Netherlands Institute for Metals Research
Eindhoven University of Technology
TNO Built Environment and Geosciences

Report no. 9

Local buckling of slender aluminium sections exposed to fire

Implementation of material model in DIANA

Date June, 2007

Author(s) Johan Maljaars

Contact data
T +31 15 276 34 64
F +31 15 276 30 18
E johan.maljaars@tno.nl

Van Mourik Broekmanweg 6
P.O. Box 49
2600 AA Delft
The Netherlands

Number of pages 41
Number of Annexes 2
Sponsor NIMR
Project name PhD Local buckling of slender aluminium sections exposed to fire

Summary

A constitutive model was developed for the material properties of fire exposed aluminium alloys. This so-called Dorn-Harmathy material model is implemented in the finite element program DIANA, both for a uniaxial and a multiaxial stress condition, via a user supplied subroutine. This report gives the strategy followed in the implementation of the material model and the validation of the model.

The routines are implemented in such a way that thermal expansion and residual stresses can be applied together with the constitutive model. The tangent stiffness matrix was estimated numerically.

Convergence of the routines was sufficiently fast (for a model consisting of a single element: 3 iterations maximum for a time increment of 2 minutes in a transient state situation).

The routines were checked with uniaxial tests and hand calculations with creep, transient state and steady state conditions. The routines were also checked with hand calculations on stress relaxation and multiaxial stress conditions. The routines gave equal results in terms of stresses and strains as the hand calculations.

Contents

1 Introduction

This report is a background document to a PhD research on local buckling of slender aluminium sections exposed to fire.

A constitutive model was developed for the material properties of fire exposed aluminium alloys (see background report no 4 [1]). This so-called Dorn-Harmathy material model is implemented in the finite element program DIANA via a user supplied subroutine. This report gives the strategy followed in the implementation of the material model and gives the validation for uniaxial cases.

Chapter 2 gives a summary of the uniaxial material model and the extension of this model to the multiaxial case. The basic strategies for the implementation are explained in chapter 3 and the fortran routines are given in Annex A. Chapter 4 gives a comparison between simulations with the implemented material model and hand calculations for uniaxial and multiaxial cases. The results are also compared with uniaxial tensile tests. Conclusions are given in chapter 5.

2 Dorn-Harmathy material model

2.1 Uniaxial material model

The Dorn-Harmathy material model is based on a description of temperature dependent creep. Creep is on-going deformation of material when a constant temperature and stress are applied. The creep process is normally divided in three stages, i.e. a primary stage, with a decreasing strain rate in time, a secondary stage, with constant strain rate and a tertiary stage, with increasing strain rate (Figure 2-1). At the end of the tertiary creep stage, creep rupture occurs. Creep thus involves time dependent deformation and fracture of materials.

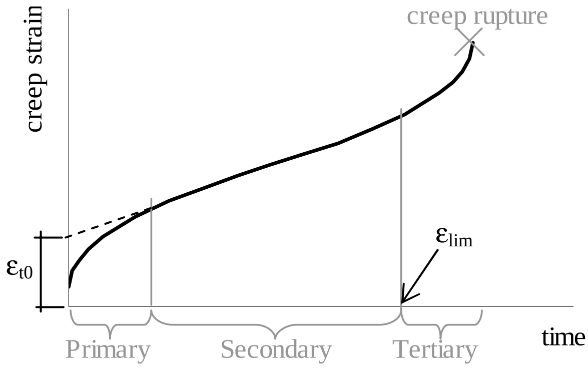

Figure 2-1 – Creep deformation of metals divided in primary, secondary and tertiary creep

A model has been developed with which it is possible to describe the creep curve. The model is validated with tests in background document no 4. The constitutive equations representing the creep curve are below.

