INVESTIGATIONS OF A COMPUTATIONAL SCHEME FOR ACCELERATED NUMERICAL SIMULATIONS OF CYCLICALLY LOADED STRUCTURES

by
Yun Wang

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

Summer 2010

INVESTIGATIONS OF A COMPUTATIONAL SCHEME FOR ACCELERATED NUMERICAL SIMULATIONS OF CYCLICALLY LOADED STRUCTURES

by

Yun Wang

Approved: _____
Anette M. Karlsson, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____
Anette M. Karlsson, Ph.D.
Chair of the Department of Mechanical Engineering

Approved: _____
Michael J. Chajes, Ph.D.
Dean of the College of Engineering

Approved: _____
Debra Hess Norris, M.S.
Vice Provost for Graduate and Professional Education

ACKNOWLEDGMENTS

I would like to record my particular gratitude to my advisor, Dr. Anette M. Karlsson. In addition to introducing me to the computational modeling, inspiring me to think creatively, and teaching me skills in technical writing, Dr. Karlsson provided support and help during these years when I confronted any difficulties. She is an extraordinary mentor.

I want to thank Dr. Michael H. Santare for his helpful suggestions regarding my research work. I also have learned necessary knowledge for my research from his course Continuum Mechanics.

I gratefully thank Dr. Bingqing Wei, who, in the midst of his activities, agreed to be members of my thesis committee.

Special thanks to Dr. Dan Cojocaru. He has given me a lot of expert advice on learning the programming language. His help on programming issues is very important for me.

I want to thank all my lab-mates: Dr. Liang Cheng, Dr. Mercedes Hernandez, Narinder Singh, Zongwen Lu, Dr. Hongzhou Li, Vasan Chandrasekaran, Melissa Lugo. Thank you for your advice on my research and study. It is great pleasure to work with you.

I also want to thank all the staff in the Mechanical Engineering Department, Lisa Katzmire, Letitia Toto and Ann Connor. They are so nice and helpful.

Special thanks to my mother, Lingyi Li and my father, Linwei Wang. They make all my efforts meaningful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Engineering structures are commonly subjected to cyclic loading. This leads to fatigue and unwanted, premature failures. Thermal barrier coatings (TBCs) are example of structures exposed to both thermal and mechanical cycles. Understanding the failure evolution in these structures during use is important to prevent the life-time limiting failures. To predict failures due to cyclic loading, finite element analysis (FEA) is commonly used, which can simulate and give the stress and strain distribution as a function of time. However, every single cycle of simulation may take significant computation time. For structures subjected to cyclic loadings, it is very time-consuming and inefficient to simulate the whole process of structural evolution.

The goal with this work is to improve upon an existing numerical scheme that can, in combination with simple testing, predict the life-time of structures subjected to cyclic loading. The numerical scheme is the "cycle-jump technique" developed previously. The idea behind the cycle-jump technique is that there is no need to calculate each individual cycle in cyclically loaded structure. The cycle-jump technique predicts the overall structural behavior utilizing an extrapolation scheme.

As part of the improvement of the numerical scheme, the simulations must be compared to experimental data. In order to do so, the cycle-jump technique code is

modified to be used for real life experiments. The materials used in these experiments are materials used in high temperature applications with particular focus on TBCs. TBCs exhibits time dependent material response, thus creep properties are incorporated into the cycle-jump technique code. Furthermore, in order to allow for displacement controlled loading, “intermediate initiation simulation” is incorporated as a numerical tool to impose the necessary initial conditions in a time-economic manner.

Suitable load sequences are established via numerical simulation so to develop experimental investigation suitable as verifying tests. In addition, sensitivity analyses are conducted to investigate the effect of expected variance of material properties in the real structure.

Finally, preliminary experimental results are used to compare with the simulation results. Even though the experimental data is very limited, it appears as cycle-jump technique can predict the evolution of the test sample.

Chapter 1

BACKGROUND

1.1 Introduction

Many structures are subjected to cyclic loading (e.g. thermal cycles). Repeated cyclic loading eventually leads to fatigue and unwanted failures of structures even if the induced stresses are significantly lower than the critical stress needed for static failure. For example, thermal barrier coatings (TBCs), used in the hot sections of gas turbine engines, are exposed to both thermal and mechanical cycles and eventually fail by spalling from the superalloy substrate [1]. To understand the failure evolution of these structures, important information of materials under specific load conditions are needed. Failures in TBCs are mostly driven by the cyclic loading and a slow evolution of material properties, e.g., the yield-strength or elastic modulus may change during time at high temperature.

To predict failures due to cyclic loading, the state of stress and/or strain is useful, even necessary, to establish. However, there are many challenges involved with investigating how the stress and strain of the structures respond as the material or structural properties evolve with time in combination with cyclic loading. Finite element analysis (FEA) [2-4] is a useful tool that can simulate and give the stress and strain distribution as a function of time. However, every single cycle of simulation may involve

numerous numerical increments and iterations, which may take significant computation time. For structures subjected to cyclic loadings, it is very time-consuming and inefficient to simulate the whole process of structural evolution.

The goal with this work is to improve and verify a numerical scheme that can, in combination with simple testing, shorten the computational time for structures subjected to cyclic loading. The numerical scheme is the "cycle-jump technique"[5], which was developed previously. The idea behind the cycle-jump technique is that it is too cumbersome and time-consuming to calculate each individual cycle in cyclically loaded structures. Consequently, the cycle-jump technique utilizes an extrapolation scheme, so that the overall structural behavior can be predicted.

This thesis is organized in the following manner. After reviewing pertinent literature considering accelerated numerical simulations in section 1.2, the extrapolation scheme that the cycle-jump technique is based on is described in section 1.3.1. In section 1.3.2, a short description of thermal barrier coatings is given since these coatings have inspired this work. The improvement of cycle-jump technique made to adopt the technique to high temperature materials is discussed in chapter 2. The experimental work that will be used for verifying our improved routine is discussed in chapter 3.

1.2 Literature Review of Accelerated Numerical Simulations

Simulations of models subjected to cyclic loadings are time-consuming. The aim of this research is to improve the computational efficiency, thus saving the computational time. Only a limited set of investigators have considered the problem of accelerated numerical simulations, and a short review is given here.

In the early 1990's, Ladeveze and colleagues [6-7] developed a scheme to investigate thermo-mechanical problems under cyclic loadings. The method is called the "Large Time Increments Method" and the aim is to improve the efficiency of the calculation. In this method, the "general" equations needed to be solved for the isothermal viscoplastic problems are all decomposed into two groups of equations: a "local" and a "global" equation. For every single cycle, the "local" equations are solved and they deal with the non-linear part of the "general" equations. The "global" equation deals with the linear part of the "general" equation and must be solved over the total loading time. The two groups of equations are calculated separately to improve the efficiency and the combined results of the two groups of equations constitute the final results for the reaction of the structures. This is an illuminate idea, but it is not very feasible to incorporate this complicated idea into a finite element programs.

In Fish's [8-9] method, the time is decomposed into two temporal coordinates. One is micro-chronological (fast time scale) and the other is macro-chronological (slow time scale). All the responses of the state variables are assumed to depend on these two coordinates. The micro-chronological coordinate represents the response during each individual cycle while the macro-chronological coordinate represents the general trend of the structure. Using the combination of the two coordinates' response, the responses of the state variables can be obtained and fatigue propagation is predicted. In theory, the method is good; however, the method is not practical to incorporate into the finite element program due to the cumbersome formulas and complicated iterative calculation.

Kiewal et al. [10] considered inelastic material models that need significant computer resources when subjected to cyclic loadings. In order to reduce the computational time, Kiewal developed a method to extrapolate the internal variables (e.g. a displacement, stress or strain component) over a certain number of cycles. The extrapolation procedure begins with a regular integration of the evolution equations of the internal variables for the first loading cycle. In this method the formula of Runge-Kutta of $4^{th}/5^{th}$ order is applied. Based on the integration, a piecewise polynomial or spline function for each integration point of the structure and for each internal variable can be obtained. Using the spline function, the internal variables can then be extrapolated. An error function is also defined to control the length of the extrapolation. In spite of achieving the goal of saving some computational time, the method is developed based on using damage parameter which does not give any new information which should be predicted into the failure evolution.

Bogard [11] developed a numerical simulation procedure to predict the fatigue damage in elasto-plastic structures subjected to cyclic loadings. The simulation incorporates a cycle jumping algorithm that is similar to our cycle-jump technique described below. However, the technique is based on the concept of damage theory. At the beginning of the simulation, damage effects are not considered. The internal variable can be expressed as a function of time (number of cycles) and then the function is transferred into a Taylor series of second order. With the number of cycles in the Taylor series, the size of the cycle-jump is determined. At the fatigue damage stage, a damage parameter is defined [12] which is related to the stress and strain and then a damage

model which provides a relation between the stress, number of cycles and the damage parameter is developed by Lemaitre and Chaboche [13]. The jump length is then determined by the damage parameter. The method is efficient; however, it is customized for the specific application investigated. The damage theory is incorporated in the method and the jump length is decided mainly based on the damage model, so it could only be used by a few structures with specific material properties.

As discussed, all the methods previously developed have some limitations: some of the ideas are hard to realize by the finite element program, others could only be used by a small group of structures. A general method to improve the computational efficiency and provide insight into failure evolution is needed. This is addressed by the method discussed in section 1.3.1.

1.3 Existing Modeling Schemes

In this section, we will discuss the numerical schemes that will serve as the foundation for this thesis: the "cycle-jump technique."

1.3.1 The Cycle-Jump Technique

As already discussed, it can be time consuming to simulate a structure under a single load cycle; to model multiple load cycles can become very lengthy and cumbersome. Thus, the concept of "cycle jumps" can be a very useful approach to reduce the computational time. To accomplish "cycle jumps", a technique was previously developed by Cojucaru and Karlsson [5]. This approach will be used in this work and we will briefly review this approach in this sub-section. In the cycle-jump technique, not all individual

cycles need to be simulated. By identifying a general evolution of the structure, the simulation could skip, or "jump," over some cycles.

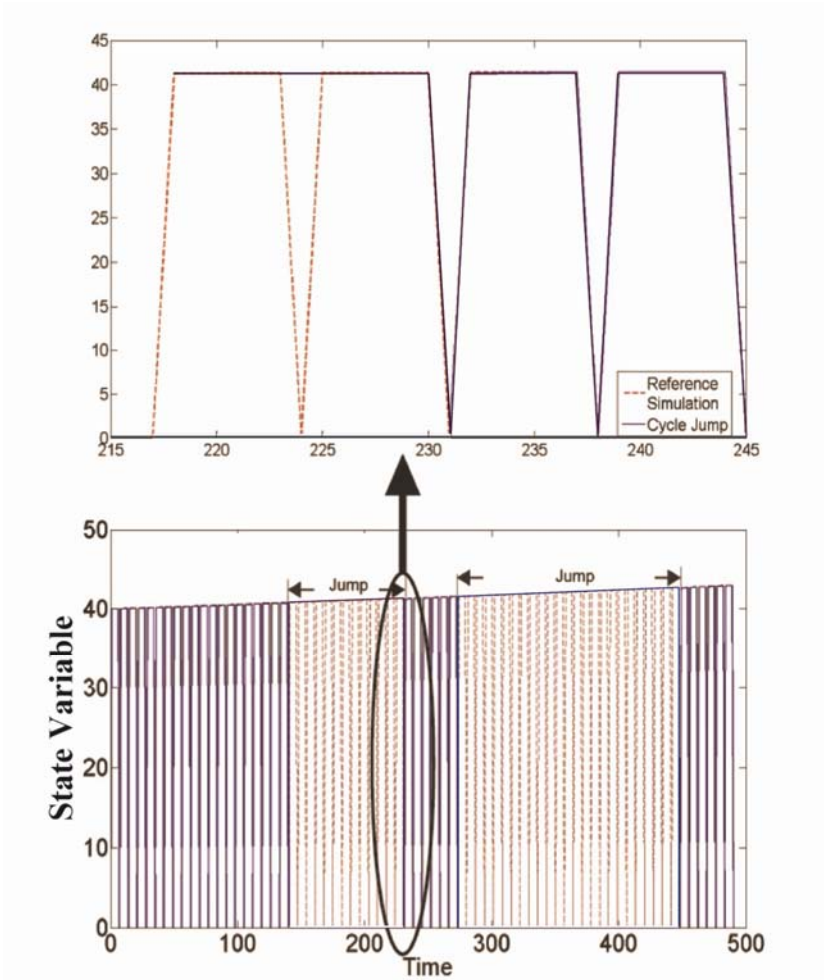


Fig. 1.1 The evolving state variable at a selected point as a function of time of a structure subjected to cyclic loading (The cycle-jump simulation is compared to the reference simulations consisting of all cycles.).

Fig. 1.1 shows an example of a state variable that changes as a function of time. The state variable can be any meaningful measurable quantity in the finite element simulation, for example the stress or the deformation at a point. As can be seen from the figure, a

global and local change can be observed. The local variation is a high frequency variation corresponding to the change during one cycle and the global change is the general trend that can be extracted over many cycles. The scheme of cycle-jump technique is as follows:

1. Conduct several cycles with finite element analysis (FEA) to establish the overall trend, e.g., global evolution function $y(t)$;
2. Extrapolate selected state variables (e.g. components of stress, strain, displacement) using the global evolution functions;
3. Impose the extrapolated state as the initial state for a new finite element analysis after the cycle jump;

The length of the jump (i.e. number of cycles) is calculated by the "control function," as defined in ref [5]. In the numerical simulation, some state variables will evolve due to cyclic loading and these state variables can be selected as the state variable used to establish the control functions. First, at least three values of the state variable are extracted after the simulation at times corresponding to the same relative time in each individual cycle. These points are defined as $P_1(t_1, y_1), P_2(t_2, y_2), P_3(t_3, y_3)$ in Fig. 1.2. From these points, slopes are obtained defined as $s_{12}(t_1) = \Delta y(t_1) / \Delta t_{cycle}$ and $s_{23}(t_2) = \Delta y(t_2) / \Delta t_{cycle}$, where $s_{12}(t_1)$ is the slope of the line connecting point $P_1(t_1, y_1)$ and $P_2(t_2, y_2)$, $s_{23}(t_2)$ is the slope of the line connecting point $P_2(t_2, y_2)$ and $P_3(t_3, y_3)$, and $\Delta t_{cycle} = t_1 - t_2 = t_2 - t_3$ represents the cycle length (time).

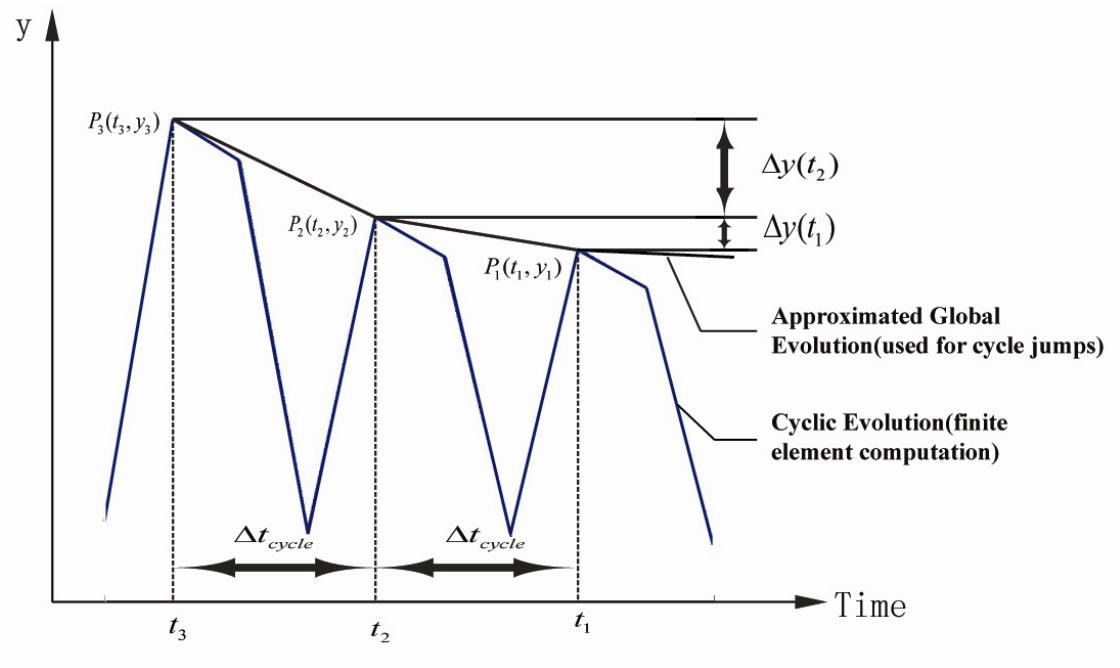


Fig. 1.2 A schematic of an evolving state variable, y, for a structure subjected to cyclic loading. (Adopted from Cojucaru and Karlsson[5])

The allowed jump length for each extrapolated parameter is dictated by the following criterion:

$$\frac{\left|s_p(t_1 + \Delta t^M_{y,jump}) - s_{12}(t_1)\right|}{\left|s_{12}(t_1)\right|} \le q_y \tag{1.1}$$

where q_y is a relative error ($q_y \ge 0$), $\Delta t^M_{y,jump}$ is the time spanned by the jump for material point M, $s_p(t_1 + \Delta t^M_{y,jump})$ is the predicted slope at the moment after the jump, obtained by linear extrapolation as:

$$s_p(t_1 + \Delta t^M_{y,jump}) = s_{12}(t_1) + \frac{s_{12}(t_1) - s_{23}(t_2)}{\Delta t_{cycle}} \Delta t^M_{y,jump} . \tag{1.2}$$

The value of allowed jump length is now easily obtained:

$$\Delta t_{y,jump}^{M} = q_y \Delta t_{cycle} \frac{|s_{12}(t_1)|}{|s_{12}(t_1) - s_{23}(t_2)|} . \tag{1.3}$$

The value of q_y is specified by the user and may vary over time. Note that this error is not the error the cycle-jump technique results in, but can be viewed as an error acceptable by the user. In the following, q_y is referred to as the "control parameter."

