The development of finite element analysis software for creep damage analysis

D. Liu¹, Q. Xu¹, Z. Lu², D. Xu¹, F. Tan¹

¹School of Science and Engineering, Teesside University, Middlesbrough, TS1 3BA, UK
²School of Computing and Engineering, University of Huddersfield, Huddersfield, HD1 3DA, UK

Abstract - The development and application of computational creep damage is very active among research community and high temperature industry. This paper presents a development and preliminary validation of in-house finite element analysis software for creep damage analysis. The Fortran 90 and existing finite element library were adopted and used in the software development. The validation case study was conducted and reported, using uni-axial creep case.

Keywords: Computational Creep Damage Mechanics; FE Algorithm; Non-linear Material; Validation

1 Introduction

This paper reports a development and preliminary validation of in-house finite element software for creep damage analysis. Creep damage mechanics has been developed and applied for the analysis of creep deformation and the simulation for the creep damage evolution and rupture of high temperature components [1]. The computational capability relies on the availability of a computational tool and a set of creep damage constitutive equations which can accurately depicts the complex phenomena. This paper addresses the former.

The need of such computational capability and the justification for developing in-house software was conducted and reported in the early stage of this research [2, 3]. Essentially, the creep damage problem is of time dependent, non-linear material behavior, and multi-material zones. The standard software does not provide the computational capability to simulate the tertiary creep stage readily without the development of complicated material user subroutine. The computational capability can only be obtained by either the development and use of special subroutine in junction with standard commercial software such as ABAQUS and ANSYS or the development and use of dedicated in-house software, each has its own advantages and disadvantages [2]. Hyde [4] and Hayhurst [5] have reported the development and the use of their in-house software for creep damage analysis; furthermore, Ling has presented a detailed discussion and use of Runge-Kutta type integration algorithm [6]. On the other hand, it is noted that Xu revealed the deficiency of KRH formulation and proposed a new formulation for the multi-axial generalization in the development of creep damage constitutive equations [7]. The new creep damage constitutive equations for low Cr-Mo steel and for high Cr-Mo steel are under developing by colleagues in this research group [8, 9].

The purpose of this paper is to present the finite element method based on CDM to design FE software for creep damage mechanics. More specifically, it reports the structure of the new FE software and the existing FE library applied in obtaining such computational tool via an approach for stress and field variable updating, and preliminary validation of current version of such software via a uni-axial tension model. The contribution of this paper is to provide a new version of in-house software to solve the whole process of all the three stages of creep deformation and damage problem.

2 The finite element method based on CDM

2.1 The Continuum Damage Mechanics (CDM)

The continuum damage mechanics (CDM) used the concept of creep damage as internal variable [10, 11, 12]. One of the features of CDM is phenomenological and the damage parameter can in this context represent creep damage. In creep damage mechanics, the material gets damaged does not essential has to be understood in detail and the damage should be considered in analysis [13].

The creep deformation is typically divided into primary, secondary and tertiary stages. Initially, the characteristics of the primary-secondary (steady state) creep deformation behavior are observed by simple experiments, but later the mathematical description of tertiary creep through the use of CDM.

The finite element technique combined with continuum damage mechanics has been testified to be an efficient tool for assessing the performance of the structural components [14-16]. In many commercial finite element packages, the user-routines can be written for implemented taking into account one or more damage mechanisms [17].

Rupture processes can also be investigated by use of CDM approach [18]. Some investigations of creep crack initiation and growth in notched specimens have been performed with promising results [19, 20].

2.2 The general structure of the finite element software

The structure of developing in-house finite element software for creep damage analysis is listed in Figure 1.
The steps for the development of finite element software can be summarized in:
1. Input the definition of a specific FE model including nodes, element, material property, boundary condition, as well as the computational control parameters
2. Calculate the initial elastic stress and strain
3. Integrate the constitutive equation and update the field variables such as creep strain, damage, stress; the time step is controlled
4. Remove the failed element [21] and update the stiffness matrix
5. Stop execution and output results

2.3 The integration of creep damage constitutive equation
The FEA solution critically depends on the selection of the size of time steps associated with an appropriate integration method. Some integration method has been reviewed in previous work [3]. In the current version, Euler forward integration subroutine, developed by colleague [22], was adopted here for simplicity. More sophisticated Runge-Kutta type integration scheme will be adopted and explored in future.