$$\begin{aligned} \varepsilon_t \le \varepsilon_{\lim}: &\quad \frac{d\varepsilon_t}{dt} = Z(\sigma)\cdot e^{\frac{-Q}{RT}}\cdot \coth^2\left(\frac{\varepsilon_t}{\varepsilon_{t,0}(\sigma)}\right) \\ \varepsilon_t > \varepsilon_{\lim}: &\quad \frac{d\varepsilon_t}{dt} = Z(\sigma)\cdot e^{\frac{-Q}{RT}}\cdot \coth^2\left(\frac{\varepsilon_t}{\varepsilon_{t,0}(\sigma)}\right)\cdot\frac{\varepsilon_t}{\varepsilon_{\lim}} \end{aligned} \quad (1)$$

Background document no 4 has shown that the model can also be applied in case of increasing temperature, such as in case of fire exposure.
In creep tests, it was found that almost all creep strain is irreversible. The model developed assumes that all creep strain is irreversible (visco-plastic strain). The total mechanical strain as a function of temperature and stress is given in equation (2):

$$\varepsilon_{tot}(\sigma,T,t) = \varepsilon_{el} + \varepsilon_t + \varepsilon_{therm} = \frac{\sigma}{E(T)} + \int_0^t \dot{\varepsilon}_t(\sigma,T,t)\,\mathrm{d}t + \int_0^t \dot{T}\cdot\alpha_{th}(T)\,\mathrm{d}t \quad (2)$$

With
α_{th} = coefficient of linear thermal expansion according to equation (6) [/K]
ε_{lim} = strain at which the strain rate increases (strain at beginning of tertiary stage), a material dependent parameter [-]
ε_t = creep strain [-]

$\dot{\varepsilon}_t$ = creep strain rate [/min]
ε_{tot} = total strain [-]
$\varepsilon_{t,0}$ = primary creep strain rate according to equation (3) [-]
σ = stress [N/mm^2]
E = modulus of elasticity, according to equation (5) [N/mm^2]
Q = activation energy (material parameter) [J/mol]
R = universal gas constant (8,3144 J/mol K)
T = absolute temperature [K]
Z = Zener Holloman parameter according to equation (4) [/min]
t = time [min]

$$\varepsilon_{t0}(\sigma) = D \cdot \sigma^m \quad (3)$$

$$Z(\sigma) = A(\sinh \alpha\sigma)^n \quad (4)$$

With α, A, D, m and n are material dependent parameters.
More information is to be found in background report no 4.

Background report no 4 shows that equation (5) gives an appropriate description of the modulus of elasticity. The coefficient of thermal expansion is described with equation (6).

$$E(\theta) = EA + EB \cdot \theta + EC \cdot \theta^2 \quad (5)$$

$$\alpha_{th}(\theta) = \alpha A + \alpha B \cdot \theta \quad (6)$$

With EA, EB, EC, αA and αB are material dependent parameters, and θ is the aluminium temperature in degrees Celsius.

2.2 Modifications of the model

The model is modified such as to obtain a robust model, which is able to handle any combination of stress, strain and temperature.

The original equations (1) do not give correct answers for the following cases:
- A strain equal to zero returns infinity for the parts of equations (1) containing coth, and thus the equation cannot be solved for $\varepsilon = 0$;
- A stress equal to zero gives zero for ε_{t0}, which returns infinity for the fraction between brackets in equations (1);
- Creep tests showed that a specimen first subjected to tension and subsequently to compression gives a new period of primary creep at the moment the stress changes from sign (black curve in Figure 2-2). This is not represented with the model. Instead, the model predicts a period of primary creep where the creep strain ε_t is equal to zero. This is not always appropriate (grey curve in Figure 2-2).