In general, $\Delta t_{y,jump}^{M}$ will be unique for each material point and variable. Thus, we select the jump length to be the minimum $\Delta t_{y,jump}^{M}$ defined as:

$$\Delta t_{jump} = \Delta t_{cycle} \left\lfloor \min\left\{ \Delta t_{y,jump}^{M} \right\} / \Delta t_{cycle} \right\rfloor . \tag{1.4}$$

This routine has been successfully used numerically to conduct cycle jumps in cyclically loaded structures [5] where it was implemented using the commercial program ABAQUS and PYTHON code. However, it has not been compared to experimental results.

1.3.2 Thermal Barrier Coatings

We will here give a short overview of Thermal Barrier Coatings (TBCs) [1,14-19]. TBCs are used in the hot sections of gas turbine engines to protect the underlying structure from the high operating temperatures. They are subjected to a complicated cyclic load, combining both thermal and mechanical loading.

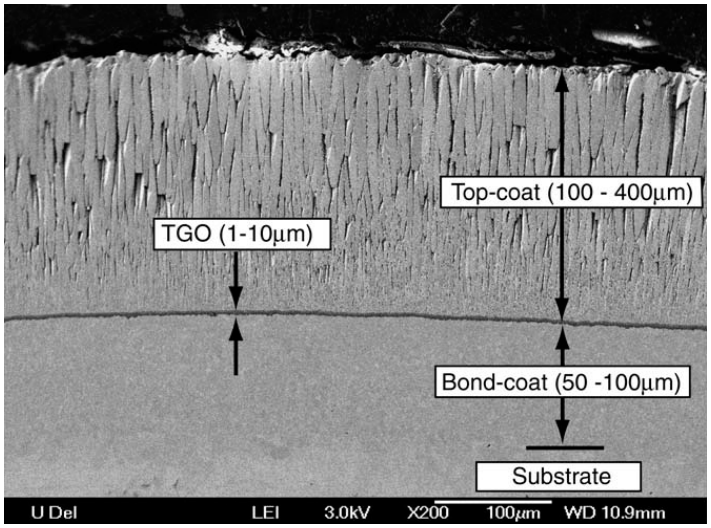


Fig. 1.3 Example of a thermal barrier coating (courtesy Jin Yan, University of Delaware and Marion Bartsch, the German Aerospace Center).

TBC consist of two layers (Fig. 1.3): (i) a metallic, aluminum rich bond coating, which prohibits oxidation of the superalloy structure; and (ii) a ceramic topcoat, which reduces heat transfer. Between these two layers, a thermally grown oxide (TGO) develops due to the oxidation of the bond coat during high temperature exposure. Primarily, the TGO consist of aluminum oxide (alumina, Al_2O_3). The TGO is initially less than 0.5 μm and grows up to 7-10 μm before failure. Due to the growth and the presence of the TGO, cracks will initiate, grow and coalesce in its vicinity, which eventually results in final failure of the structure. Moreover, since the bond coat provides aluminum to the TGO, the properties of the bond coat will change as the TGO grows.

1.4 Objectives of Our Work

The overall objective of our work is to develop a numerical scheme that can be used to develop life prediction models for structures subjected to cyclic loading.

In what follows, the enhancement of the cycle-jump technique code will be discussed and the results obtained by running the augmented cycle-jump technique code are compared to preliminary experimental investigations.

The modifications consist of two main parts: one is to extend the cycle-jump technique to include creep properties, and the second is to extend cycle-jump technique to include displacement as the loading. Furthermore, some of the PYTHON code is changed into C++ code [20] to further improve the calculation efficiency. After improving the cycle-jump technique code, the code is used to design the load sequence of the experiments that need to be conducted to verify the cycle-jump technique. A sensitivity analysis is conducted since a small variance of material properties in the real specimen are to be expected. Finally the numerical results are compared to the preliminary experimental results.

Chapter 2

ENHANCEMENTS OF THE CODE FOR THE CYCLE-JUMP TECHNIQUE

2.1 Including Creep in the Model of the Cycle-Jump Technique

TBCs are subjected to a complicated cyclic load combining both thermal and mechanical loading, and to model this complicated load cycle for multiple cycles can be very time consuming. Therefore, we will investigate if the cycle-jump technique can be realized to increase the computational efficiency when simulating the structure. In order to use the method for TBCs, we need to extend the current computational scheme to incorporate creep properties. To this end, we define an FE-model with time dependent properties. A reference simulation is conducted where all cycles are calculated. The result obtained by the cycle-jump technique code is compared with the reference simulation to numerically verify the technique. We note that none of the results from the reference simulations are used for the cycle-jump technique. These are two separate simulations then are compared after they are completed to evaluate the success of the cycle-jump technique.

2.1.1 Basic Finite Element Model

We will first investigate if time dependent properties can be incorporated into the

cycle-jump technique; to this end we investigate a two-dimensional finite element model. The commercially available program ABAQUS [2] is used in our simulations. We will analyze a plane-stress specimen and the initial width is 1mm. To reduce the model size, the model in our simulation is symmetric about the vertical and horizontal axes and only ¼ of the cross section is meshed. Bilinear plane-stress quadrilateral elements (CPS4) are used for the entire domain. The generated FE model contains 121 nodes and 100 elements. Models with finer meshes are also simulated, and the results were the same. Thus, the coarser mesh is used for the simulation to save computational time. The boundary conditions are shown in Fig. 2.1(A). Two sides of the model are constrained to simulate symmetry and the model is under cyclic loading. A time-hardening creep-law is adopted [21]:

$$\frac{d\varepsilon}{dt} = A\sigma^n \tag{2.1}$$

where ε is the creep strain, t is time, σ is the current stress, A is the creep pre-factor and n is the creep exponent. Thus, A and n are material properties. The material properties used in the initial simulations are shown in Table 2.1[22].

The model is stress-free initially and then subjected to a cyclic in-plane force. One cycle consists of three parts: (1) loading to the maximum force $F_{\max}$, (2) holding the force for several seconds, and (3) unloading the force to zero. The loading sequence is illustrated in Fig. 2.1(B).

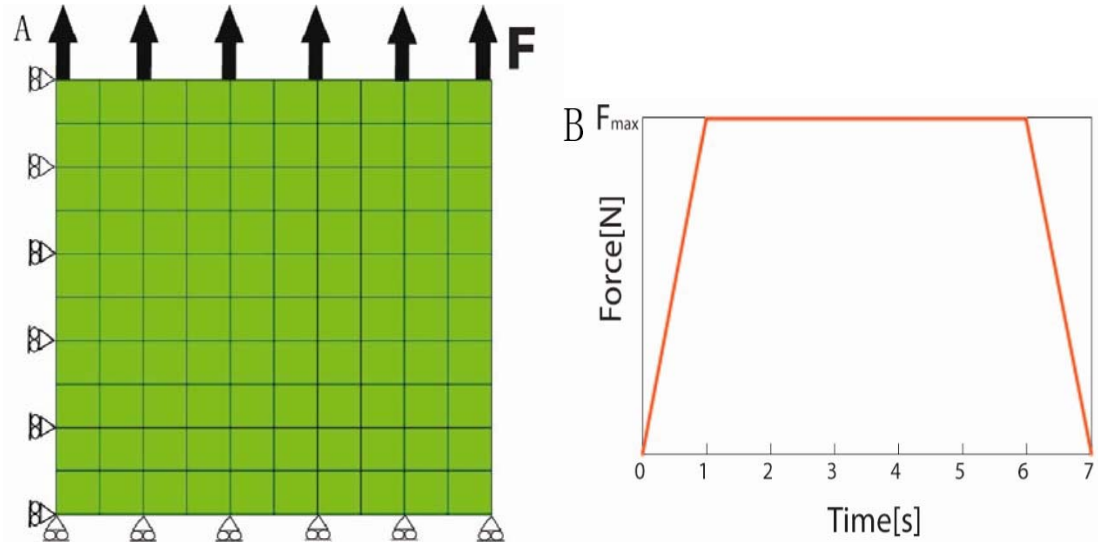


Fig. 2.1 (A) Schematic of the finite element model used for cycle-jump with creep properties. (B) A schematic of one cycle of loading.

Elastic modulus E (MPa)	$167,000$
Creep Exponent (n)	3.0
Creep pre-factor $(A)[MPa^{-n}s^{-1}]$	5.31×10^{-9}

Table 2. 1 Material properties used [22]

2.1.2 Results

2.1.2.1 Reference Simulation

Reference simulations were conducted which include all individual cycles and steps during the complete simulation time. The results are used to compare with the results obtained from the "cycle-jump technique."

Since the cycle-jump technique is used to predict the general trend of a state variable, we are interested in simulations where a state variable evolves clearly over time. Three cases are simulated. The first case has 70 cycles of loading with a maximum force of 40N corresponding to a maximum nominal stress of $40MPa$. The second case has 70 cycles

of loading with a maximum force of 60N ($60MPa$). In the first two cases, with the increase of the maximum loading, the global trend will change from linear to nonlinear. In the third case, the simulation time is increased and the total cycle of loading is 150, and with maximum force 40N. The holding time is set for 5 seconds and loading/unloading time is one second.

The effective stress according to von mises, "Mises stress," is defined by:

$$\sigma = \sqrt{\frac{1}{2}[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2]} \tag{2.2}$$

where σ_1, σ_2 and σ_3 are the principal stresses. As can be seen in Fig. 2.2(A), the Mises stress starts at zero, increases with increasing force, and reaches a maximum when maximum force is reached and decreases as the structure is unloaded to zero force. This is the local response for the structure and it is repeated for every cycle. The maximum stresses change with every cycle and this trend is the global evolution of the structure. The global trend of the stresses is increasing. This is due to the fact that, even though the maximum force remains the same, the area of the cross section is decreasing under cyclic loading which leads to the increasing of the stress. For this case, the creep strain (Fig. 2.2(B)) and the total strain (Fig. 2.2(C)) accumulate with each cycle and they change approximately linearly.

The force for the second case is 60N. As can be seen in Fig. 2.3(A), the global trend of the Mises stress, changes nonlinearly. For this case, the creep strain (Fig. 2.3(B)) and the total strain (Fig. 2.3(C)) accumulate with each cycle and also change nonlinearly.

As can be seen from the two cases, with a lower maximum force (40N), the general trend of the Mises stress do not change significantly. As the loading force increases, the global trend of the Mises stress changes from linear to nonlinear.

We also consider the cases to investigate the effect of holding time where we changed the holding time from 5 to 10 seconds. The results for 5 seconds and 10 seconds are similar, so we only present the results for 5 seconds here for simplicity.

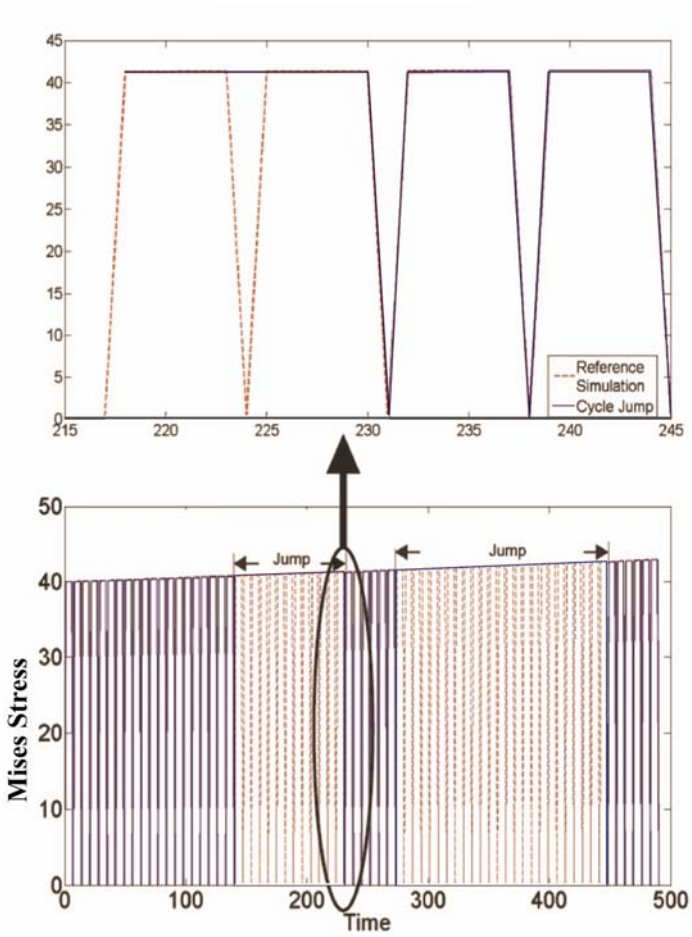
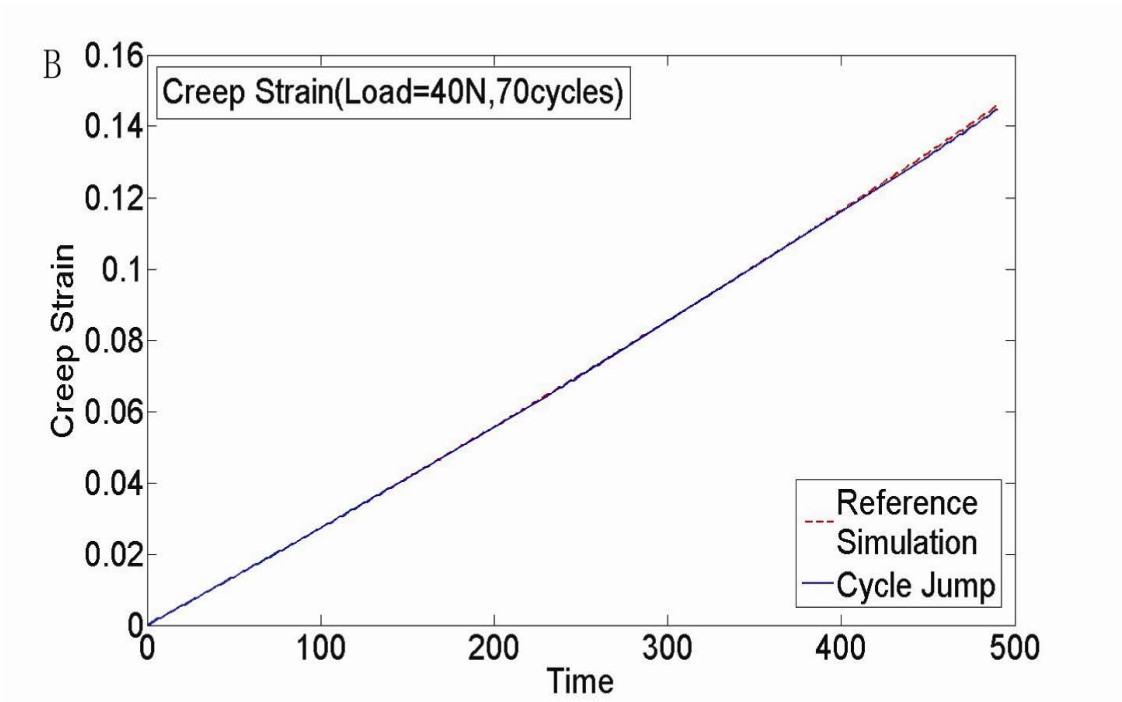


Fig. 2.2 (A) Stress as a function of time for load=40N. (B) Creep strain as a function of time for load=40N. (C) Total strain as a function of time for load=40N.