2.4 The finite element algorithm for updating stress
The Absolute Method [23] has been given for the solution of the structural creep damage problems. The principle of virtual work applied to the boundary value problem is given:

\[ P_{\text{load}} = [K_v] \cdot \text{TOTD} - P_c \quad (1) \]

Where \( P_{\text{load}} \) is applied force vector, and \([K_v]\) is the global stiffness matrix, which is assembled by the element stiffness matrices \([K_m]\); \text{TOTD} is the global vector of the nodal displacements and \( P_c \) is the global creep force vector.

\[ [K_m] = \iint [B]^T [D] [B] \, d\alpha \, dy \quad (2) \]

The \([B]\) and \([D]\) represent the strain-displacement and the stress-strain matrices respectively.

\[ \text{TOTD} = [K_v]^{-1} \cdot (P_{\text{load}} + P_c) \quad (3) \]

The initial \( P_c \) is zero and the Choleski Method [14] is used for the inverse of the global stiffness matrix \([K_v]\). By giving the \( P_{\text{load}} \), the elastic strain \( \varepsilon_{ek} \) and the elastic stress \( \sigma_{ek} \) for each element can be obtained:

\[ \varepsilon_{ek} = [B] \cdot \text{ELD} \quad (4) \]
\[ \sigma_{ek} = [D] \cdot \varepsilon_{ek} \quad (5) \]

The element node displacement \( \text{ELD} \) can be found from the global displacement vector and the creep strain rate \( \varepsilon_{cKrate} \) for each element can be obtained by substituting the element elastic stress into the creep damage constitutive equations. The creep strain can be calculated:

\[ \varepsilon_{ck} (t + \Delta t) = \varepsilon_{cK}(t) + \varepsilon_{cKrate} \cdot \Delta t \quad (6) \]

The node creep force vectors for each element are given by:

\[ P_{ek} = [B]^T [D] \cdot \varepsilon_{ek} \quad (7) \]

The node creep force vector \( P_{ek} \) can be assembled into the global creep force vector \( P_c \) and the \( P_c \) is used to update the equation (1). Thus, the elastic strain can be updated:

\[ \varepsilon_{totk} = [B] \cdot \text{ELD} = \varepsilon_{ek} + \varepsilon_{ck} \quad (8) \]
\[ \varepsilon_{ek} = [B] \cdot \text{ELD} - \varepsilon_{ck} \quad (9) \]

Where the \( \varepsilon_{totk} \) and \( \varepsilon_{ek} \) represent the total strain and creep strain for each element respectively; and the elastic strain \( \varepsilon_{ek} \) is used to up-date the equation (5).

3 The application of existing finite element library
In the development of this software, the existing FE library and subroutines such as [23] was used in programming. The subroutines can perform the tasks of finite element meshing, computing and integrating the element matrices, assembling element matrices into system matrices and carrying out appropriate equilibrium, eigenvalue or propagation calculations.

Fig. 1. The flow diagram of the new finite element software
3.1 The finite element meshing
The existing subroutines [23] are available in performing the mesh of element for the triangles, quadrilaterals and hexahedra “bricks”. For example, the subroutine geometry_3tx (iel, itx, a, b, coord, num) can form the coordinates and node vector for a rectangular mesh of uniform 3-node triangles. More specifically, it is counting in the x-direction and local numbering clockwise. For the quadrilateral elements and the hexahedra “bricks” elements, the subroutine geometry_4qx and the subroutine geometry_8bxz are also available in [23].

3.2 The element stiffness matrix assembly
The special purpose subroutines such as subroutine formnf, subroutine formkb, subroutine formku, and subroutine fsparv can assemble the individual element matrices to form the global matrices. The selection of element stiffness matrix subroutine is according with the definition of the geometrical details, especially in the nodal coordinates of each element and the element’s place in overall node numbering scheme. More details see [23].

3.3 The solution of equilibrium equation for creep problem
Direct solution methods and iterative solution methods have been used in solving the creep problem [13]. In direct solution method, the subroutine sparin and subroutine spabac based on Cholesky direct solution method are used to solve the sets of linear algebraic equations [23]. In iterative solution method, the subroutine cholin and subroutine chobac based on Jacobi iterative solution methods can be used in programming.

4 Validation of finite element software and FE model

4.1 The creep constitutive equation
Creep damage constitutive equations are proposed to depict the behaviors of material during creep damage (deformation and rupture) process, especially for predicting the lifetime of materials, within the CDM-based numerical computational tool. Kachanov-Rabatnov-Hayhurst (KRH) constitutive equations [24] are introduced as follows in details and used in current program.

- Uni-axial form

\[
\dot{\epsilon} = A \sinh \left( \frac{B(1-H)}{1-\varphi(1-\omega)} \right) 
\]

(10.1)

\[
\dot{\gamma} = \frac{h}{\sigma} (1 - \frac{H}{H^*}) \dot{\epsilon} 
\]

(10.2)

\[
\phi = \frac{Kc}{3} (1 - \varphi)^4 
\]

(10.3)

\[
\dot{\varphi} = C \dot{\epsilon}^\gamma 
\]

(10.4)

Where A, B, C, h, H*, and Kc are material parameters. H (0 < H < H*) indicates strain hardening during primary creep, \(\varphi\) (0 < \varphi < 1) describes the evolution of spacing of the carbide precipitates [24].