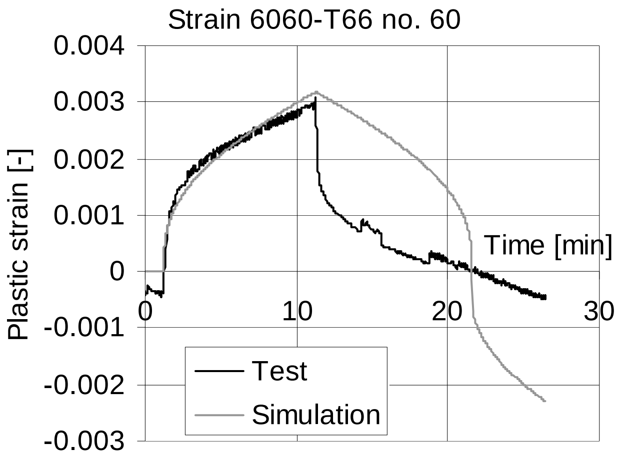

Figure 2-2 – Plastic strain as a function of time in a test first subjected to tension and after 12 minutes subjected to compression and the simulation of this plastic strain curve with the model

The first two problems are solved by increasing the creep strain ε_t and the value of ε_{t0} with a small number, so that the values cannot be equal to zero. The number must be so small as to have a negligible influence on the results when ε_t or ε_{t0} are not equal to zero. A number of $1 \cdot 10^{-7}$ was selected, which results in the following equations:

$$
\begin{aligned}
\varepsilon_t \le \varepsilon_{\lim}: \quad & \frac{d\varepsilon_t}{dt} = Z(\sigma)\cdot \mathrm{e}^{\frac{-Q}{RT}} \cdot \coth^2\left(\frac{\varepsilon_t + 1\cdot 10^{-7}}{\varepsilon_{t,0}(\sigma) + 1\cdot 10^{-7}}\right) \\
\varepsilon_t > \varepsilon_{\lim}: \quad & \frac{d\varepsilon_t}{dt} = Z(\sigma)\cdot \mathrm{e}^{\frac{-Q}{RT}} \cdot \coth^2\left(\frac{\varepsilon_t + 1\cdot 10^{-7}}{\varepsilon_{t,0}(\sigma) + 1\cdot 10^{-7}}\right)\cdot \frac{\varepsilon_t}{\varepsilon_{\lim}}
\end{aligned}
\tag{7}
$$

In order to solve the third problem, it is first investigated in which cases primary creep occurs. In the creep tests carried out, the following is observed concerning primary creep:

- primary creep does not (or hardly) occur when the stress is reduced, up to the ‘original’ stress level of zero N/mm^2 (i.e. completely unloaded);
- primary creep does occur when the stress is brought *beyond* the ‘original’ stress level of zero N/mm^2 (i.e. the stress changes in sign, from tension to compression or vice versa). Primary creep occurs each time the stress changes from sign.
- primary creep occurs when the stress is increased (test 9). Based on a number of tests, it is concluded that the total primary creep strain evolved in a material subjected to a stepwise increased stress, is equal to the primary creep strain in a material in which the specimen is stressed at once.

A simple and effective description for the occurrence of primary creep, which satisfies the observations above, is that primary creep starts each time the stress changes from sign. At that moment, the creep history is ‘erased’ and the specimen acts as if it is not loaded before.

The model is modified in such a way that it satisfies this description of primary creep. Instead of the total creep strain ε_t, a modified creep strain ε_t^* is introduced in the second part of the equation:

$$\begin{aligned} \varepsilon_t \le \varepsilon_{\lim}: \quad & \frac{d\varepsilon_t}{dt} = Z(\sigma)\cdot e^{\frac{-Q}{RT}}\cdot \coth^2\left(\frac{\varepsilon_t^* + 1\cdot 10^{-7}}{\varepsilon_{t,0}(\sigma)+1\cdot 10^{-7}}\right) \\ \varepsilon_t > \varepsilon_{\lim}: \quad & \frac{d\varepsilon_t}{dt} = Z(\sigma)\cdot e^{\frac{-Q}{RT}}\cdot \coth^2\left(\frac{\varepsilon_t^* + 1\cdot 10^{-7}}{\varepsilon_{t,0}(\sigma)+1\cdot 10^{-7}}\right)\cdot \frac{\varepsilon_t^*}{\varepsilon_{\lim}} \end{aligned} \tag{8}$$

A description of the initiation of primary creep that satisfies most cases is obtained when specifying that the modified creep strain ε_t^* is equal to the creep strain ε_t, but starts at zero when the stress changes form sign. Such a description would, however, result in an improper description for example in case of a creep test, where halfway the test the stress would vary around zero during a certain period. This model would result in a new period of primary creep at the end of the period with $\sigma \approx 0$ (black curve in Figure 2-4 b.), while this is not observed in tests (Figure 2-3).