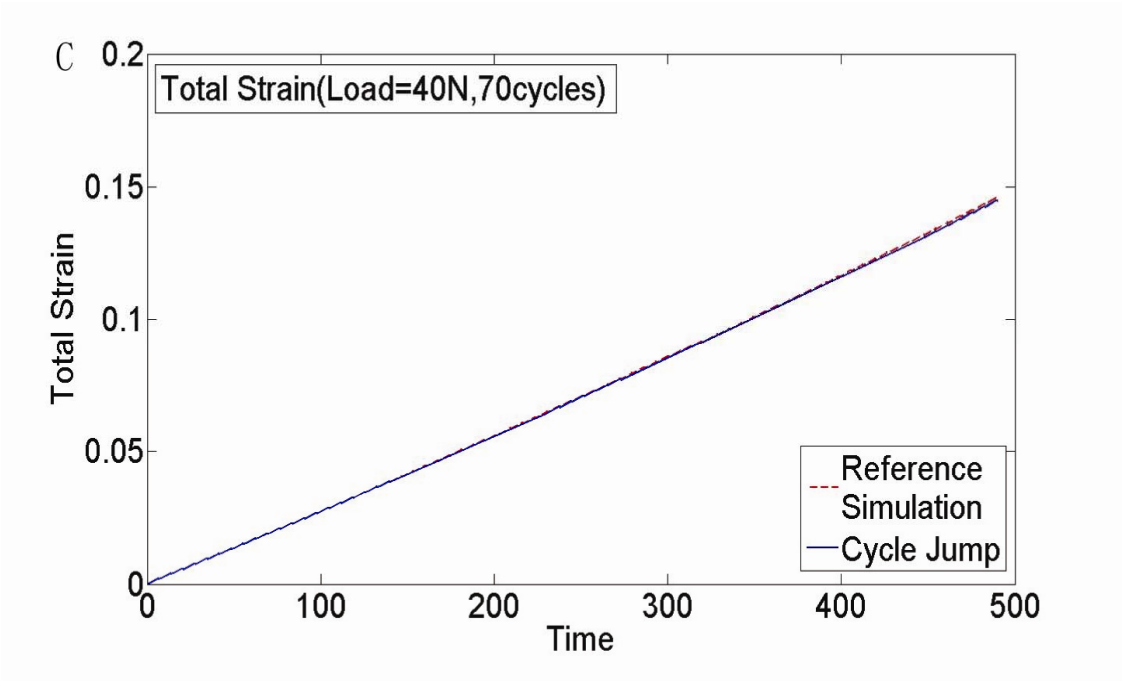


Fig. 2.2 Continued

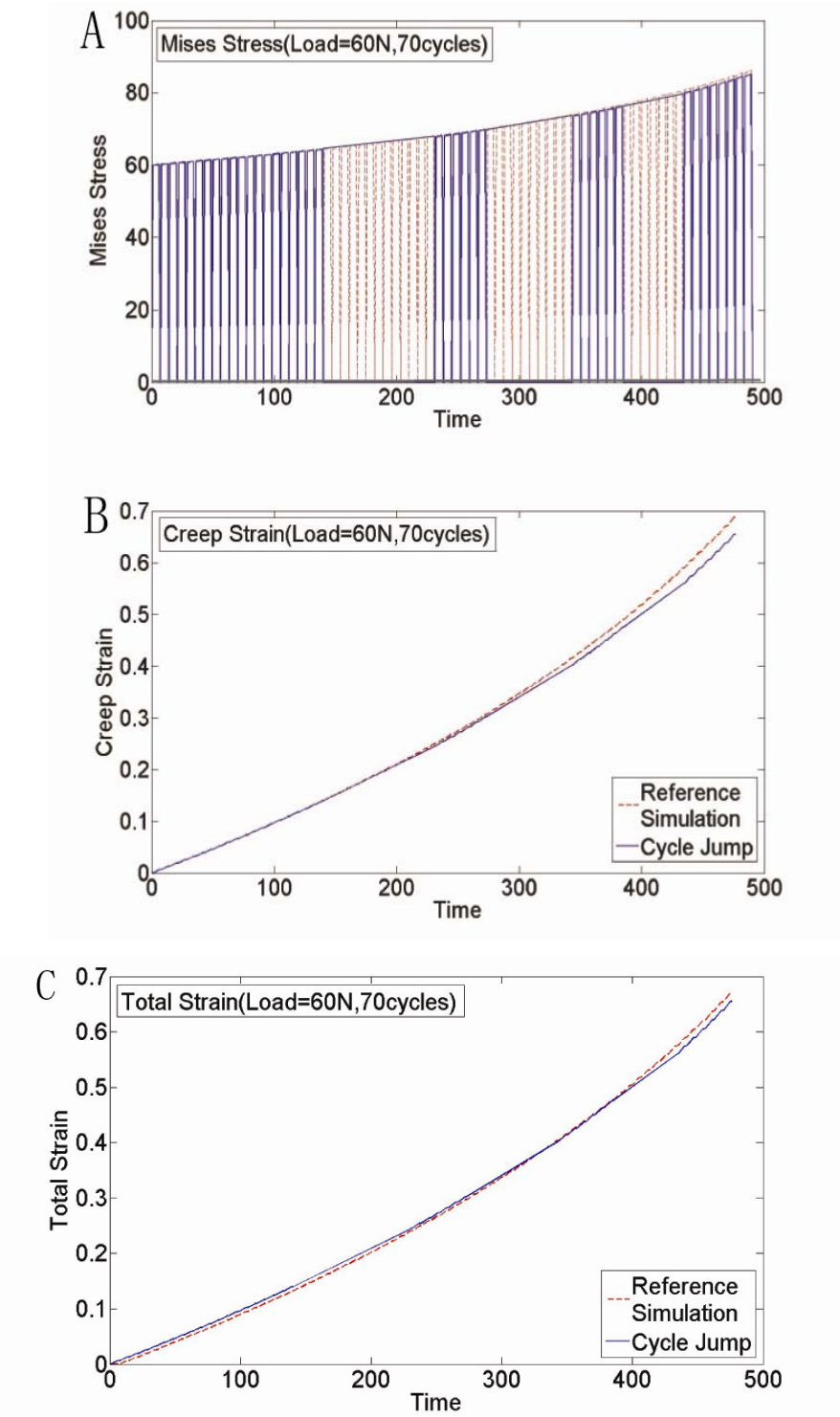


Fig. 2.3 (A) Mises stress, (B) Creep strain and (C) Total strain as a function of time for load=60N.

2.1.2.2 Cycle-Jump Simulation

We will now investigate if the cycle-jump technique can capture the behavior seen in the cycle by cycle simulations.

First, the two reference simulations we discussed above are investigated. From Fig. 2.2 and 2.3 we can see the results of the cycle-jump technique compared with the reference simulation. Note that none of the information obtained in the reference simulations was used in the cycle-jump simulations. The results from the two separate simulations are plotted together after completion of the numerical simulations for comparison.

From the two cases it can be seen that:

(1) As the maximum force increases, the global trend of the state variables changes from linear to nonlinear. The cycle-jump technique captures the nonlinear behavior of the state variables successfully.

(2) As can be seen from the figures, the simulations with the linear results are more accurate than the nonlinear cases when compared to the reference simulations.

(3) The jump "length" of the simulation with linear results is longer than those nonlinear cases. With both cases running for 70cycles, it can be seen that for the 40N case, there are two jumps while in the 60N case there are three jumps, thus the linear cases need less computational time.

Simulations with longer time (more cycles) are also investigated. The model is simulated with up to 150 cycles of loading. The maximum force here is 40N.

It can be seen from Fig. 2.4 that the results of the cycle-jump technique capture the reference simulation and the technique captures the nonlinear behavior of the state variables successfully.

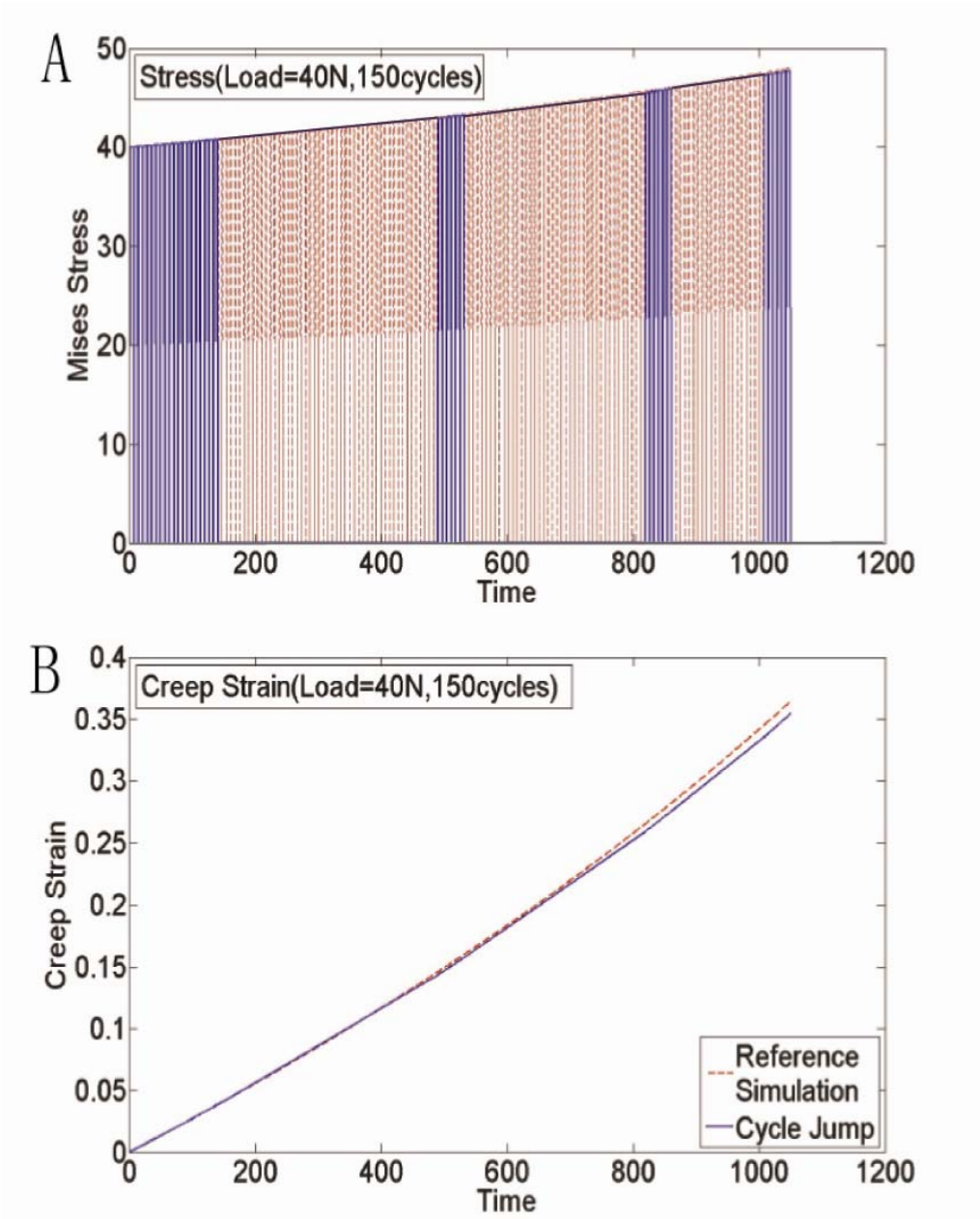
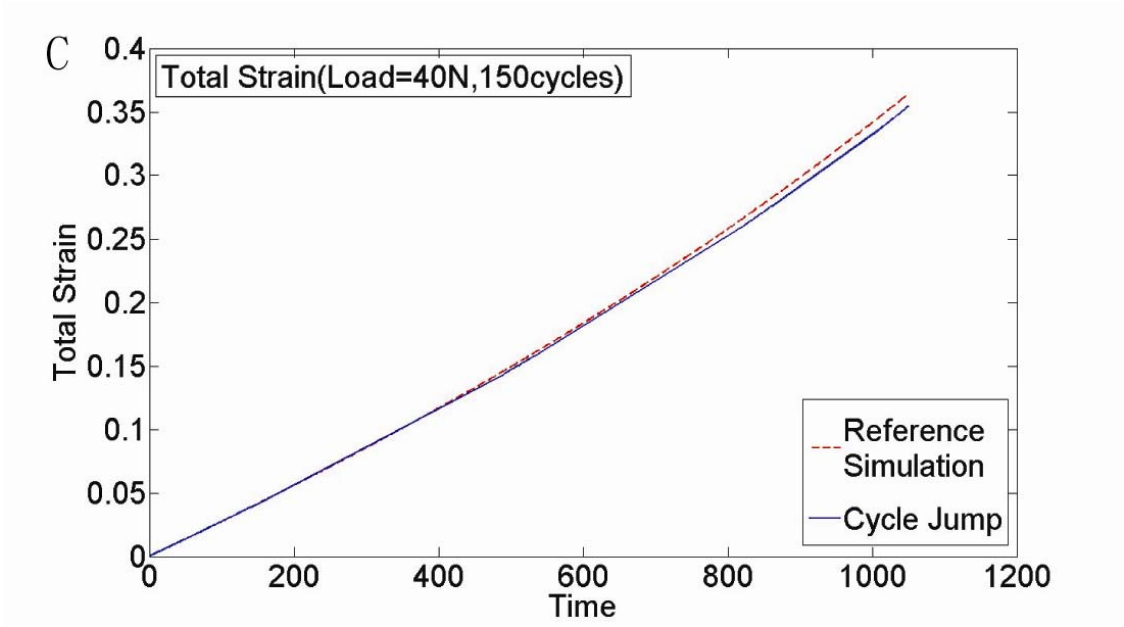


Fig. 2.4 (A) Mises stress, (B) creep strain, (C) Total strain and (D) Displacement as a function of time for load=40N,150cycles.



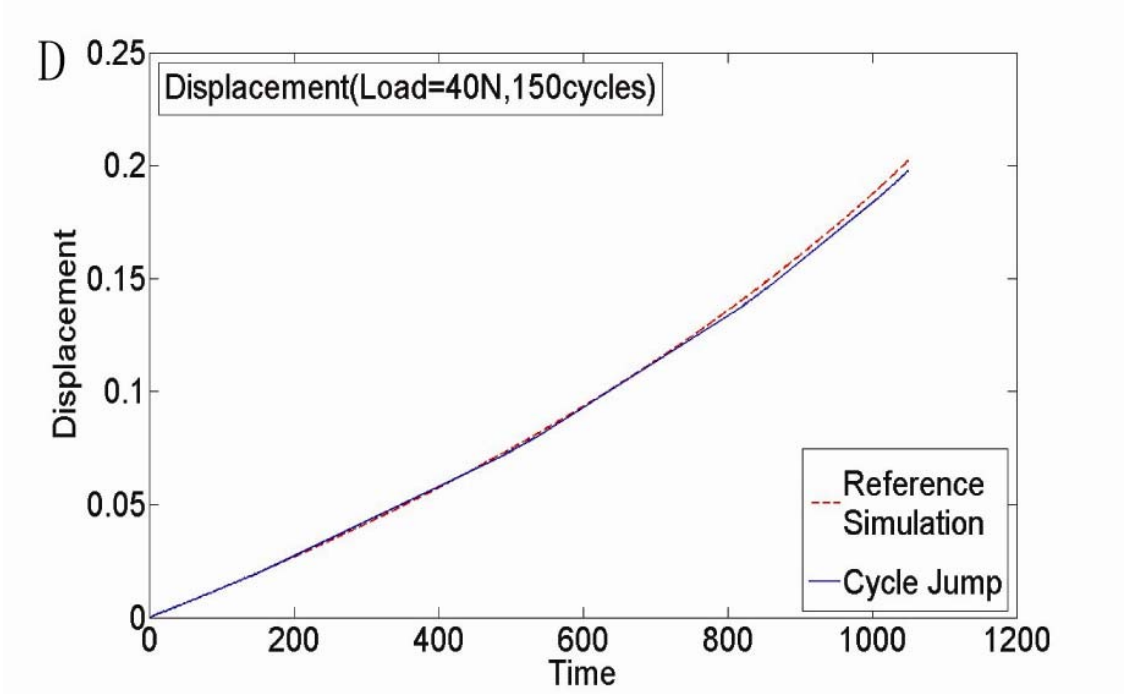


Fig. 2.4 Continued

2.1.2.3 Influence of the Control Parameter

The influence of the control parameter q_y on the results is investigated in this section. This parameter, as discussed previously, can be considered as an error that the user is willing to accept. The control parameter is here used to control the acceptable error on the state variable and denoted q in the following for simplicity. Four different q (q=0.25, 0.5, 1, 2) are used to run the simulations with 150 cycles and force of 40N.