- Multi-axial form

\[
\begin{aligned}
\dot{\epsilon}_{ij} &= \frac{3 \sigma_{ij}}{2 \epsilon_e} \text{Asinh} \left( \frac{B \sigma_{ij} (1-H)}{(1-\varphi)(1-\omega)} \right) \\
\dot{H} &= \frac{h}{\sigma} (1 - \frac{H}{H^*}) \dot{\epsilon} \\
\dot{\varphi} &= \frac{Kc}{3} (1 - \varphi)^4 \\
\dot{\omega} &= C \dot{\epsilon}^\gamma
\end{aligned}
\]

(11)

Where \(\sigma_e\) is the Von Mises stress, \(\sigma_t\) is the maximum principal stress and \(\varphi\) is stress state index defining the multi-axial stress rupture criterion [24].

TABLE I.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (h^-1)</td>
<td>2.1618 \times 10^7</td>
</tr>
<tr>
<td>B (MPa^-1)</td>
<td>2.0524 \times 10^7</td>
</tr>
<tr>
<td>C (-)</td>
<td>1.8537</td>
</tr>
<tr>
<td>h (MPa)</td>
<td>2.4326 \times 10^7</td>
</tr>
<tr>
<td>H* (-)</td>
<td>0.5929</td>
</tr>
<tr>
<td>Kc (h^-1)</td>
<td>9.2273 \times 10^7</td>
</tr>
</tbody>
</table>

The intergranular cavitation damage varies from zero, for the material in the virgin state, to 1/3, when all of the grain boundaries normal to the applied stress have completely cavitated, at which time the material is considered to have failed [25]. Thus, the critical value of creep damage is set to 0.3333333. The time increment is set to 1.0 hour. Once the creep damage reaches the critical value, the program will stop execution and the results will be output automatically.

4.2 Validation
Preliminary validation of such software was performed and it was conducted via a two-dimensional uni-axial tension model given below.

![Uni-axial tension model and boundary conditions](image)

The length of a side is set to 1 meter. The Young’s modulus and Poisson’s ratio are set to 100000 KN/m² and 0.3 respectively. A uniformly distributed load 40 KN/m was applied on the top line of this uni-axial tension model. The boundary constraint conditions are given in Table II.

TABLE II.

<table>
<thead>
<tr>
<th>Node number</th>
<th>Constraint in x direction</th>
<th>Constraint in y direction</th>
<th>Load in x direction</th>
<th>Load in y direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node No.1</td>
<td>shut</td>
<td>open</td>
<td>0 KN/m²</td>
<td>10 KN/m²</td>
</tr>
<tr>
<td>Node No.2</td>
<td>open</td>
<td>open</td>
<td>0 KN/m²</td>
<td>20 KN/m²</td>
</tr>
<tr>
<td>Node No.3</td>
<td>open</td>
<td>open</td>
<td>0 KN/m²</td>
<td>10 KN/m²</td>
</tr>
</tbody>
</table>
5 FE results and discussion

The initial elastic stress obtained from FE software:

<table>
<thead>
<tr>
<th>Element number</th>
<th>Stress in x-direction</th>
<th>Stress in y-direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element No.1</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.2</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.3</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.4</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.5</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.6</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.7</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.8</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
</tbody>
</table>

The initial elastic stress and the stress involving creep deformation and stress updating are shown in Table IV and Table V, respectively. Both confirmed the uniform distribution of stresses, and the values of stress in Y direction obtained from FE software are correct, and the stress in X direction is negligible.

The stress obtained from FE software with the stress update program:

<table>
<thead>
<tr>
<th>Element number</th>
<th>Stress in x-direction</th>
<th>Stress in y-direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element No.1</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.2</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.3</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.4</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.5</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.6</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.7</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
<tr>
<td>Element No.8</td>
<td>-0.3388E-06</td>
<td>0.4000E+02</td>
</tr>
</tbody>
</table>

The percentage errors:

- Rupture time percentage error = \( \frac{104062 - 104062}{104062} \times 100 = 0.0596\% \)
- Strain rate percentage error = \( \frac{0.000065438 - 0.000065438}{0.000065438} \times 100 = 0.0581\% \)
- Creep strain percentage error = \( \frac{0.3333E+00 - 0.3333E+00}{0.3333E+00} \times 100 = 0.0747\% \)
- Damage percentage error = \( \frac{0.3334 - 0.3334}{0.3333} \times 100 = 0.02\% \)

Other results are shown in Tables VI to VII. They show the results obtained from the FE software do agree with the expected theoretical values and the percentage errors are negligible.

In the current version, Euler forward integration subroutine, developed by colleague [22] was adopted here. Rupture time, strain rate, creep strain and damage obtained from FE software have been revealed that have a good agreement with the theoretical values obtained from the excel program. Work is ongoing in this area and will be reported in future.

6 Conclusion

This paper reports the finite element method based on CDM to design FE software for creep damage mechanics. It presents the structure of the new FE software and the use of existing FE library in obtaining such computational tool via an approach for stress and field variable updating. It further investigates preliminary validation of current version of such software via a uni-axial tension model.

The immediate future development work includes: 1) multi-materials; 2) implementing R-K integration scheme; 3) intelligent and practical control of time step; 4) removal of failed element and update stiffness matrix; and 5) further validation.

7 Reference