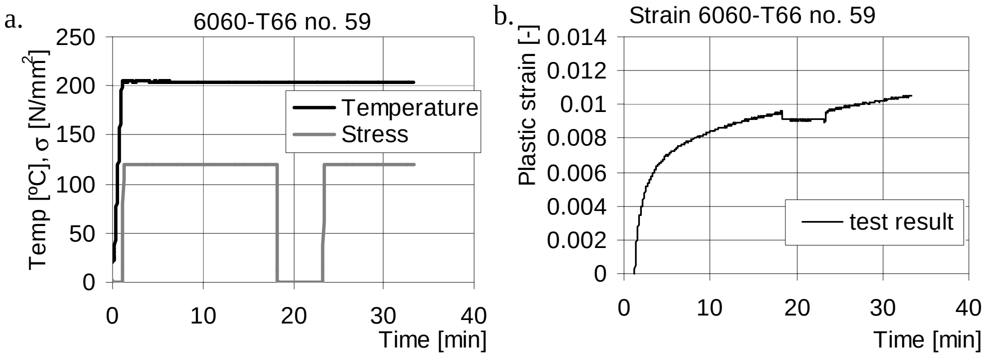

Figure 2-3 – Creep test with a period without load halfway the test
a. Temperature and stress applied in the test
b. Resulting plastic strain of the test

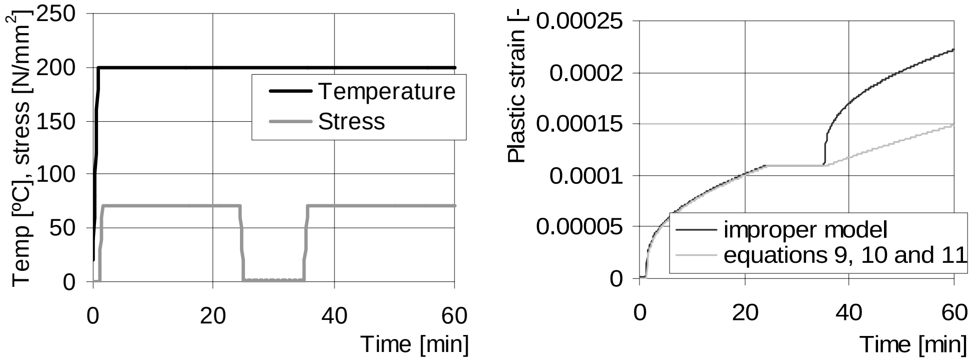

Figure 2-4 – Simulation of strain in a creep test with a period without load halfway the test
a. Temperature and stress in the simulation
b. Resulting plastic strain of the simulation

The following despcription of ε_t^* was selected to appropriately describe the creep strain curve for cases where the stress changes from sign.

$$\varepsilon_t^* = \begin{cases} \varepsilon_{t\ tens}^* & if \quad \sigma \geq 0 \\ \varepsilon_{t\ comp}^* & if \quad \sigma < 0 \end{cases} \tag{9}$$

$$\varepsilon_{t\ tens}^* = \max\left(0, \int_0^t \dot{\varepsilon}_t dt\right) \tag{10}$$

$$\varepsilon_{t\ comp}^* = \max\left(0, -\int_0^t \dot{\varepsilon}_t dt\right) \tag{11}$$

The result of a simulation with this model is indicated with a grey curve in Figure 2-4 b.