Comparisons of the results with the selected control parameter q are shown in Fig. 2.5. Lower q results in higher accuracy. The relative error between the cycle-jump simulations and reference simulations are calculated and compared. It can be seen from Fig. 2.6 that with smaller q, the relative error is smaller thus the results are more accurate as expected. Simulations with q=2 and 3 are conducted and the results are the same as the results with q=1.

However, the computational time decreases with the increasing of q. It is therefore important to select q to optimize between accuracy and computational time.

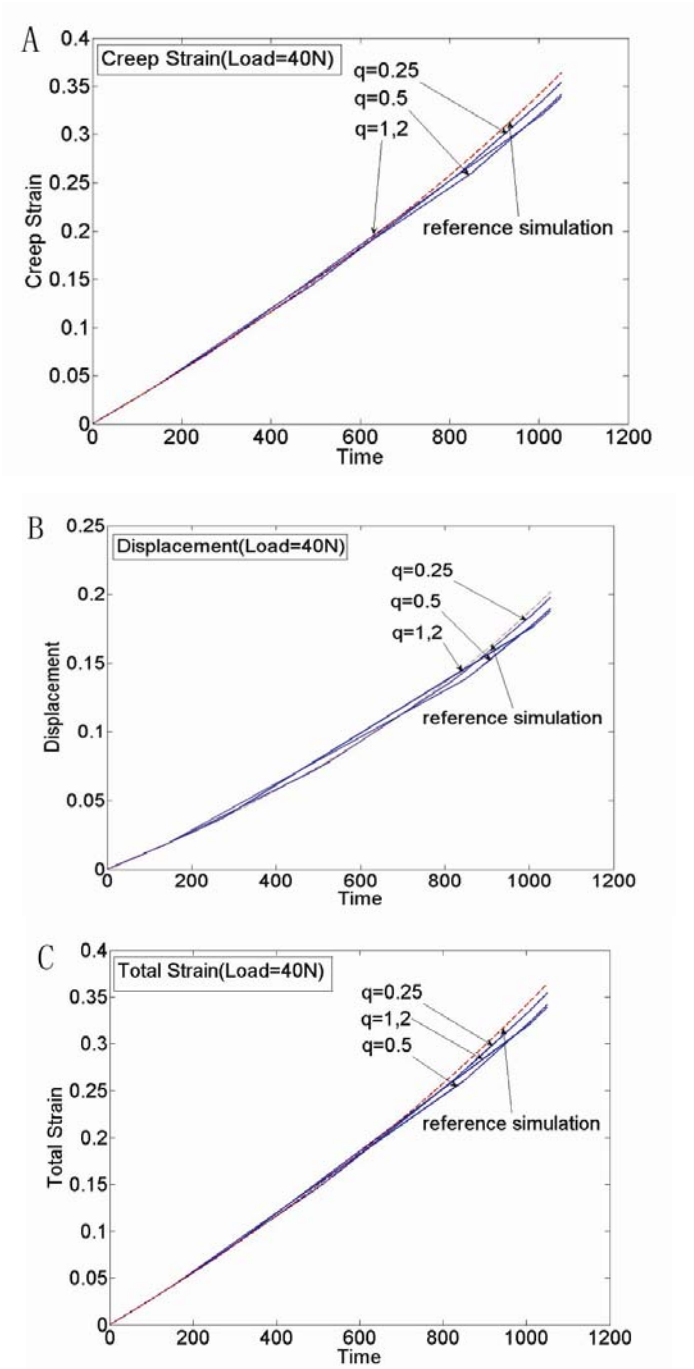
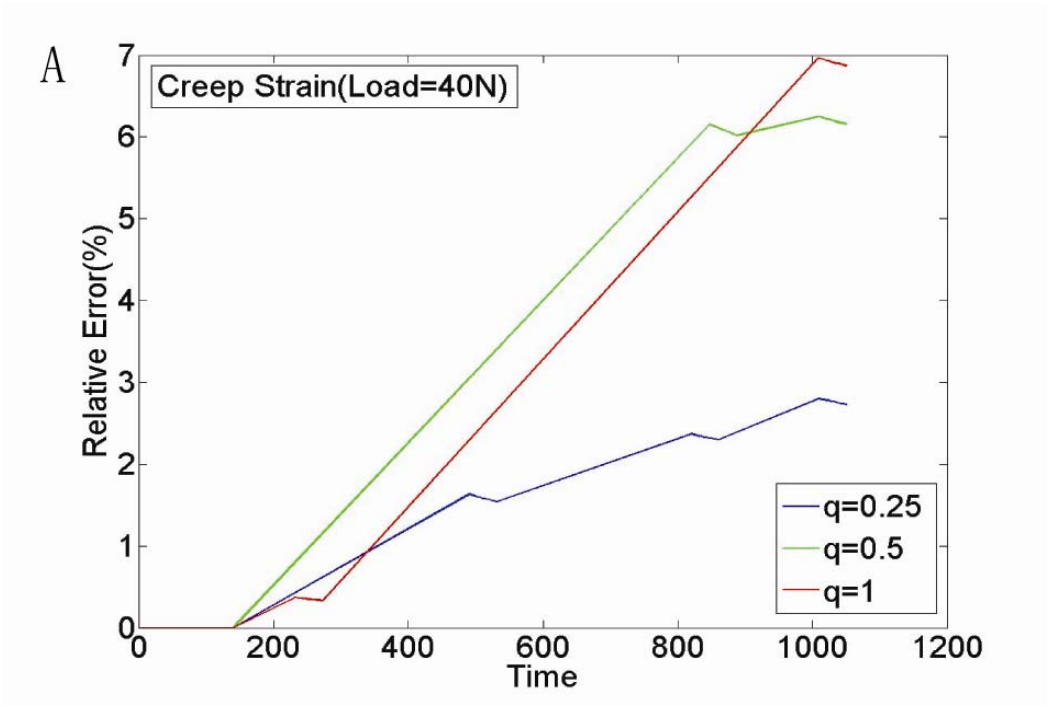


Fig. 2.5 (A) Creep strain, (B) Displacement and (C) Total strain as a function of time for q =0.25, 0.5, 1 and 2, load=40N, 150cycles.



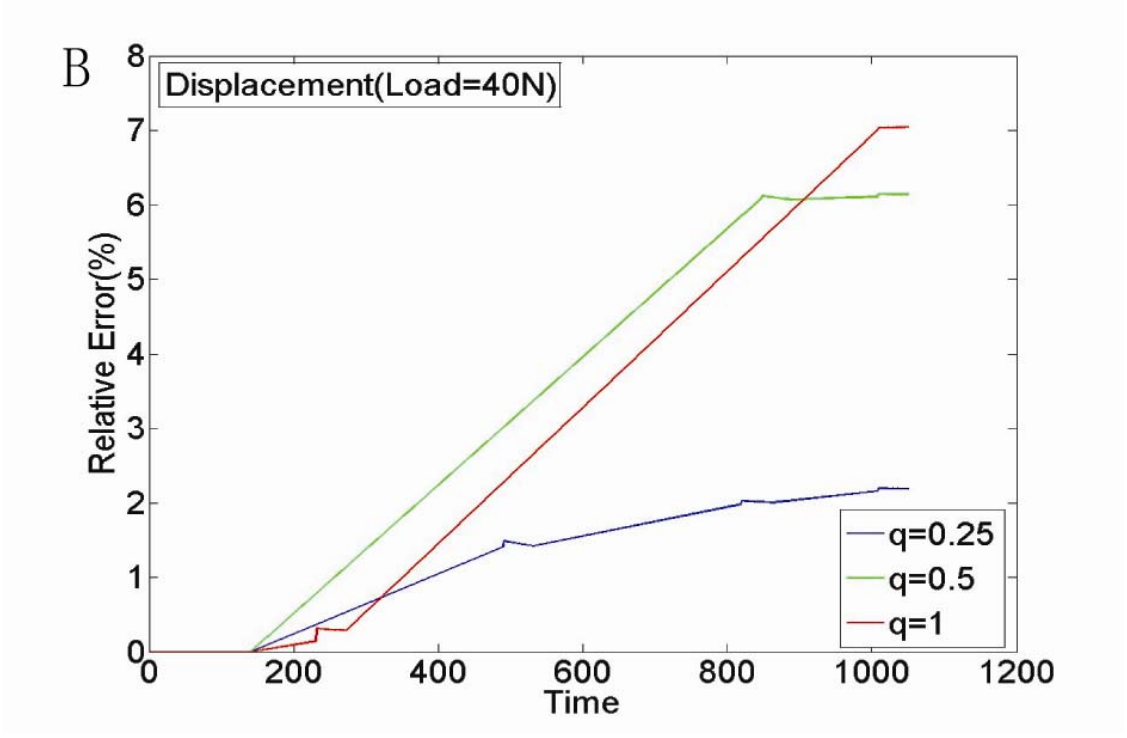


Fig. 2.6 Relative error of (A) creep strain and (B) displacement as a function of time for q=0.25, 0.5, 1 and 2, load=40N, 150cycles.

2.2 Displacement Controlled Loading

2.2.1 Description of the Change of the Model

In the previous model, force controlled loading was used. However, displacement controlled loading is commonly used in real experiments. Thus, we will augment the cycle-jump scheme to allow for displacement controlled loading

As previously discussed, the procedure of the cycle-jump technique is:

1. Conduct several cycles with finite element analysis (FEA) to establish the overall trend, e.g., global evolution function $y(t)$;
2. Extrapolate the state variable using the global evolution function;
3. Impose the extrapolated state as the initial state for a new finite element analysis after the cycle jump;

After extrapolating the state variable, the predicted values of the state variable must be imposed as initial conditions for a new finite element analysis after the cycle jump. For the case of the force controlled loading, the displacement was extrapolated and imposed as the initial state. This was accomplished via user's subroutine "DISP" [3,23]. In the case of displacement controlled loading, both stress and creep strain must be imposed as the initial state. It is straight forward to impose the extrapolated stress as the initial state (i.e. set the extrapolated stress as loading) but there is no specific user's subroutine that allows us to impose the creep strain.

2.2.2 Imposing Creep Strain

In order to impose the creep strain as the initial state, we need to "trick" the program to accept the predicted creep strain. Thus, a method is developed based on utilizing an "intermediate initiation simulation (IIS)" after the jump to impose the creep strain. Thus, this is done before the new finite element analysis and after the cycle jump. As a result, the new procedure of the cycle jump is as follows:

1. Conduct several cycles with finite element analysis (FEA) to establish the overall trend, e.g., global evolution function $y(t)$;
2. Extrapolate the state variable using the global evolution functions;
3. Run the intermediate initiation simulation (IIS) as step one of the new finite element analysis after the cycle jump. This gives the correctly (predicted) state of stress and strain as new initial conditions;
4. Initiate a regular finite element analysis after the cycle jump.

The extrapolated stress, σ_E, and creep strain, ε_E, after the cycle jump are determined based on first establishing the global evolution functions and then imposing these functions on the stress and strain to obtain the extrapolated value σ_E and ε_E. In the IIS, σ_E is constant. After integrating equation (2.1), we get (2.3).

$$\varepsilon_E = A\sigma_E^n t_{IIS} \tag{2.3}$$

Since ε_E and σ_E were established via the extrapolation, we can calculate out the time t_{IIS} needed for the creep strain to accumulate to ε_E. Thus, by conducting the IIS (illustrated in Fig. 2.7) using σ_E as the loading and t_{IIS} as the step time, after time t_{IIS},

the predicted ε_E will be "captured." After this initiation simulation, the program continues to run the regular simulation after the cycle jump, with the initial conditions being the extrapolated state variables.

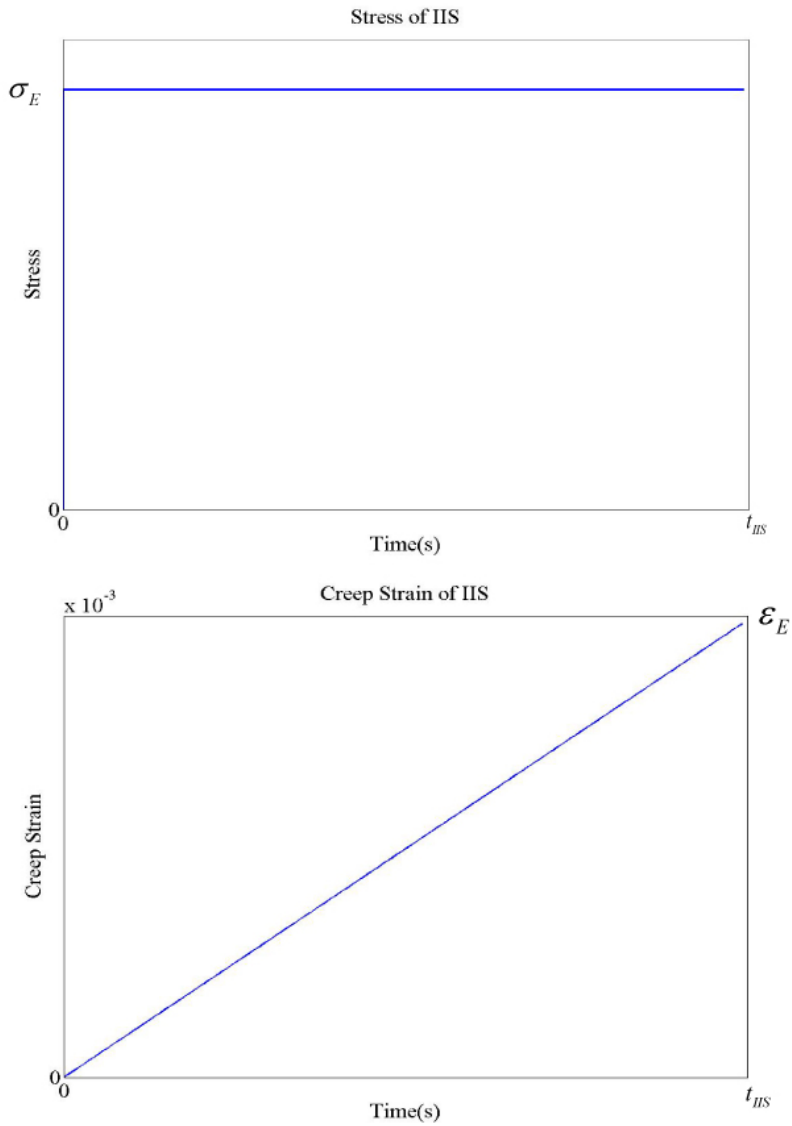


Fig. 2.7 Intermediate Initiation Simulation (σ_E is the predicted/extrapolated stress, and ε_E is the predicted/extrapolated creep strain)

To further improve the efficiency of the simulation, two sets of material properties (i.e. A in equation (2.1)) are used. Define $A_{\exp}$ as the real material property of the investigated structure and A_{IIS} as the material property used in the IIS to improve the efficiency. A_{IIS} is determined as follows:

$$A_{IIS} = A_{\exp} * 10^{n} \tag{2.4}$$

where n is an integer and $n > 0$. With the same ε_E in equation (2.3), if A_{IIS} is used instead of $A_{\exp}$, the time t_{IIS} can be reduced by a factor of 10^{n}.

