt25*xi*xim*e2m1

der(2,2)=-xi*xim*eta

der(2,3)=pt25*xi*xim*e2p1

der(2,4)=-pt5*xip*xim*e2p1

der(2,5)=pt25*xi*xip*e2p1

der(2,6)=-xi*xip*eta

der(2,7)=pt25*xi*xip*e2m1

der(2,8)=-pt5*xip*xim*e2m1

der(2,9)=two*xip*xim*eta

CASE DEFAULT

WRITE(*,*)"wrong number of nodes in shape_der"

END SELECT

RETURN

END SUBROUTINE G_shape_der

SUBROUTINE TM_TD(T,Q,coord)

!-----

! This subroutine forms the transmission matrix of 3D

! goodman element

! Research:J.Tu

! The University of Huddersfield

! 09/10/2018

!-----

!-----

! Input:(coord(:,,:): Goodman_element_node_coordinate_array_in_global_

! _Coordinates_system)

!Output:(T(:,,:): the_T_matrix_in_Equation_2.8.21;

! Q(:,,:): the_Q_matrix_in_Equation_2.8.21)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

REAL(iwp),INTENT(IN)::coord(:,,:)

REAL(iwp),INTENT(OUT)::T(:,,:),Q(:,,:)

INTEGER::i

REAL::x1,y1,z1,x2,y2,z2,x4,y4,z4,A,B,C,D,E,F,LXX,LXY,LXZ, &

LYX,LYY,LYZ,LZX,LZY,LZZ,LX,LY,LZ,COSXX,COSXY,COSXZ,COSYX, &

COSYY,COSYZ,COSZX,COSZY,COSZZ,zero=0.0_iwp

!-----inport the coordinate -----

x1=coord(1,1)

y1=coord(1,2)

z1=coord(1,3)

x2=coord(2,1)

y2=coord(2,2)

z2=coord(2,3)

x4=coord(4,1)

y4=coord(4,2)

z4=coord(4,3)

!-----

A=x4-x1

B=y4-y1

C=z4-z1

D=x2-x1

E=y2-y1

$$F=z_2-z_1$$

!----- For Local X direction-----

$$LXX=A$$

$$LXY=B$$

$$LXZ=C$$

$$LX=\text{SQRT}(LXX^2+LXY^2+LXZ^2)$$

$$\text{COSXX}=LXX/LX$$

$$\text{COSXY}=LXY/LX$$

$$\text{COSXZ}=LXZ/LX$$

!----- For Local Y direction-----

$$LYX=(C^2)*D-A*C*F-A*B*E+(B^2)*D$$

$$LYY=(A^2)*E-A*B*D-B*C*F+(C^2)*E$$

$$LYZ=(B^2)*F-B*C*E-A*C*D+(A^2)*F$$

$$LY=\text{SQRT}(LYX^2+LYY^2+LYZ^2)$$

$$\text{COSYX}=LYX/LY$$

$$\text{COSYY}=\text{LYY}/\text{LY}$$

$$\text{COSYZ}=\text{LYZ}/\text{LY}$$

!----- For Local Z direction-----

$$\text{LZX}=\text{B}*\text{F}-\text{C}*E$$

$$\text{LZY}=\text{C}*D-\text{A}*F$$

$$\text{LZZ}=\text{A}*E-\text{B}*D$$

$$\text{LZ}=\text{SQRT}(\text{LZX}^{**2}+\text{LZY}^{**2}+\text{LZZ}^{**2})$$

$$\text{COSZX}=\text{LZX}/\text{LZ}$$

$$\text{COSZY}=\text{LZY}/\text{LZ}$$

$$\text{COSZZ}=\text{LZZ}/\text{LZ}$$

!-----

$$\text{Q}=\text{ZERO}$$

$$\text{Q}(1,1)=\text{COSXX}$$

$$\text{Q}(1,2)=\text{COSXY}$$

$$\text{Q}(1,3)=\text{COSXZ}$$

Q(2,1)=COSYX

Q(2,2)=COSYY

Q(2,3)=COSYZ

Q(3,1)=COSZX

Q(3,2)=COSZY

Q(3,3)=COSZZ

T=ZERO

!-----

do i = 1, 8

 T((i-1)*3+1:i*3,(i-1)*3+1:i*3)=Q

end do

RETURN

END SUBROUTINE TM_TD

SUBROUTINE T_coord(Q,coord)