2.3 Extension to multi-axial case

The material model presented in the previous paragraph is calibrated and validated for a uniaxial stress condition with tests. In order to be used also for two dimensional or three dimensional elements, such as shell or solid elements, it is necessary to extend the theory to a multiaxial stress condition. In extending the theory, the following assumptions are made:

1. Assuming isotropic material, the principal directions of stress and strain should coincide;
2. In creep tests, it has been observed that the volume remains constant (Kraus [2]). Thus the plastic value for the Poisson ratio of ν_p = 0,5 applies for creep strains;
3. In creep tests, it has been observed that creep strains do not develop under a hydrostatic stress condition, i.e. a situation in which a normal stress is applied in all directions (Kraus). The Von Mises yield criterion (equation (12)), usually applied for aluminium at ambient temperature, satisfies this observation. It is assumed that the Von Mises yield criterion, also applies for creep strains at elevated temperature;

$$\sigma_{VM} = \sqrt{\sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2 - \sigma_{xx}\sigma_{yy} - \sigma_{yy}\sigma_{zz} - \sigma_{zz}\sigma_{xx} + 3\tau_{xy}^2 + 3\tau_{yz}^2 + 3\tau_{zx}^2} \tag{12}$$

In which σ_{VM} is the Von Mises yield stress, σ-symbols indicate normal stresses and τ-symbols indicate shear stresses. The equation usually considered for the accompanying effective (or Von Mises) strain is:

$$\varepsilon_{t,eff} = \frac{2}{3}\sqrt{\varepsilon_{t,xx}^2 + \varepsilon_{t,yy}^2 + \varepsilon_{t,zz}^2 - \varepsilon_{t,xx}\varepsilon_{t,yy} - \varepsilon_{t,yy}\varepsilon_{t,zz} - \varepsilon_{t,zz}\varepsilon_{t,xx} + \frac{3}{4}\gamma_{t,xy}^2 + \frac{3}{4}\gamma_{t,yz}^2 + \frac{3}{4}\gamma_{t,zx}^2} \tag{13}$$

The multiaxial stress formulation given below is based on the derivation in Kraus.
The strain rate in the Dorn Harmathy model depends on the stress. At a certain time and temperature, the relation between strain rate and stress for a uniaxial stress condition is written as:

$$\dot{\varepsilon}_t = C_1 \cdot \sigma \tag{14}$$

In the same way, it is possible to write for the 3D case:

$$\dot{\varepsilon}_{t,ij} = C_2 \cdot S_{ij} \tag{15}$$

In which $\dot{\varepsilon}_{t,ij}$ is the strain rate in direction ij according to equation (16)

$$\dot{\varepsilon}_{t,ij} = \begin{bmatrix} \dot{\varepsilon}_{t,xx} & \frac{1}{2}\dot{\gamma}_{t,xy} & \frac{1}{2}\dot{\gamma}_{t,zx} \\ \frac{1}{2}\dot{\gamma}_{t,xy} & \dot{\varepsilon}_{t,yy} & \frac{1}{2}\dot{\gamma}_{t,yz} \\ \frac{1}{2}\dot{\gamma}_{t,zx} & \frac{1}{2}\dot{\gamma}_{t,yz} & \dot{\varepsilon}_{t,zz} \end{bmatrix} \tag{16}$$

In which ε are the normal components and γ are the shear components of the strain tensor.
S_{ij} is a stress deviator tensor. As the hydrostatic stress has no effect on the creep strain rate, the stress deviator contains only the distortional components of the stress. The stress deviator thus consists of the following components:

$$S_{ij} = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{zx} \\ \tau_{xy} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{yz} & \sigma_{zz} \end{bmatrix} - \frac{1}{3}\begin{bmatrix} \sigma_{xx}+\sigma_{yy}+\sigma_{zz} & 0 & 0 \\ 0 & \sigma_{xx}+\sigma_{yy}+\sigma_{zz} & 0 \\ 0 & 0 & \sigma_{xx}+\sigma_{yy}+\sigma_{zz} \end{bmatrix}$$
(17)

In which the quantity 1/3 (σ_{xx} + σ_{yy} + σ_{zz}) is the hydrostatic stress.
From assumptions 1 and 3, it follows directly that the strain vector has to be perpendicular to the yield surface. This is represented by equation (15). In this equation, C2 is a scalar.