2.2.3 Verifying Simulations of Displacement Controlled Loading

We now show that the cycle-jump technique works for displacement controlled loading including creep properties, by comparing results from cycle-by-cycle simulations to results obtained by the cycle-jump technique.

The same overall geometry and boundary conditions as described in section 2.1 are used. However, the applied force is replaced with an applied displacement at the upper boundary. Two cases are investigated and compared to the cycle-by-cycle simulation. In both cases the maximum displacement of 0.03mm is used. The holding time is 60s and the total cycle number is 180 for the first case (Fig. 2.8). The holding time is 120s and the total cycle number is 90 for the second case (Fig. 2.9). It can be seen from the figures that the modified code allowing for displacement controlled loading captures the nonlinear behavior of the stress and creep strain successfully and the results of the cycle-jump technique agree well with the results of cycle by cycle simulation.

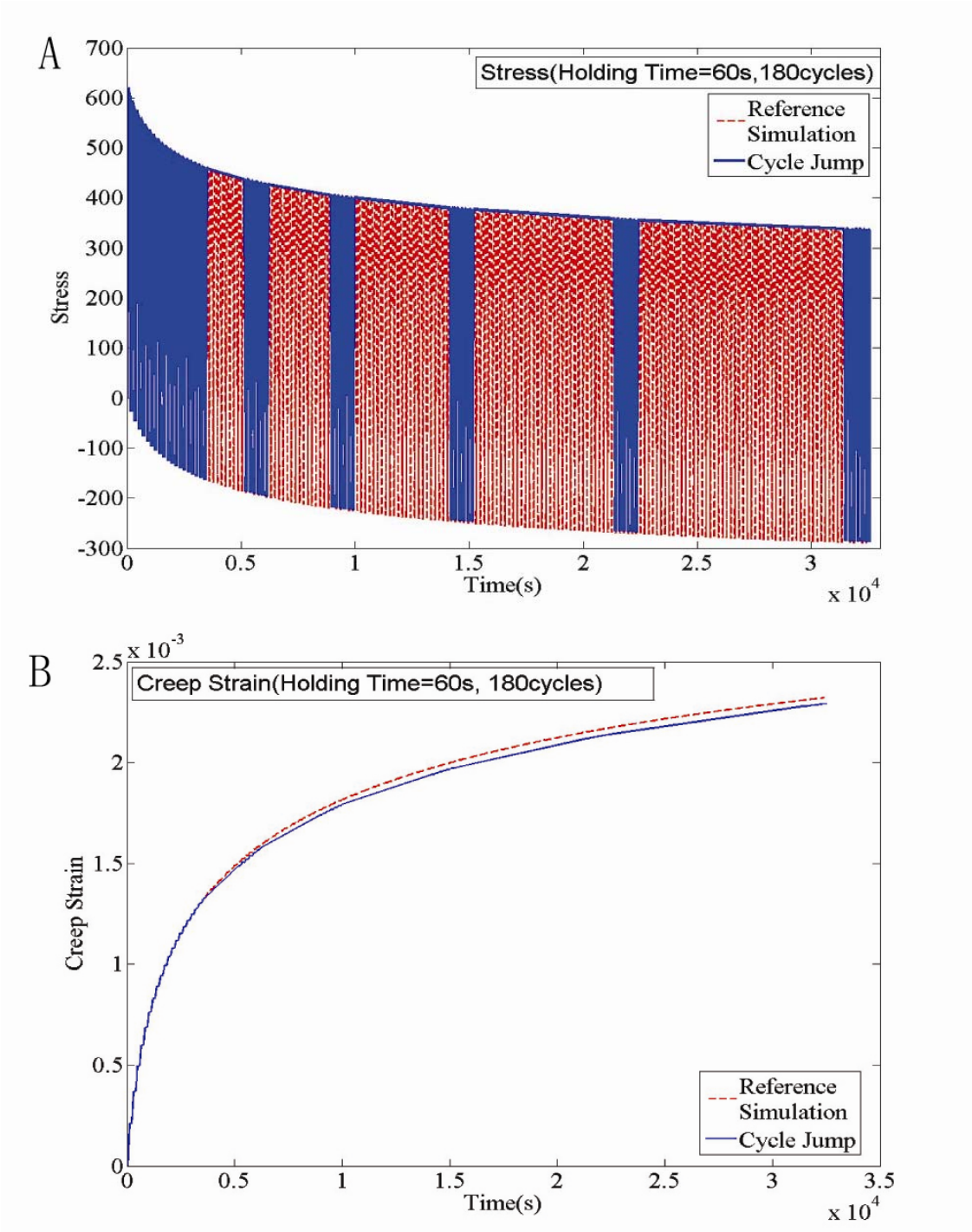


Fig. 2.8 (A) Stress and (B) Creep strain as a function of time for displacement=0.03mm, 180 cycles.

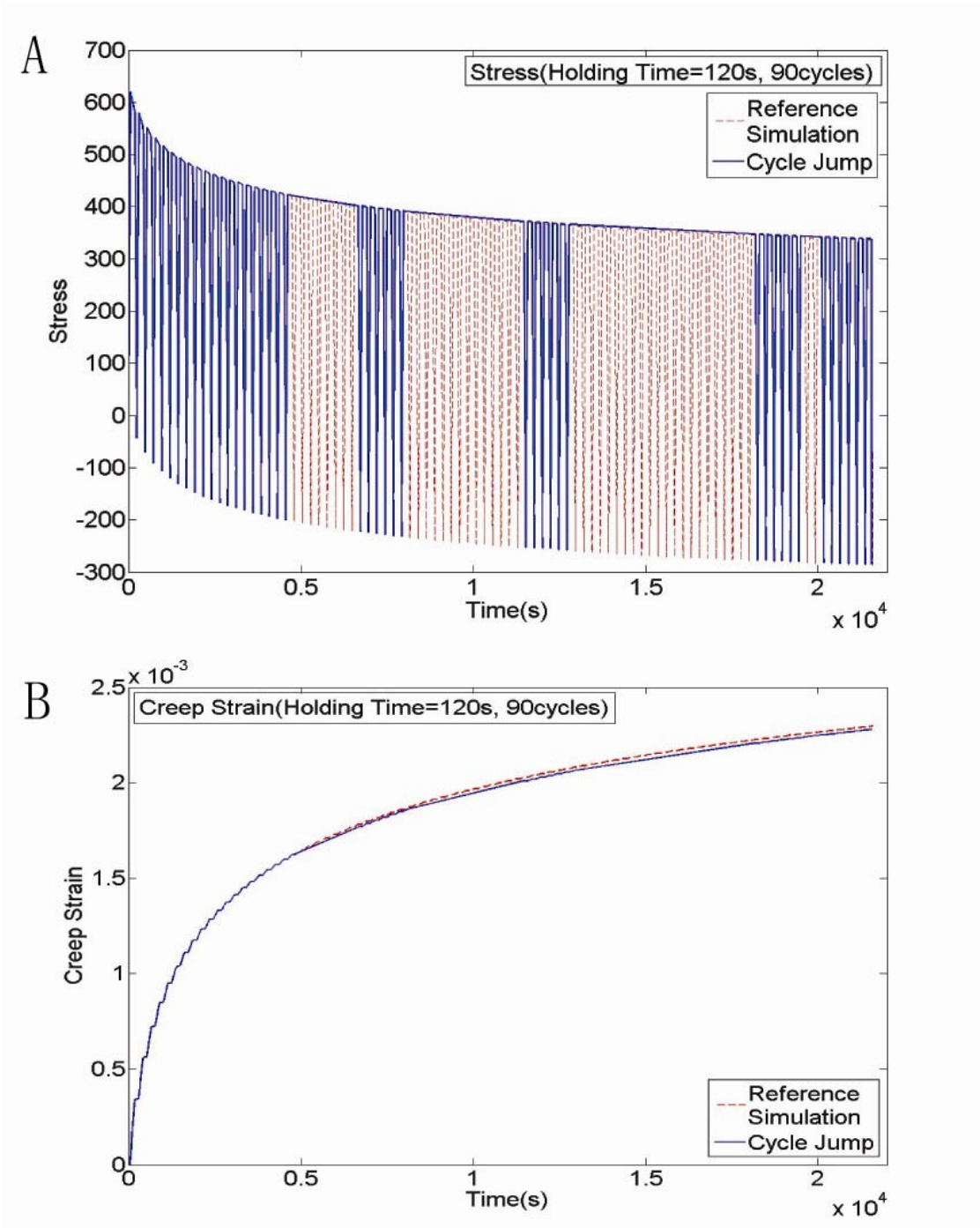


Fig. 2.9 (A) Stress and (B) Creep Strain as a function of time for displacement=0.03mm, 90 cycles.

Chapter 3

DESIGN OF THE EXPERIMENTAL LOAD SEQUENCES

3.1 Overall Procedure

The cycle-jump technique is based on a purely numerical scheme, and has no particular physical meaning. However, the scheme has been successfully verified numerically by comparing cycle-by-cycle simulations to cycle-jump simulations. As seen in Chapter 2, excellent agreement is obtained. Thus, numerically, the cycle-jump technique predicts the correct evolution ("if you believe that cycle-by-cycle FEA gives correct results, the cycle-jump scheme is correct").

However, the cycle-jump technique code has not been used for real life, practical problems. Thus, the cycle-jump technique must be verified for real problems. This will be achieved by comparing the computational results with experiment results. If these two sets of results agree with each other, the cycle-jump technique can be considered a viable code and can be used to predict the evolution of real engineering structures.

In this chapter, a framework for developing a test scheme with the purpose of verifying the cycle-jump technique will be outlined and developed. The proposed framework will be compared to a limited set of experimental data. The following

framework is proposed to verify the cycle-jump technique, which combines experimental and numerical work.

(1) Select a material system and geometry to test.

(2) Conduct material testing to determine material properties, including creep-relaxation test to determine time dependent properties.

(3) Develop a FE-model that corresponds to the geometry to be tested in cyclic loading, including the established material properties. From this model, establish a suitable load sequence for the experiments. This includes time for each cycle: time to load up to maximum load, time to hold at constant load, time to unload and time at minimum load. This will give guidelines for the experimental investigations.

(4) Conduct cyclic testing based on the numerical simulations.

(5) Compare the results from the numerical model to the experimental work. The model will incorporate the cycle-jump technique. The results from the cycle-jump technique simulations should agree with the experimental results.

(6) The cycle-jump technique will be verified if the simulations agree with the experimental results.

3.2 Experimental Conditions

The material system and test setup selected for verifying the cycle-jump technique will now be discussed. When selecting a system to investigate, it is important to ensure that the material system investigated experimentally in the verifying tests has one and only one parameter that evolves during the cyclic loading. In this case, we selected to

investigate a material under elevated, but iso-thermal conditions, subjected to cyclic mechanical loading. By holding the specimen at constant load for a specified time, it will "evolve" by creep. The material selected is a material commonly used as the bond coat in TBC and is referred to as an "MCrALY". A time-hardening law is adopted for creep according to:

$$\frac{d\varepsilon}{dt} = A\sigma^n \tag{3.1}$$

The tests will be conducted at $870\,^0C$, from which we evaluate the material properties, Table 3.1. The gauge length and associated geometry of the specimen used in the cyclic testing are described in Fig. 3.1. The tests are conducted in collaboration with Dr. Marion Bartsch at the German Aerospace Center (DLR).

Elastic Modulus (E)[MPa]	$125,000$
Creep Exponent (n)	7.3607
Creep pre-factor $(A)[MPa^{-n}s^{-1}]$	9.0157×10^{-27}

Table 3.1 Material properties for the MCrALY used based on creep test.

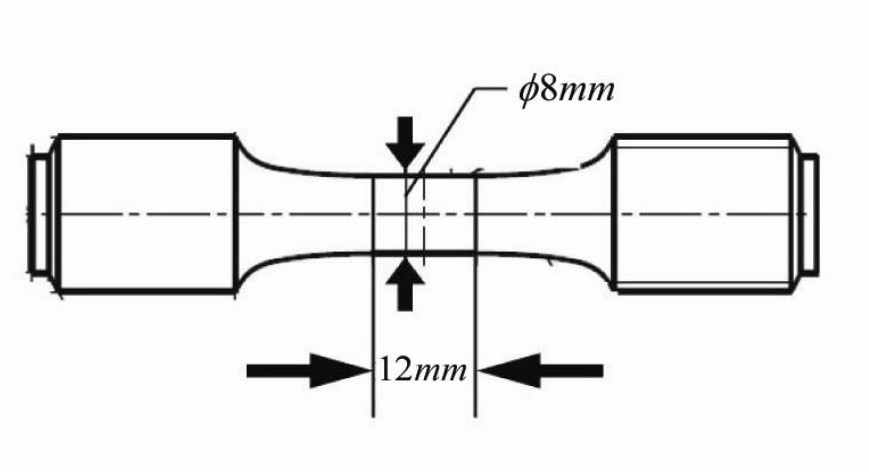


Fig. 3.1 Specimen used in the experiments. (Courtesy Dr. Bartsch, the German Aerospace Center)

3.2.1 Finite Element Model

With the material properties known from previous creep-relaxation tests, finite element simulations are used to propose suitable test schemes (i.e. load sequence). To successfully verify the cycle-jump technique, the load rates need to be optimized so that the material system experiences a global evolution in at least one of the state variables. In the high temperature test conducted here, the specimen is subjected to cyclic loading. The material will creep (or relax) during maximum load in particular if the maximum load is held constant for a specified hold time. In the experiments conducted at DLR, a strain controlled load will be used, thus the stress may relax during the hold. We note that during normal operation, a load that causes creep relaxation will lead to early failures. Here we are only be using this high load to verify the cycle-jump technique.

A finite element model with axisymmetry about the horizontal axis is utilized. The dimensions of the finite element model are shown in Fig. 3.2(A) and are based on the

geometry of the experimental specimen. The boundary conditions are shown in Fig. 3.2(A) and the model is under cyclic loading. Bilinear axisymmetric quadrilateral reduced integration elements (CAX4R) are used for the entire domain. The generated FE model contains 88 nodes and 70 elements. The material properties used in our simulations are evaluated from experiments as shown in Table 3.1.

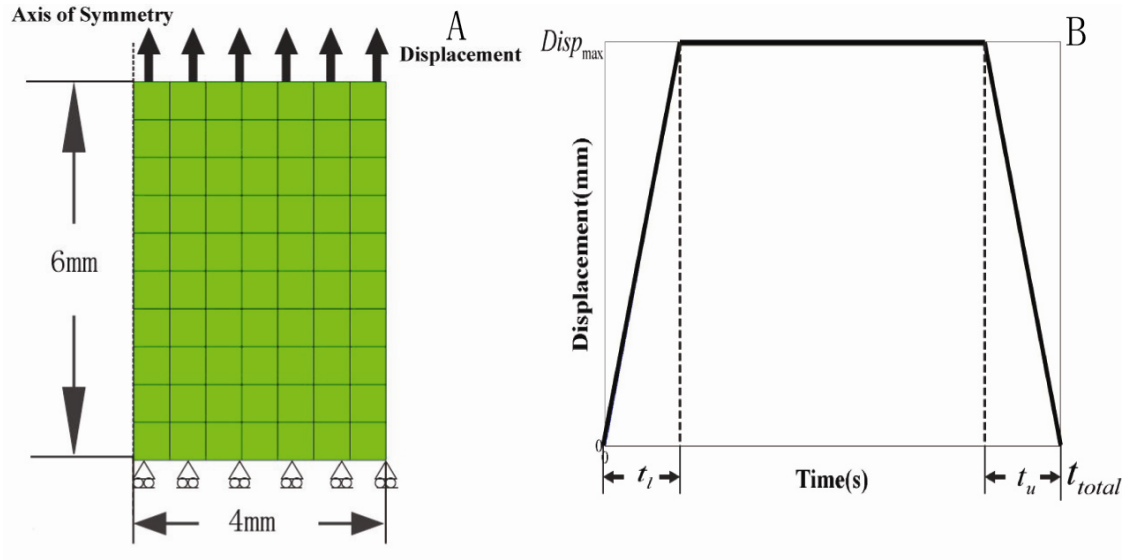


Fig. 3.2 (A) A schematic of the finite element model, including boundary conditions. (B) A schematic of one load cycle , spanning time t_{total}

In the experiment, strain controlled loading is used. For small strains, displacement is in proportion to strain. Dividing the displacement by a constant specimen length, strain can be obtained. Thus, in our finite element analysis, the strain controlled loading is converted into displacement controlled loading. One cycle of loading consists of three parts: (1) loading to the maximum displacement, (2) holding the displacement for several

seconds, and (3) unloading the displacement to zero. The loading sequence is illustrated in Fig. 3.2(B).