!-----

! This subroutine forms the transmittion matrix of 3D

! goodman element

! Research:J.Tu

! The University of Huddersfield

! 09/10/2018

!-----

!-----

! Input:(coord(:,,:): Goodman_element_node_coordinate_array_in_global_

! _Coordinates_system)

!Output:(Q(:,,:))

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

REAL(iwp),INTENT(IN OUT)::coord(:,,:)

REAL(iwp),INTENT(IN)::Q(:,,:)

INTEGER::i

```
REAL::gcoord(3)
```

```
!-----inport the coordinate -----
```

```
DO i=1,8
```

```
    gcoord(:)=coord(i,:)
```

```
    gcoord=MATMUL(Q,gcoord)
```

```
    coord(i,:)=gcoord(:)
```

```
END DO
```

```
RETURN
```

```
END SUBROUTINE T_coord
```

```
SUBROUTINE gbvm3(dp,db,w,uslide,rt,gcprop,p,b,sigma1)
```

```
!-----
```

```
! This subroutine is used to implement the creep constitutive equation
```

```
! of Markus Vose creep model for grain boundary. The reference for this
```

```
! model is :
```

```
!   Vöse, M., Otto, F., Fedelich, B. and Eggeler, G. (2014).
```

```
!   Micromechanical investigations and modelling of a
```

```
!   Copper–Antimony-Alloy under creep conditions.
```

! Mechanics of Materials, 69(1), pp.41-62..

20/11/2018

! The output result is normalized.

!-----

! Input: (gcprop(:):grain_boundary_creep_constitutive_equation

! _paramaters_array;

! p :input_grain_boundary_creep_cavity_density;

! b :input_grain_boundary_creep_damage_variable)

!Output: (dp :output_grain_boundary_creep_cavity_density_rate;

! db :output_grain_boundary_creep_damage_variable_rate;

! w :output_grain_boundary_creep_damage;

! uslide:output_grain_boundary_creep_sliding_rate

! rt :real_time_step)

!-----

IMPLICIT NONE

INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)

Doubleprecision,INTENT(IN)::gcprop(:),p,b,sigma1(:)

Doubleprecision,INTENT(OUT),OPTIONAL::dp,db,w,uslide,rt

Doubleprecision::a1,rs,xp,r,theta,x1,x2,x3,x4,theta1,flow1, &

flow2,flow3,flow4,flow5,flow6,flow7,y,xa,up,stip0,qw0,pi=3.1415926, &

one=1.0_iwp,two=2.0_iwp,d3=3.0_iwp,d4=4.0_iwp,flow8,flow9,htheta,
&
flow10,flow11,flow12,flow13,flow14,f,dup,dgb,da,a,zero=0.0_iwp, &
a2,xa1,a3,x5,rdsbar,rap,pdsbar,flow15,flow16,flow17,flow18,flow19, &
d1,bp,ap,d2,dsbar1,p1,dsbar,tdsbar,yslide,rtdsbar

!-----input the creep parameters for the constitutive equation-----

a1=gcprop(1)

dgb=gcprop(2)

rs=gcprop(3)

ap=gcprop(4)

bp=gcprop(5)

theta=gcprop(6)

x1=gcprop(7)

x2=gcprop(8)

x3=gcprop(9)

x4=gcprop(10)

x5=gcprop(11)

pdsbar=gcprop(12)

!-----import the stress state-----

$$\text{dsbar}=\text{sigma1}(2)$$

$$\text{tdsbar}=\text{sigma1}(1)$$

!-----the normalization parameters-----

$$\text{rdsbar}=\text{rs}/\text{a1}$$

$$\text{rt}=(\text{a1}^{**}(\text{d4})) / (\text{dgb} * \text{rs})$$

$$\text{dsbar1}=\text{dsbar} / \text{rdsbar}$$

$$\text{rap}=\text{ap} * \text{rt} * (\text{a1}^{**}2) * ((\text{rdsbar} / \text{pdsbar})^{**}(\text{bp}))$$