If a hydrostatic stress is applied in equation (15), i.e. $\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = \sigma$, it follows that all strain rate components are equal to 0, hence assumption 2 is satisfied.
The assumption of constancy of volume applied in equation (15) results: $\dot{\varepsilon}_{t,xx} + \dot{\varepsilon}_{t,yy} + \dot{\varepsilon}_{t,zz} = C_2\left(S_{11} + S_{22} + S_{33}\right) = 0$. Thus the relation satisfies all assumptions.

C_2 is determined by recognising that equation (15) should reduce to equation (14) for the uniaxial stress case. Consider a uniaxial test with σ_{xx} = σ and all other stress components equal to zero. In this case, $\dot{\varepsilon}_{t,xx} = \dot{\varepsilon}_t$ and $\dot{\varepsilon}_{t,yy} = \dot{\varepsilon}_{t,zz}$. Using the assumption of constant volume, i.e. $\dot{\varepsilon}_{t,xx} + \dot{\varepsilon}_{t,yy} + \dot{\varepsilon}_{t,zz} = 0$, it follows that $\dot{\varepsilon}_{t,yy} = \dot{\varepsilon}_{t,zz} = -1/2\,\dot{\varepsilon}_t$ and all shear strain rates are equal to zero.

If all stress and strain rate components of this uniaxial case are applied in equation (15), it follows that
$\dot{\varepsilon}_{t,eff} = \dot{\varepsilon}_t$, σ_{VM} = σ and an expression is found for C_2:

$$\dot{\varepsilon}_{t,eff} = \frac{2}{3} C_2 \sigma_{VM} \quad \Rightarrow \quad C_2 = \frac{3}{2} \frac{\dot{\varepsilon}_{t,eff}}{\sigma_{VM}} \tag{18}$$

Since the effective strain rate and the von Mises stress are defined so as to reduce to the uniaxial strain and stress, it is possible to extend the model for the uniaxial strain rate (equations(1)) to the multiaxial case by introducing the effective quantities:

$$\varepsilon_{t,eff} \leq \varepsilon_{t.,lim}: \quad \dot{\varepsilon}_{t,eff} = Z(\sigma_{VM}) \cdot e^{\frac{-Q}{RT}} \cdot \coth^2 \left(\frac{\varepsilon_{t,eff}^{*} + 1 \cdot 10^{-7}}{\varepsilon_{t,0}(\sigma_{VM}) + 1 \cdot 10^{-7}} \right) \tag{19 a.}$$

$$\varepsilon_{t,eff} > \varepsilon_{t.,lim}: \quad \dot{\varepsilon}_{t,eff} = Z(\sigma_{VM}) \cdot e^{\frac{-Q}{RT}} \cdot \coth^2 \left(\frac{\varepsilon_{t,eff}^{*} + 1 \cdot 10^{-7}}{\varepsilon_{t,0}(\sigma_{VM}) + 1 \cdot 10^{-7}} \right) \frac{\varepsilon_{t,eff}}{\varepsilon_{t,\lim}} \tag{19 b.}$$

Equations (19) are applied in equation (18) to determine quantity C_2. The model for multiaxial creep is then completed by substitution of C_2 in equation (15):

$$\begin{pmatrix} \dot{\varepsilon}_{t,xx} \\ \dot{\varepsilon}_{t,yy} \\ \dot{\varepsilon}_{t,zz} \\ \dot{\gamma}_{t,xy} \\ \dot{\gamma}_{t,yz} \\ \dot{\gamma}_{t,xz} \end{pmatrix} = C_2 \cdot \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 \end{pmatrix} \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{pmatrix} \tag{20 a.}$$