Three parameters will be used to define the experiment:

(1) Maximum Displacement;

(2) Holding time;

(3) Loading and unloading time;

In the simulation, the maximum displacement should not be too large to prevent failure of specimen in the experiment. As a result, the maximum displacement should be chosen considering this limitation while global evolution of the stress can be observed. The maximum displacement is set to 0.03mm.

Preliminary numerical investigations showed that the results were insensitive for loading time t_l and unloading time t_u. Thus, these are both set to 60 seconds for simplicity, both for the model and for the experimental results.

The DLR suggested a maximum testing time of 11000 seconds (3 hours) before creep rupture occurs. Based on this, two sets of holding time and the corresponding total number of cycles are investigated as shown in Table 3.2.

Holding time (s)	Total number of cycles
60	180
120	90

Table 3.2 Load sequence data

Fig. 2.8-Fig. 2.9 show the results of the simulation with the holding time and total number of cycles described in Table 3.2. In both cases, a nonlinear global evolution can

be observed. The load sequence data and the maximum displacement in these cases can be considered to be used in the experiments.

3.3 Sensitivity Analysis

In real samples, there is always a small variation of material properties. Therefore, simulations will be conducted with a range of material properties to investigate the sensitivity of the results to different material properties. After manipulating the creep law (3.1), we get:

$$Log(\frac{d\varepsilon}{dt}) = Log(A) + nLog(\sigma) \tag{3.2}$$

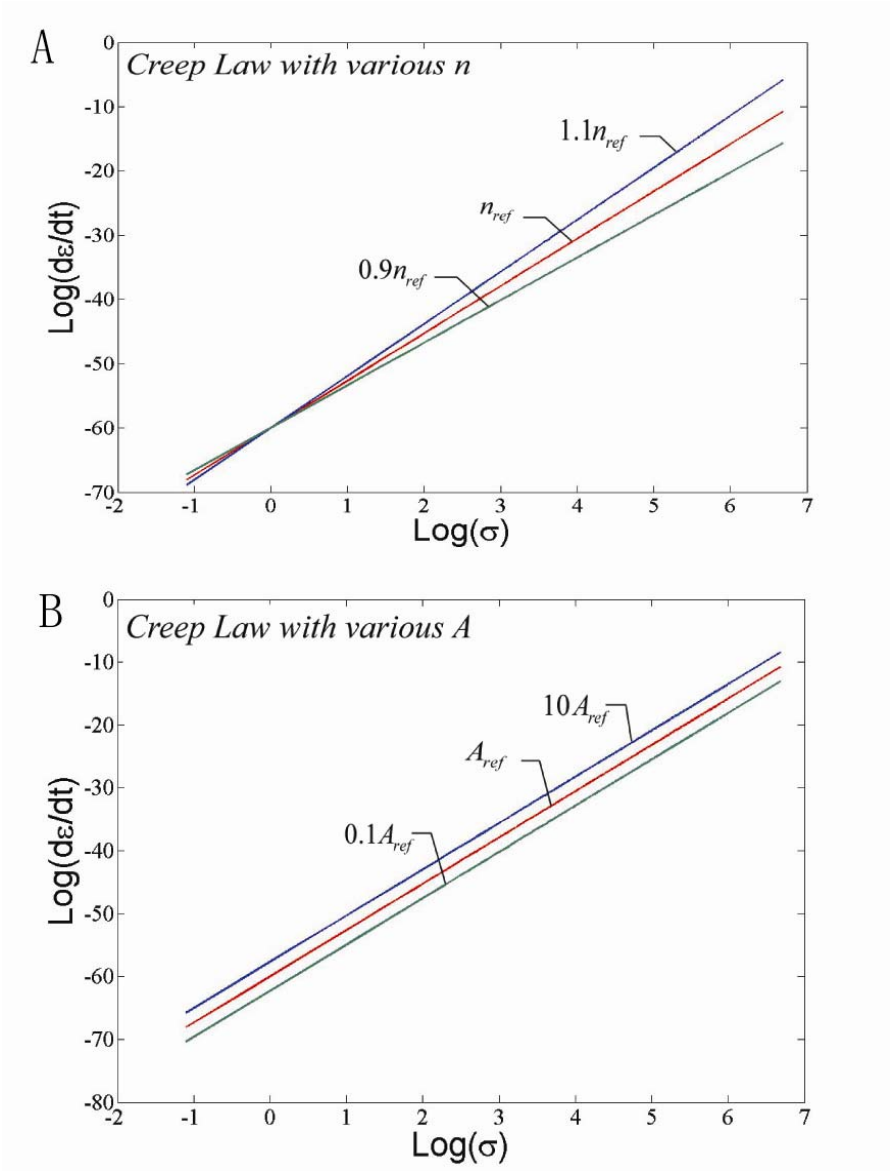


Fig. 3.3 Creep law with variation of (A) n and (B) A.

As can be seen from Fig. 3.3 when the creep law is expressed by equation (3.2), it takes the form of a straight line. The variation of n (i.e. small variation of material properties) in equation (3.2) corresponds to the change of the slope of creep law line. The variation of A corresponds to the translation of creep law line. The sensitivity

simulations are conducted by varying A and n. In the following, the material properties shown in Table 3.1 are denoted A_{ref} and n_{ref}. This is the reference case based on experiments that will be compared to the results of simulations with varied material properties. Fig. 3.3 also shows the range of the variation of the material properties which will be investigated. This range is within the expected variance of material properties of the real experiment specimen.

3.3.1 The Effect of the Material Constant A in the Creep Law

Simulations with different material property A are conducted here. All investigated cases are compared to the reference case $A_{\mathrm{ref}} = 9.0157 \times 10^{-27}$.

The A values of the six cases investigated are listed in Table 3.3.

Case	A	
1	$0.1A_{ref}$	9.0157×10^{-28}
2	$0.5A_{ref}$	4.50785×10^{-27}
3	$0.9A_{ref}$	8.11413×10^{-27}
4	$1.1A_{ref}$	9.91727×10^{-27}
5	$1.5A_{ref}$	13.52355×10^{-27}
6	$10A_{ref}$	9.0157×10^{-26}

Table 3.3 A value of investigated cases in the sensitivity analysis.

Fig. 3.4 shows the comparison results of stress evolution over 90 cycles (hold time 120 seconds) with $0.1A_{ref}$ and A_{ref}. This figure includes the details of every cycle. Since

the plot is hard to read, the results will in the following be presented as the global trends only. In particular the maximum stress for each cycle is plotted. Fig. 3.5(A) shows the evolution of stress for the selected A. Fig. 3.5(B) show the evolution of the creep strain for the selected A.

According to equation (3.2), with the translation of the creep law line (Fig. 3.3(B)), the maximum stress and creep strain will change accordingly. It can be seen from the figure that as A increases, the maximum stress decreases and the creep strain increases.

As can be seen from Fig. 3.5(A), a lower value of A results in a slower stress-relaxation and a larger value of A results in a faster stress-relaxation. However, the rate is not very sensitive to small variations of A. When A changes 10%, the results of the stress changes far less than 10%. Fig. 3.5(B) shows similar trends for the creep strain.

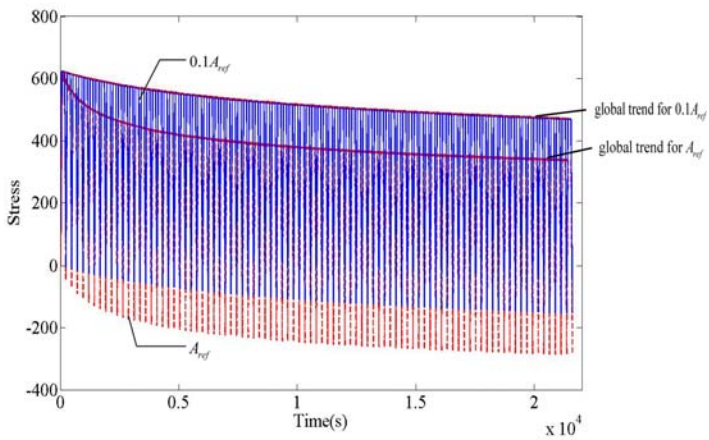


Fig. 3.4 Stress as a function of time for A_{ref} and $0.1A_{ref}$ showing each computational cycle.

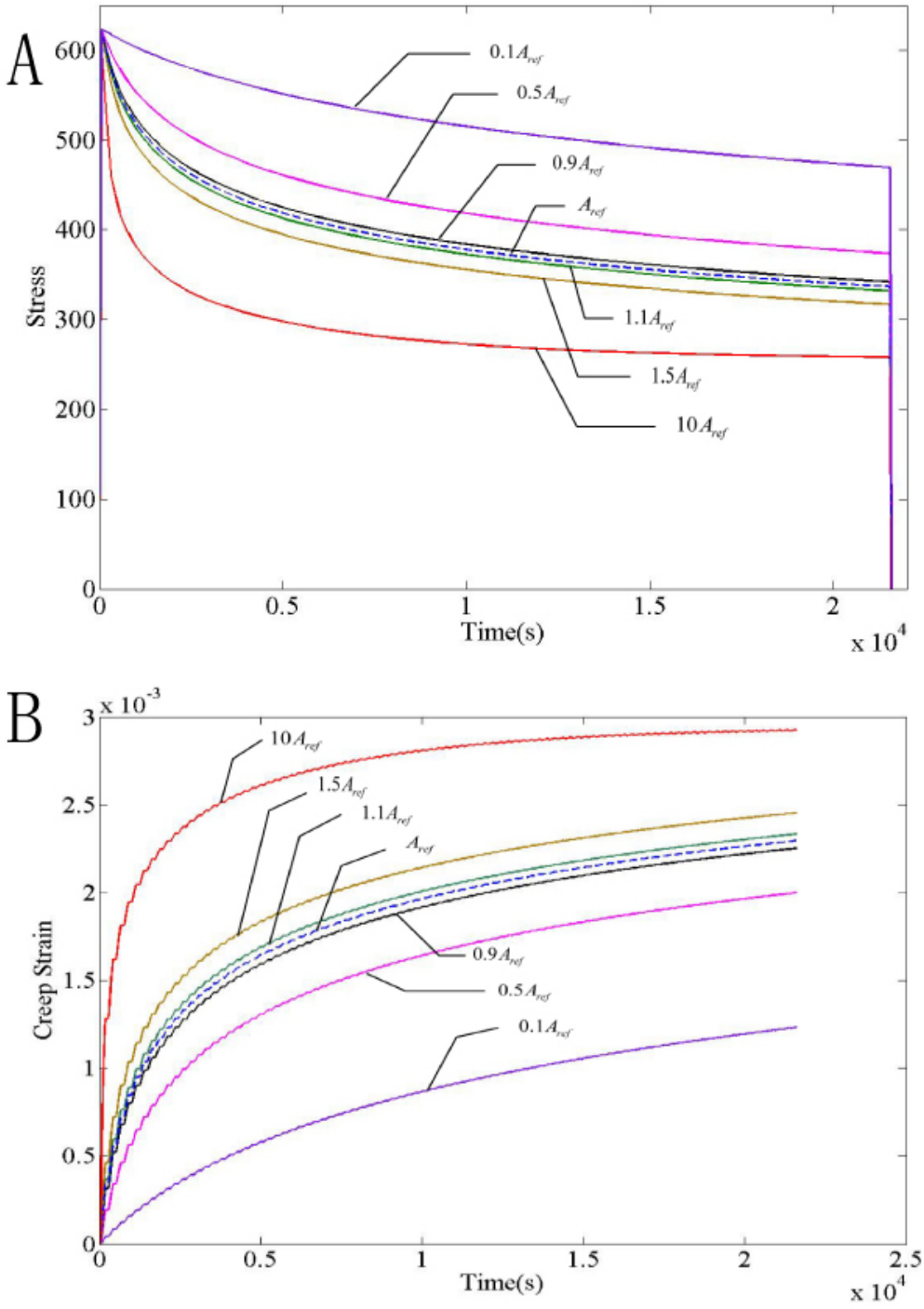


Fig. 3.5 (A) Stress and (B) Creep strain as a function of time for selected A (Holding time=120s, 90 cycles)

3.3.2 The Effect of the Material Constant n in the Creep Law

Simulations with different material property n are conducted as well. All results of the investigated cases are compared with the reference case n_{ref}.

The n values of the four cases investigated are listed in Table 3.4.

Case	n	
1	$0.9n_{ref}$	6.62463
2	$0.95n_{ref}$	6.992665
3	$1.05n_{ref}$	7.728735
4	$1.1n_{ref}$	8.09677

Table 3.4 n value of investigated cases in the sensitivity analysis.

According to equation (3.2), with the change of slope of the creep law line (Fig. 3.3(A)), the maximum stress and creep strain will change with the same trend as translation of the creep law line. It can be seen from the figure that as n increases, the maximum stress decreases and the creep strain increases.

As seen in Fig. 3.6 the results are more sensitive to the variation of n than the previously discussed changes in material parameter A. When n changes 10%, the results of the stress changes about 50%. The creep strains changes even more. It is clear that the variation of n has a significant influence on the results.

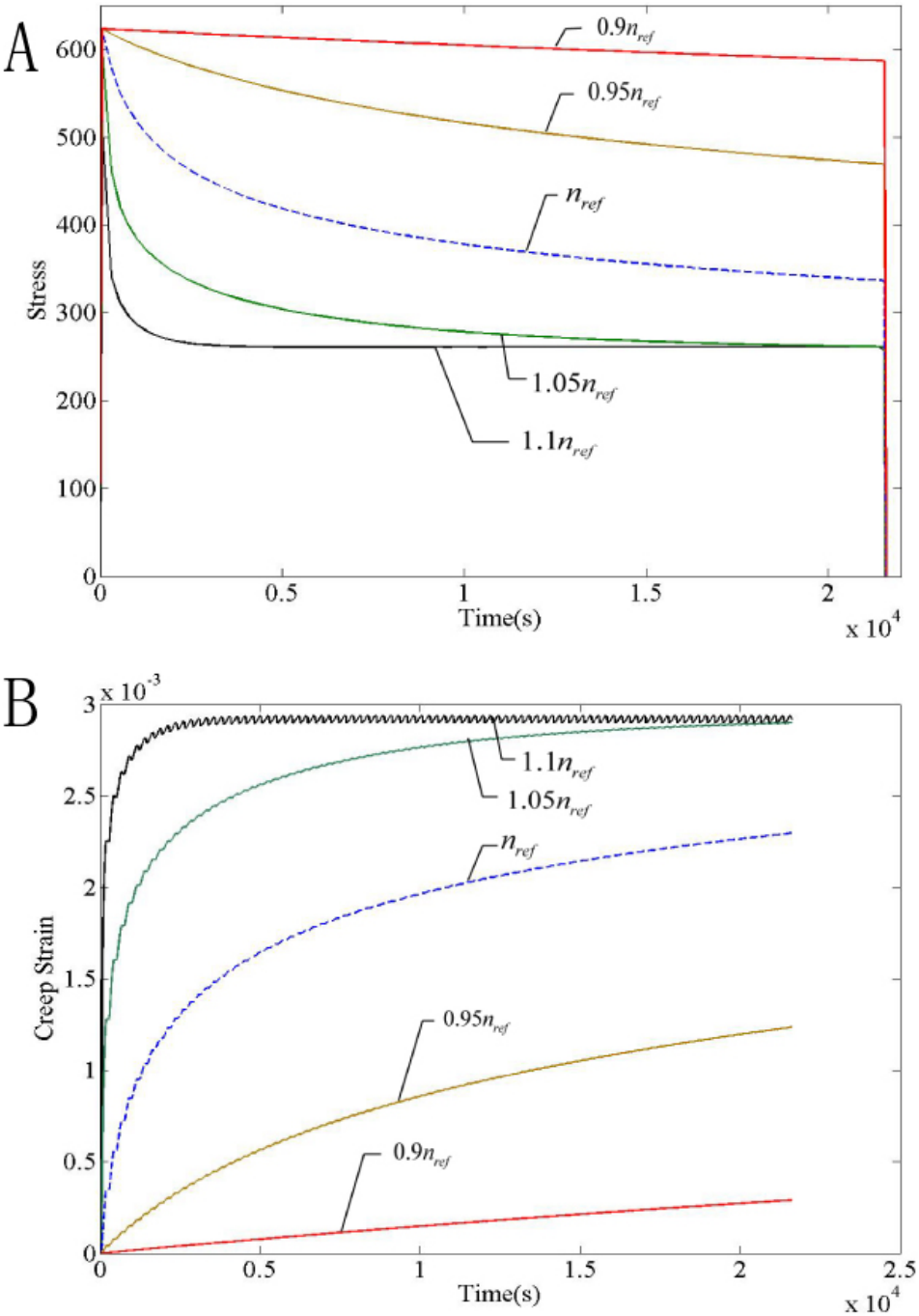


Fig. 3.6 (A) Stress and (B) Creep strain as a function of time for selected n (Holding time=120s, 90 cycles)

3.3.3 Comparison between Cyclic Loading and Hold

As a final numerical investigation before the experiments, we compare a constant load with the cyclic load conditions. How does cyclic loading affect the results? The constant load is hold for 3 hours' experimental time. For the structure under cyclic loading, the holding time for every cycle is 120 seconds and there are 90 cycles of loading so that the total holding time is 3 hours during the simulation.