$$\text{rtsbar}=\text{tdsbar} / \text{rdsbar}$$

$$\text{p1}=\text{p} * (\text{a1}^{**}2)$$

!----- prepare for simulation-----

$$\text{theta1}=(\text{theta} * \text{pi}) / 180$$

$$\text{flow1}=(1 + \cos(\text{theta1}))^{**}(-1) - (\cos(\text{theta1})) / 2$$

$$\text{htheta}=\text{flow1} / \sin(\text{theta1})$$

!-----the constitutive equation part-----

!-----equation 14-----

$$\text{w}=(9 * \text{pi} * (\text{b}^{**}2)) / (16 * (\text{htheta}^{**}2))^{**}(\text{one} / \text{d3})$$

$$a=\sqrt{w/(\pi*p1)}$$

!-----equation 12-----

$$stip0=(2*\sin(\theta_1))/a$$

$$qw0=(-2)*d\log(w*x^2)-((3-w*x^2)*(1-w*x^2))$$

$$flow2=dsbar1-stip0*(1-x^2*w)$$

$$flow3=(a^2)*qw0$$

$$da=(x^2*flow2)/(h\theta*flow3)$$

!-----equation 15-----

if(da.GE.0)then

$$xa=x^3*4*\pi*(p1^2)*a*da$$

else if(da.LT.0) then

$$a3=1.1$$

$$a2=a$$

if(a.LE.a3)then

$$a2=a3$$

end if

$$xa1=x^5*p1*da*(-1)$$

$xa = xa1 / (2 * (a2 - 1))$

end if

!-----equation 17-----

if(da.GT.zero)then

$flow15 = stip0 * (1 - x2 * w)$

$flow16 = rap * ((dsbar1) ** (bp))$

$flow17 = rap * (flow15 ** (bp))$

$flow18 = (flow16 - flow17) ** (two)$

$flow19 = rap * (dsbar1 ** (bp))$

$xp = flow18 / flow19$

else if(da.LE.zero)then

$xp = zero$

end if

!-----equation 18(3)-----

$$\text{flow4}=(b*(xp-xa))/(\text{sqrt}(p1**d3))$$

$$\text{flow10}=d3*b$$

$$\text{flow11}=a*(\text{sqrt}(p1))$$

$$\text{flow12}=\text{flow10}/\text{flow11}$$

$$\text{flow5}=\text{flow12}*da$$

$$\text{dup}=\text{flow4}+\text{flow5}$$

!-----equation 18(1&2)-----

$$\text{flow14}=-\text{one}$$

$$\text{flow6}=x4*2*\text{pi}*p1*((\text{dup})**(\text{flow14}))$$

$$\text{flow7}=(2*\text{sin}(\text{theta1}))-stip0$$

$$y=\text{exp}(\text{flow6}*\text{flow7})$$

$$f=((y-1)*w)/(1-w)$$

if(f.GE.one)then

$$f=\text{one}$$

end if

!-----equation 11(1&2)-----

$dp = xp \cdot (1-f) - xa$

$flow8 = (3 \cdot b \cdot (xp - xa)) / (2 \cdot p1)$

$flow9 = (flow10/a) \cdot da$

$db = flow8 + flow9$

!-----output the grain boundary sliding-----

!-----equation 19-----

$y_{slide} = gcprop(13)$

$flow10 = a1 / (rt \cdot rdsbar)$

$u_{slide} = rtdsbar / (y_{slide} \cdot flow10)$

$u_{slide} = u_{slide} \cdot a1$

! write(11,*)"xp",xp

! write(11,*)"xa",xa

RETURN

END subroutine gbvm3

End Module math

Appendix III Tutorial on INP file.

The INP file is used to store and import the FE model's information into the main program, including node coordinates, element topology, boundary conditions, material information, loading information, and so on. This information is read line by line through the interface; hence these instructions need to have a specific format.

I. INP file for Notched Bar Case Study

In the main body of the procedure, the data is read by the channel 10 to import the data line by line. The INP file has 6 main part.

The initialization section.

In this section, the main purpose is to evaluate the case size to allocate the size for these dynamic arrays. The main data and its sequence in this section are shown in below.