Where $\varepsilon_{t,eff} \leq \varepsilon_{t.lim}$: $$C_2 = \frac{1}{2} \frac{Z(\sigma_{VM}) \cdot e^{\frac{-Q}{RT}}}{\sigma_{VM}} \cdot \coth^2 \left(\frac{\varepsilon_{t,eff}^{*}}{\varepsilon_{t0}(\sigma_{VM})} \right) \tag{20 b.}$$

$$\varepsilon_{t,eff} > \varepsilon_{t.lim}: \quad C_2 = \frac{1}{2} \frac{Z(\sigma_{VM}) \cdot e^{\frac{-Q}{RT}}}{\sigma_{VM}} \cdot \coth^2 \left(\frac{\varepsilon_{t,eff}^{*}}{\varepsilon_{t0}(\sigma_{VM})} \right) \cdot \frac{\varepsilon_{t,eff}}{\varepsilon_{t,\lim}} \tag{20 c.}$$

The constitutive relations for the multiaxial case are described with equations (20 a.), (20 b.) and (20 c.). These relations are used in the remaining part of this document.

3 Description of basic routines in DIANA

3.1 Description of predefined user supplied subroutine

DIANA offers the possibility to supply Fortran source in a predefined subroutine in order to implement a new material model.

The user supplied subroutine gives the strain at the start of the step ε_0, the strain increment in the current iteration $\Delta\varepsilon$ and the stress at the start of the step σ. The subroutine should give the stress at the end of the iteration σ and the tangent stiffness matrix D. This tangent stiffness matrix is used to provide a new strain increment $\Delta\varepsilon$ in the new iteration.

Apart from the basic variables ε_0 and σ, the predefined subroutine also gives the following values which are used in the current model:
Δt = time increment
T_0 = temperature at the start of the step
ΔT = temperature increment

Further, it is possible to apply user supplied material variables in the subroutine as well as user supplied state variables. Material variables are set at the beginning of the analysis and do not change during the analysis, while state variables are updated at the end of each step.

The remaining part of this chapter gives the description of the basic routines of the material model as determined in the predefined user supplied subroutine.

3.2 Specified variables

The user supplied material variables specified for the material model are parameters α, Q, R, A, D, m, n and ε_{lim}. for the Dorn Harmathy model (equation (1)), parameters EA , EB and EC for the modulus of elasticity (equation (5)), the coefficient of lateral contraction ν and αA and αB for the coefficient of thermal expansion (equation (6)).

The state variables defined are ε^*_{tens} , ε^*_{tens} , ε^* and ε_t in all directions, the thermal strain ε_{th}, the time t and the preststrain ε_{pre} (strain caused by residual stress at the start of the calculation) in all directions.

3.3 Basic routine of the uniaxial model

In the user supplied subroutine, the stress has to be updated in the routine based on the strain increment. In the material model, however, the creep strain rate is determined as a function of the stress. A numerical iteration procedure is applied in order to determine the stress and creep strain. The following steps are applied:

1. The maximum and minimum possible values of the creep strain increment are determined (being the total strain increment and zero) $\Delta\varepsilon_{t,max}$ and $\Delta\varepsilon_{t,min}$;
2. An estimation of the creep strain increment is determined, being the average value of the maximum and minimum values of the creep strain increment $\Delta\varepsilon_{t,est}$. The

estimated creep strain is equal to the creep strain at the start of the increment $\varepsilon_{t,0}$ plus the average creep strain increment: $\varepsilon_{t,est} = \varepsilon_{t,0} + ½ (\Delta\varepsilon_{t,max} + \Delta\varepsilon_{t,min})$;