Fig. 3.7 shows the comparison of the two groups with several A values. In all cases, the results of stress for constant load decrease faster than for cyclic loading. The results of creep strain for constant load increase faster than for cyclic loading.

Fig. 3.8 shows the comparison of the two groups with several n values. For different n values, the results of stress for constant load decrease faster than for cyclic loading. The results of creep strain for constant load increase faster than for cyclic loading.

Thus, the evolution of state variables of structures under cyclic loading is not the same as structures without cyclic loading. The difference may depend on the holding time (i.e. number of cycles) and needs to be investigated further. However, these simulations show that cyclic loading will not produce the same results as constant load.

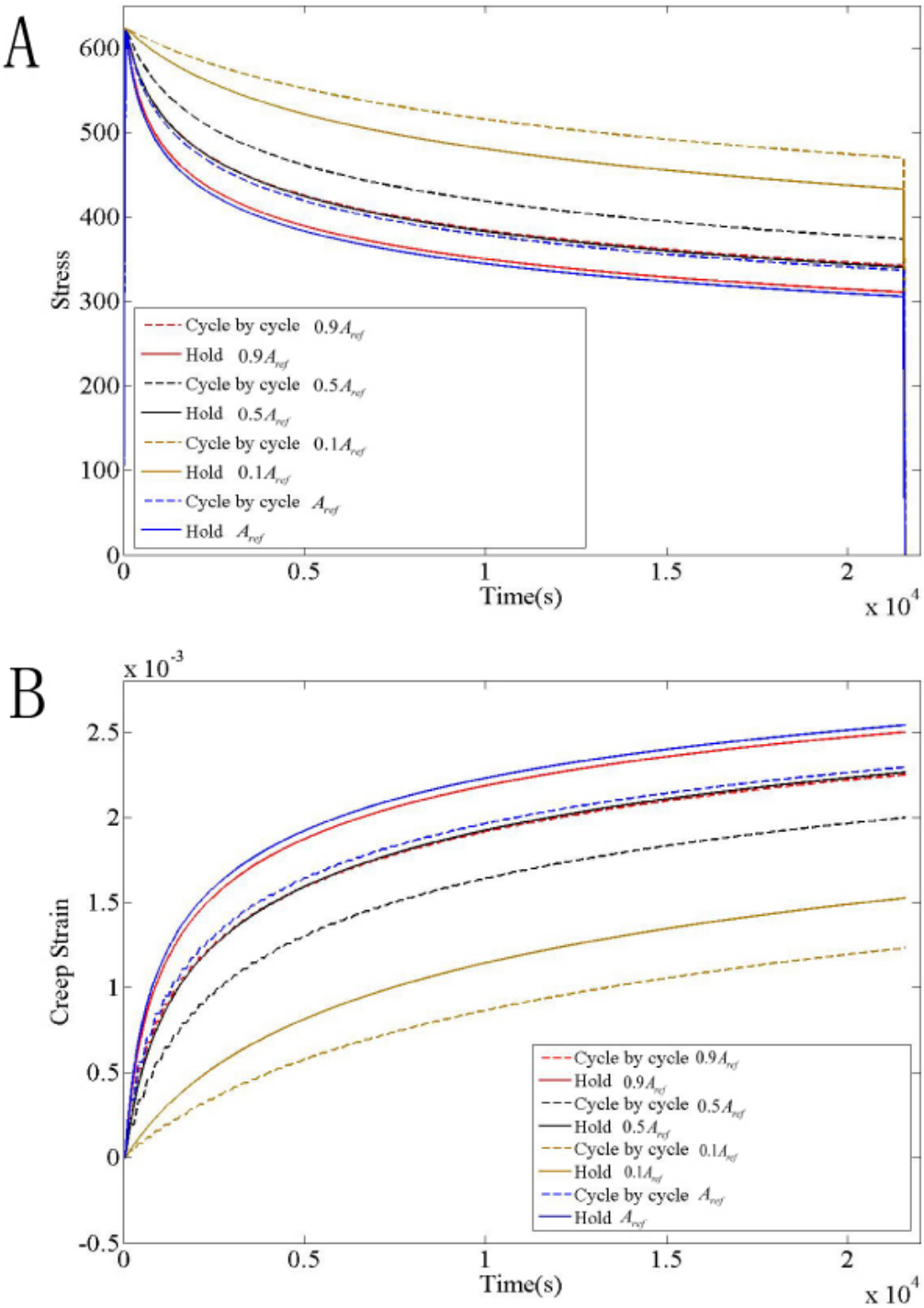


Fig. 3.7 (A) Stress and (B) Creep strain as a function of time with selected A

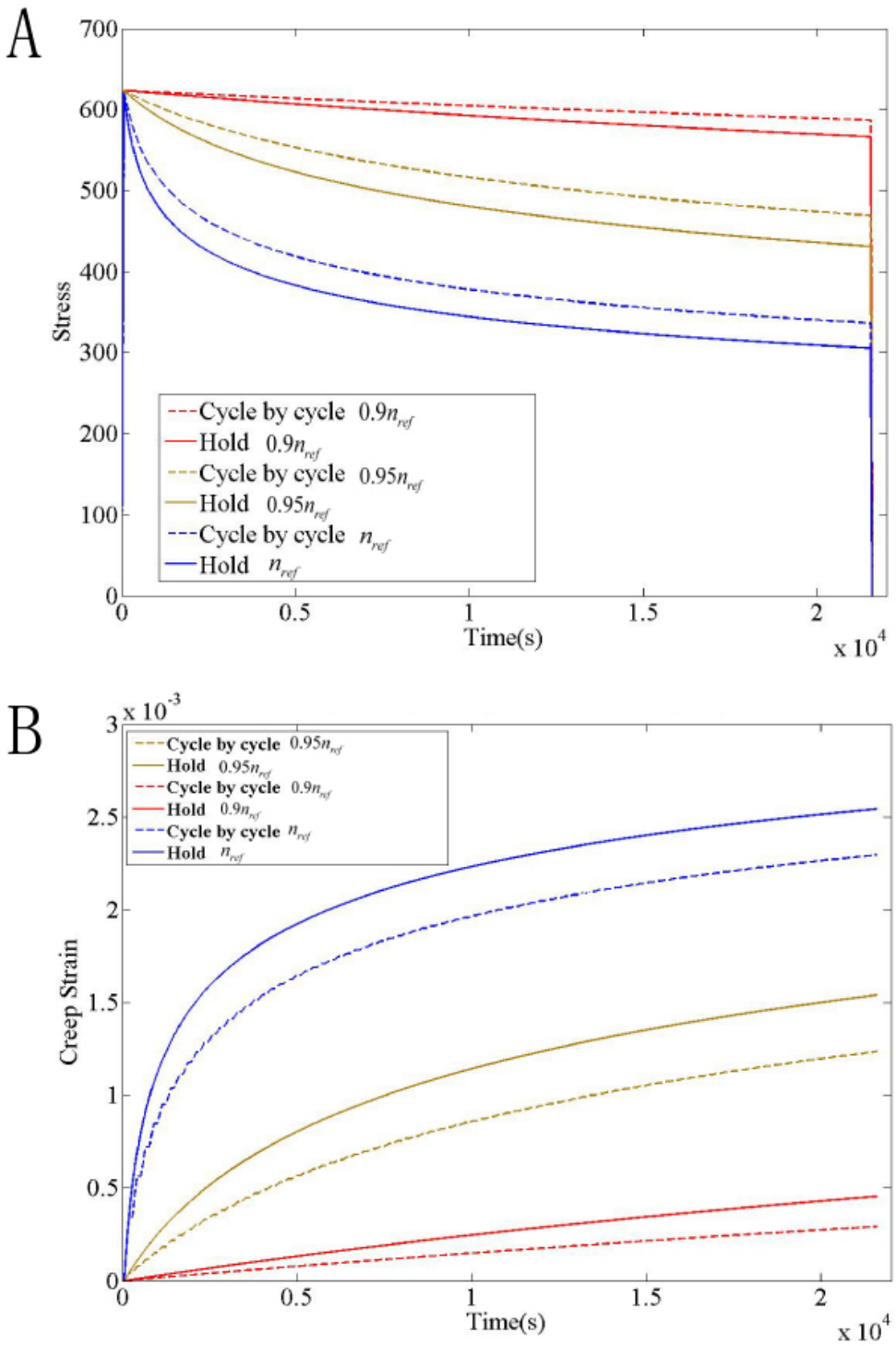


Fig. 3.8 (A) Stress and (B) Creep strain as a function of time with selected n

3.4 Comparison of Numerical and Experimental Results

Results from preliminary experimental investigations will be presented here. These investigations are ongoing and only a limited set of results is available.

Two sets of experimental results are obtained and used to compare to the results from the cycle-jump technique. The load sequence is shown in Table 3.5. The initial experiments used a lower displacement than recommended, to avoid premature failure.

Test	Number of Cycles	Loading/Unloading time (s)	Holding time (s)	Applied displacement (mm)
1	180	10	60	0.0084
2	90	10	120	0.0168

Table 3.5 Test Conditions

As pointed out previously, there are only limited experimental data available, both with respect to creep-test, which establish the necessary mechanical properties, and the fatigue simulations, which are intended to be used for verifying the cycle jump technique. These limited data provides significant limitation in actually proving that the cycle jump technique will predict true material behavior, but we will below discuss how we deal with this limited data set, and show that these test together can provide information about the material data.

Only four creep tests were conducted at DLR to evaluate the material properties, which is, four groups of data with different steady-state stress and the corresponding creep strain rate were obtained. The steady-state stress and the corresponding creep strain rate are plotted on log-log scale in Fig. 3.9. A linear curve fit of these experimental data

is shown with straight line in the plot. This is “the creep law line” which describes the constitutive equations of the material. The limited number of creep-tests conducted results in a high uncertainty of the material properties. The material properties obtained using the four points are summarized in Table 3.1.

However, after comparing the results obtained in the cycle-jump technique with the results of cyclic experiments, it is found that if the data point of the highest stress (500MPa in the figure) is not used to fit the creep law line, an improved fit is obtained. The creep law line obtained by three experimental data points is also shown in Fig. 3.9. The material properties used in the numerical simulation to be compared with the experimental results are shown in Table 3.6 and in Table 3.6, material properties obtained using the four points are also shown for comparison.

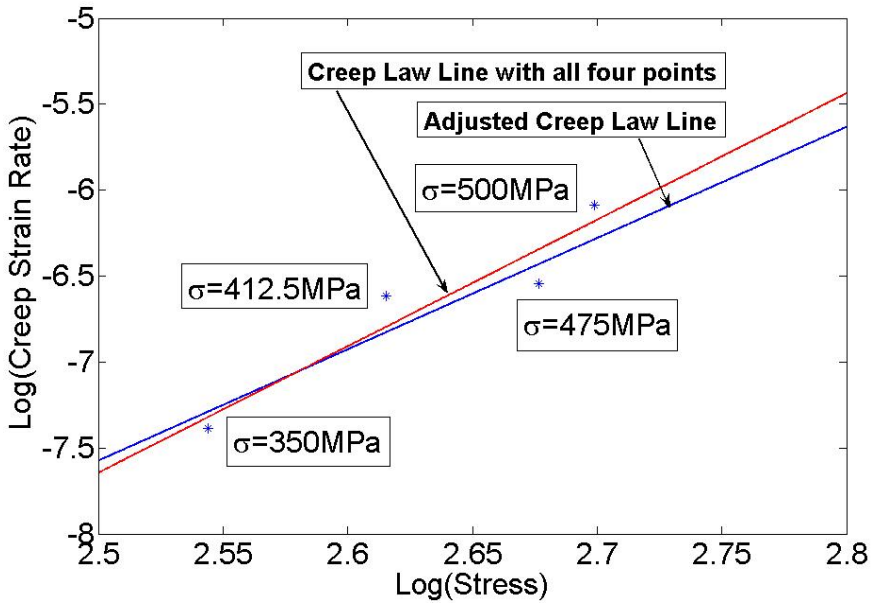


Fig. 3.9 Creep Law Lines, using all four points and omitting the point corresponding to 500 MPa

	three points	four points
Elastic Modulus (E)[MPa]	$125,000$	$125,000$
Creep Exponent (n)	6.474	7.3607
Creep pre-factor $(A)[MPa^{-n}s^{-1}]$	1.74×10^{-24}	9.0157×10^{-27}

Table 3.6 Material properties evaluated from creep tests using all four points and omitting the point corresponding to 500 MPa (three points)

The numerical simulation results are compared with the experimental results and these results are shown in Fig. 3.10 and Fig. 3.11. In these two figures, the material properties used in the numerical simulations are summarized in Table 3.7 where E_{ref} and n_{ref} are material properties evaluated by ignoring the point for 500MPa which is summarized in Table 3.6. A small variation of the material properties has been adopted in the numerical simulations. This is valid because a small variation of material properties occurs naturally for real samples. In test 2, the experimental specimen fails after about 6000 seconds. It can be seen from the figure that the experimental results agree with the numerical results.

Elastic Modulus (E)[MPa]	$0.95E_{ref}$
Creep Exponent (n)	$1.08n_{ref}$
Creep pre-factor $(A)[MPa^{-n}s^{-1}]$	1.74×10^{-24}

Table 3.7 Material properties used in the numerical simulation based on the material properties evaluated by three data points.

It can be concluded that results above using the cycle-jump technique show a very promising indication of correlating to the experimental results. However, additional experiments need to be conducted for a satisfying verification.

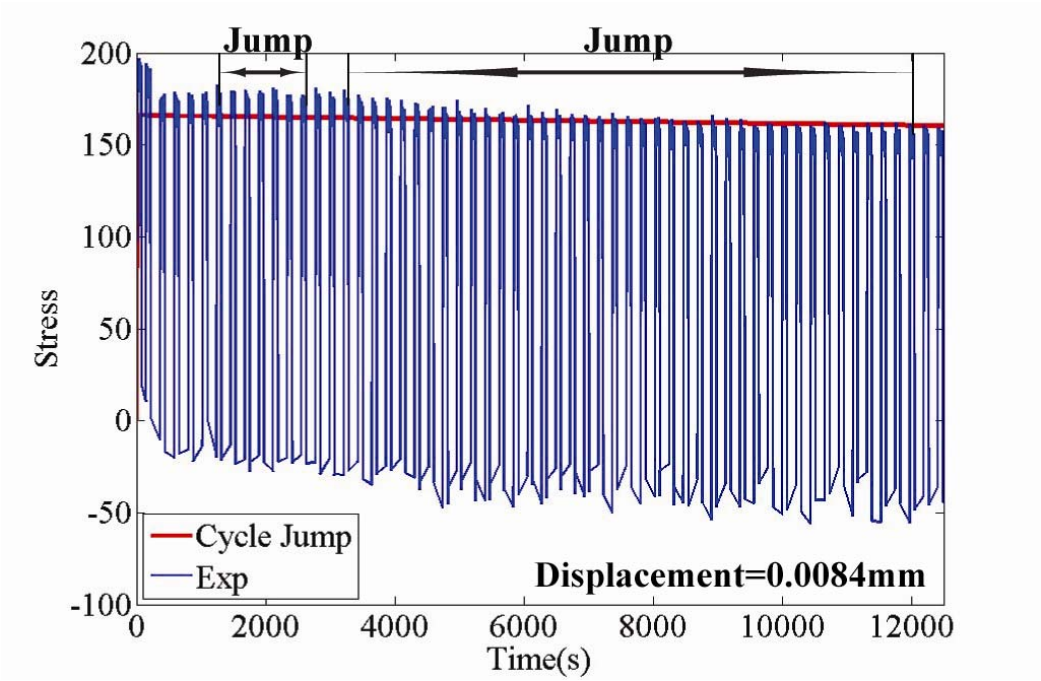


Fig. 3.10 Stress as a function of time (holding time=60s, 180 cycles, material properties used according to Table 3.7)
(Note: only every second experimental cycle is included for clarity. Since the plot is hard to read with details of every simulation cycle, only the global trend of the state variable is plotted here and the jump length is marked in the figure.)