In this section, the data is read in through these interfaces.

```
READ (10,*) element, reboots, step, bs  
READ (10,*) ndim, nst, nip, nprops  
READ (10,*) nod, ndof, nodof, np_types  
READ (10,*) nn, nels
```

List AP3.1

The sequence of these data is:

'element': the element type (Triangle or Quadrilateral).

'reboots': the switcher of the restart facility. (1 is open, 0 is closed)

'step': the results are exported in this iterating step.

'bs': the stiffness matrix reducing factor of a failed element.

'nidm': the dimension of the case. (2 or 3)

nst: the number of stress/strain terms.

nip: the number of Gaussian Points per element.

nprop: the number of elastic material parameters.

nod: the number of node per element.

ndof: the number of freedom degrees per element.

nodof: the number of freedom degrees per node.

np_type: the number of material in this FE model.

nn: the total number of nodes in this FE model.

nels: the total number of elements in this FE model.

The boundary condition section.

In this section, the main purpose is to import the boundary conditions into the main program.

In this section, the data is read in through these interfaces, using List AP4.1 as an example.

```
READ (10,*) nr, (k, nf(:,k), i=1, nr)
```

List AP3.2

The sequence of these data is:

nr: the total number of restrained nodes. in this case, it is '67'

K: is the node number of the restrained nodes.

nf(:,K): is the restrained direction of these nodes. ('0' is fixed, '1' is open). In this

case, (2 1 0) is means: fixed the Y direction of NO.2 node.

The node coordinates

In this section, the code coordinates of all node is read in through these interfaces, using List AP4.7 as an example.

```
do i=1,nn
    read (10,*)k,g_coord(:,i)
end do
```

List AP3.3

The sequence of these data is:

K: the node number.

g_coord(:,i): the coordinate of each node.

In this case, (1, 2.24907994, 0.496734006) is means: the coordinate of NO.1 node is (2.24907994, 0.496734006).

The element node numbering section.

In this section, the code coordinates of all node is read in through these interfaces, using List AP4.7 as an example.

```
do i=1,nels
    read (10,*)s, g_num(:,s)
end do
```

List AP 3.4

g_num(:,i): the array stores the node numbers of all element.

In this case, (11 28162 280472792278) is means: the topology of NO.1 element.

The loading section.

In this section, the node information is read in through these interfaces, using

List AP4.7 as an example.

```
DO i=1,loaded_nodes
  loads (nf (:,node (i)))=val (i,:) *qinc (iy)
END DO
```

List AP3.5

loaded_nodes: the number of loaded nodes.

In this case, the external load is loaded in 26 nodes. For example, NO.43, the loading factor in the x direction is 0, and in the Y direction is 0.620963042.

The Gaussian Point coordinate section.

In this section, the coordinate of Gaussian points is to import into the main program.

quadrilateral' 1 26334 10000
 2 4 4 3
 8 16 2 1
 793 240
 1 160e3 0.3

Section 1

67
 2 1 0

 668 0.1

Section 2

1, 2.24907994, 0.496734006

 793, 6.26, 2.63162886

Section 3

1	1	281	62	280
47279		2	278	
.....				
240	162	790	166	793
46792	46	77826		

Section 4

26
 39 0.0 0.0

 43 0.0 0.620963042
 200000
 1
 33.48

Section 6

1		
	1	2.64796490221226 0.361688666968341
	2	2.64680617061469 9.691416942806762E-002
	3	2.32916698281864 0.383126321238609
	4	2.32868728446441 0.102668388374992
	2	
.....		

Section 6

List. AP3.6

Appendix IV Publication Contribution List

Q.X. conceived the idea to simulate the grain and grain boundary separately and preferred the in-house software development approach; Q.X. also participated in drafting, finalizing and submitting the paper; rewrote the abstract, introduction, discussion and future work for the revised version; advised what extra data/graphs (primarily Sections 4 and 5) to be produce in revised version and wrote the discussion; Q.X. finalized the paper and made submissions for all the three versions.

J.Tu proceeded with the theory, programming, and testing. J.Tu also drafted the paper and participated in revising prior to its first submission, in addition to producing the data/diagrams, and provided some discussion regarding the results for the revised version;

Z.L. participated in the initial drafting process.