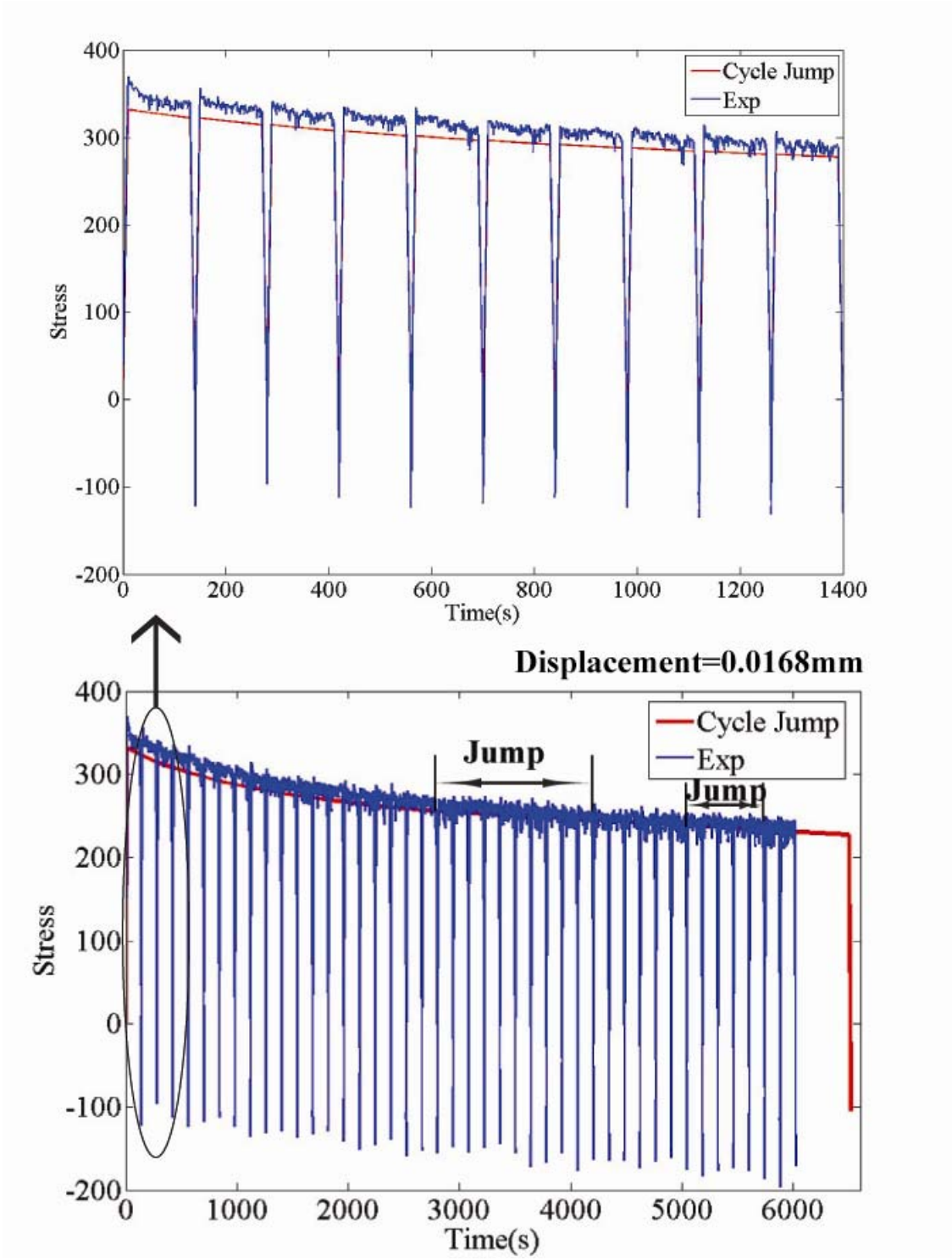


Fig. 3.11 Stress as a function of time (Holding time=120s, 90 cycles, material properties used according to Table 3.7)

According to the experimental and numerical results obtained here, more experiments can be conducted to verify the cycle-jump technique. The suggested test conditions are shown in Table 3.8.

The maximum displacement used in test 2 in Table 3.5 could be decreased to avoid failure in the experiment. With smaller maximum displacement, linear global evolution is expected. Furthermore, the experimental time could be increased and this will lead to a more obvious linear global evolution of the state variables.

Test	Number of Cycles	Loading/Unloading time (s)	Holding time (s)	Applied displacement (mm)
1	360	10	60	0.0084
2	90	10	120	0.0138
3	180	10	120	0.0138
4	360	10	120	0.0138
5	180	10	120	0.0168

Table 3.8 Suggested Test Conditions

Chapter 4

CONCLUDING REMARKS

Fatigue and unwanted failures of structures can be caused by repeated cyclic loading. Understanding this type of failure is important to obtain reliable engineering structures. This thesis discussed the investigation of structures subjected to cyclic loading using numerical tools. In particularly, this work focused on enhancing a previously developed numerical tool, which enables accelerated numerical simulations.

Since thermal barrier coatings (TBCs) have inspired this work, the TBC system is introduced. Even though some work on accelerated numerical simulations exists, the existing work found in the literature has significant limitations. The numerical tool used here, the "cycle-jump technique", is a general method to improve the computational efficiency that was proposed by Cojocaru and Karlsson [5]. This method is a promising technique, but needs to be augmented and verified experimentally to become a reliable tool.

In order to investigate the TBC system, creep was incorporated in the numerical scheme for the cycle-jump technique. The scheme is also extended to handle force controlled loading as well as displacement controlled loading. The improved code is verified numerically by comparing the results from cycle-jump technique to cycle-by-cycle simulation and excellent agreement is obtained.

It is unfortunately not sufficient to verify the cycle-jump technique with numerical simulations only, but experimental investigations and comparisons are needed as well. Thus, a procedure of designing experimental load sequences was discussed in this work. After developing a numerical model based on geometry and material properties of the experimental specimen, numerical simulations were conducted and several groups of appropriate load sequence data were suggested to be used in the experiments to verify the cycle-jump technique.

A sensitivity analysis was conducted due to the ever-present expected variation of material properties in real samples. The sensitivity analysis shows that the expected variance of material properties has an influence on material response of the investigated structures. Simulation results also show that there is a difference between material response of a structure under constant load and cyclic loading, which shows the importance of investigation of structures under cyclic loading.

Numerical simulations are conducted using the material properties evaluated from the creep tests. After obtaining the results of the experiment, the numerical results are compared with the experimental results. These two groups of results agree with each other which show a promising expectation that the cycle-jump technique can successfully predict the evolution of the state variables of structures under cyclic loading.

Thus, in conclusion, this work suggests that the previously proposed cycle-jump technique can work as a tool to predict the evolution of structures subjected to cyclic loading. However, additional experimental investigations are needed to verify this.

REFERENCE

[1] T.E. Strangman, Thermal Barrier Coatings for Turbine Airfoils, Thin Solid Films (1985), pp. 93–105.

[2] ABAQUS, ABAQUS 6.7, ABAQUS Inc., Pawtucket, Rhode Island, 2006

[3] ABAQUS 6.7 Scripting User's Manual. 2006, Providence, RI: ABAQUS, Inc.

[4] ABAQUS 6.7 Scripting Reference Manual. 2006, Providence, RI: ABAQUS, Inc.

[5] D. Cojocaru and A.M. Karlsson, A simple numerical method of cycle jumps for cyclically loaded structures, Int J Fatigue (2006), pp. 1677–1689.

[6] P. Boisse, P. Bussy and P. Ladeveze, A new approach in nonlinear mechanics – the large time increment method, Int J Numer Meth Eng (1990), pp. 647–663.

[7] J.Y. Cognard, P. Ladeveze and P. Talbot, A large time increment approach for thermo-mechanical problems, Adv Eng Software (1999), pp. 583–593.

[8] J. Fish and Q. Yu, Computational mechanics of fatigue and life predictions for composite materials and structures, Comput Meth Appl Mech Eng (2002), pp. 4827–4849.

[9] C. Oskay and J. Fish, Fatigue life prediction using 2-scale temporal asymptotic homogenization, Int J Numer Meth Eng (2004), pp. 329–359.

[10] H. Kiewel, J. Aktaa and D. Munz, Application of an extrapolation method in thermocyclic failure analysis, Comput Meth Appl Mech Eng (2000), pp. 55–71.

[11] F. Bogard, P. Lestriez, and Y.Q.Guo, Numerical modeling of fatigue damage and fissure propagation under cyclic loadings, International Journal of Damage Mechanics (2008), p 173-187

[12] Saanouni, K., Nesnas, K. and Hammi, Y., Damage Modeling in Metal Forming Process, Int. J. Damage Mech.(2000), 9(3): 196-240.

[13] Lemaitre, J. and Chaboche, J. L., Mechanics of Solid Materials (Trans. B. Shrivastava)(1990), Cambridge University Press, Cambridge, UK.

[14] A.M. Karlsson, J.W. Hutchinson and A.G. Evans, A fundamental model of cyclic instabilities in thermal barrier systems, J Mech Phys Solids (2002), pp. 1565–1589.

[15] A.M. Karlsson, J.W. Hutchinson and A.G. Evans, The displacement of the thermally grown oxide in thermal barrier systems upon temperature cycling, Mater Sci Eng A (2003), pp. 244–257.

[16] A.M. Karlsson, Evans AG. A numerical model for the cyclic instability of thermally grown oxides in thermal barrier systems. Acta Mater 2001; 49: 1793-804.

[17] A.M. Karlsson, Levi CG, Evans AG. A model study of displacement instabilities during cyclic oxidation. Acta Mater 2002; 50: 1263-73.

[18] A.M. Karlsson, Xu T, Evans AG. The effect of the thermal barrier coating on the displacement instability in thermal barrier systems. Acta Mater 2002; 50: 1211-8.

[19] M. Bartsch, B. Baufeld, M. Heinzelmann, A. M. Karlsson, S. Dalkilic, L. Chernova, Multiaxial thermo-mechanical fatigue on material systems for gas turbines, Materialwissenschaft und Werkstofftechnik (2007), Volume 38 Issue 9, p. 712 – 719.

[20] Stanley B. Lippman, Josée Lajoie, C++ primer, Reading, Mass.: Addison-Wesley, c1998.

[21] Betten, Josef, Creep mechanics, Berlin ; New York : Springer, c2002.

[22] Evans HE, Strawbridge A, Carolan R.A., Ponton C.B.. Creep effects on the spallation of an alumina layer from a NiCrAlY coating. Materials Science and Engineering A-Structural Materials Properties Microstructure and Processing 1997;225:1.

[23] Stephen J. Chapman, Introduction to Fortran 90/95 , Boston : WCB/McGraw-Hill, c1998.

APPENDIX

In this section, the function of the PYTHON and FORTRAN code are described. There are six PYTHON files and one FORTRAN file making up the code for the cycle-jump technique. The six python files are: main_script_cycle_jump.py, JumpAnalysis_v2_v6_7.py, utilities.py, importsFile.py, mathFunctions.py and ivStorage_v6_7.py. These six files are used to control the whole procedure of the cycle jump technique: set up the model, write the data to the output file, calculate the appropriate jump length, calculate the initial condition for the simulation after the cycle jump, write the data to the input file, and update the model. The FORTRAN file is named as DISP.for. This FORTRAN file is used to impose the initial condition of displacement. If the loading is stress controlled, we need to call the FORTRAN code in the simulation. If the loading is displacement controlled, we do not need to use the FORTRAN file. Instead, we will run the intermediate initiation simulation.

main_script_cycle_jump.py

This file contains the main script code for conducting cycle-jump analysis. It controls the whole procedure of the cycle jump technique by calling all other PYTHON files.

The user can set the target time of the simulation, the control parameter and the first extrapolation time here.

JumpAnalysis_v2_v6_7.py

This file has two main parts. In the first part, the finite element analysis model is created. In the second part, the length of the cycle jump is calculated based on the data obtained from the previous results of the simulation. The following are main functions in this two parts.

class CJumpAnalysis:

def __init__

Define initial cycles and names of every new job.

def deleteStorage

Delete the storage variables to free memory.

def RunAdaptiveLinearME

Run the analysis in an adaptive way using linear extrapolation. Based on the absolute error of two extrapolated values. In this function, the length of the cycle jump is calculated based on the results of previous simulation.

def GenerateModel

This function generates a parametric model associated with the current jump analysis, within the Abaqus. This part set up the finite element model that we want to simulate. The model can be updated after the cycle jump.

def creepaccumulation

This function calculates the time of the intermediate initiation simulation.

def onMessage

Job message callback

utilities.py

This file contains some basic functions which are used to interact with the input file and the CAE.

def getKeywordBlockPosition

Find the exact location where we want to insert keyword and then insert keyword into the input file.

def getOdbObject

Returns an Odb Object.

def openModel(name):

Open a model in CAE.

importsFile.py

This file imports all modules in CAE used in the code.

mathFunctions.py

This file contain all mathematic functions used in the code.

def getRelError

Evaluate the relative error between two values, using one of them (third parameter) as reference.

def Spline

Computes spline of data=[[x0,y0][x1,y1],...]

def approxByLine(next_time,d12,d23)

Compute the slope at next_time using two known slope d12,d23

ivStorage_v6_7.py

This file is used for reading the data from the output file and writing the data into the input file. It will interact with the RunAdaptiveLinearME function to calculate the length of the cycle jump.

class C_IVarStorage:

(This part is used to read or write stress and strain from or to a file.)

def splineData

This compute 2nd derivatives used for spline interpolation

def slopeData

This function compute the discrete slopes of the stored values

def computeLsfBasedAt(self,time):

This function is meant to compute and add the value at input time based on previously computed lsf functions at each leaf entry.

def computeLineBasedAt(self,time,index1=-2,index2=-1,time1=None,time2=None):

This function is meant to compute and add the value at input time based on line equation through values at index1(time1) and index2(time2) at each leaf entry.

def computeSplineBasedAt(self,time,index1=-2,index2=-1):

This function is meant to compute and add the value at input time based on spline approximation previously computed by splineData() at each leaf entry.

def computeToNewComponent

This function computes the value of a new component newCompName, based on given function and variable list varList

def createXYData

Create XYData for plotting inside Abaqus

def checkKeys

Checks the existance of a given set of keys.

def exportElemDataToCSV

Export the information from object associated with instance/element given.

def extractData

Extract Data from Odb file and fill the dictionary of the object. These data are used to calculate the length of the cycle jump.

def getClosedTimeIndex

This function return the index of the pair [time_stored,value_stored] for which the time_stored is most closed to input time.

def removeValuesAt

This function remove all stored values at given index, or time.

def writeToFileValuesAt

Create the list of stored field components which will further be output to outputFile. These data will be used as initial condition for a new finite element analysis after the cycle jump.

def LsfFieldAtTime

LeastSquareFit of outputComp(inputComponents) as a combination of functions in functionsList

class C_NodalStorage:

(This part is used to read or write displacement from or to a file.)

def extractData

Extract nodal Data from Odb file and fill the dictionary of the object. These data are used to calculate the length of the cycle jump.

def writeToFileValuesAt

Create the list of stored field components which will further be output to outputFile. These data will be used as initial condition for a new finite element analysis after the cycle jump.

DISP.for

Impose displacement as the initial condition into the finite element analysis if force is the controlled loading. In this file, the displacement of every node is different. The initial condition will be imposed node by node.