3. The elastic strain is equal to the total strain minus the creep strain and the thermal strain, thus $\varepsilon_{elas,est} = \varepsilon - \varepsilon_{t,est} - \varepsilon_{th}$. The stress is then determined for the estimated creep strain, being $\sigma_{est} = E (\varepsilon - \varepsilon_{t,est} - \varepsilon_{th})$;
4. Equations (8) are applied to determine the creep strain rate corresponding to this stress: $\dot{\varepsilon}_{t,1}(\sigma_{est})$;
5. The creep strain increment, belonging to this stress, is determined by multiplying the average creep strain rate with the time increment. The average creep strain rate is the average of the creep strain rate at the start of the increment and the creep strain rate at the end of the increment: $\Delta\varepsilon_{t,1} = \Delta t \cdot \frac{1}{2}\left(\dot{\varepsilon}_{t,0} + \dot{\varepsilon}_{t,1}\right)$
6. The new value of the creep strain increment, obtained in step 5, is compared with the estimated creep strain increment of step 2: $\Delta\varepsilon_{t,1} \leftrightarrow \Delta\varepsilon_{t,est}$
   - In case the new value of the creep strain is smaller than the estimated value, the maximum possible value of the creep strain $\Delta\varepsilon_{t,max}$ is replaced by the estimated creep strain ($\Delta\varepsilon_{t,1} < \Delta\varepsilon_{t,est} \rightarrow \Delta\varepsilon_{t,max} = \Delta\varepsilon_{t,est}$) and the procedure is repeated from step 2 onwards;
   - In case the new value of the creep strain is larger than the estimated value, the minimum possible value of the creep strain is replaced by the estimated creep strain ($\Delta\varepsilon_{t,1} > \Delta\varepsilon_{t,est} \rightarrow \Delta\varepsilon_{t,min} = \Delta\varepsilon_{t,est}$) and the procedure is repeated from step 2 onwards;
   - In case the new value of the creep strain is equal to the estimated value, the required values of the creep strain increment and the stress are determined and the procedure is finished.

Note that the strain rate at the start of the increment, $\dot{\varepsilon}_{t,0}$, which is used in step 5, can be determined directly with equations (8), as the stress at the start of the increment σ_0 is known.

3.4 Tangent stiffness matrix in the uniaxial model

In order to determine the tangential stiffness $D = d\sigma / d\varepsilon$, the derivative of the creep strain function should be determined. This is however a difficult task. Instead, an alternative was applied in which the tangential stiffness is determined by running the procedure again with a slightly different value of the strain increment (ε_{err}). The procedure then results in a slightly different value for the stress increment (σ_{err}). A numerical estimation of the tangential stiffness is obtained by:

$$\frac{d\sigma}{d\varepsilon} \approx \frac{\sigma - \sigma_{err}}{\varepsilon - \varepsilon_{err}} = \frac{\sigma - \sigma_{err}}{d\varepsilon - d\varepsilon_{err}} \tag{21}$$

In case $d\varepsilon$ is positive, the value applied for $d\varepsilon_{err}$ is the maximum value of $\left(d\varepsilon + 1\cdot 10^{-5}\right)$ and $\left(d\varepsilon \cdot 1,01\right)$. In case $d\varepsilon$ is negative, the value applied for $d\varepsilon_{err}$ is the minimum value of $\left(d\varepsilon - 1\cdot 10^{-5}\right)$ and $\left(d\varepsilon \cdot 1,01\right)$.

The tangent stiffness matrix is used to provide a new strain increment $\Delta\varepsilon$ in the new iteration. Applied on a single truss element, the analysis converged in three iterations in case of a maximum step size of 2 minutes up to a strain of 2 %. This is considered as sufficiently fast for the procedure, so that the numerical estimation of *D* is sufficiently accurate.

The fortran code for the uniaxial material model is given in Annex A.

3.5 Incorporation of residual stress

Due to e.g. welding, residual stresses may be present in a structure. In DIANA, it is possible to define residual stresses at the start of the analysis. In the predefined user supplied subroutine, this appears as a stress at the start of the first increment. In the material model implemented, this residual stress is handled as follows.
At time $t = 0$, the stress σ_0 is used to define a prestrain ε_{pre}, with:

$$\varepsilon_{pre} = \frac{\sigma_0 (t=0)}{E(T)} \tag{22}$$

Instead of the total strain according to equation (2), an artificial total strain is applied in the model, being:

$$\varepsilon_{tot} = \varepsilon_{el} + \varepsilon_{th} + \varepsilon_t + \varepsilon_{pre} \tag{23}$$

The rest of the procedure is identical to paragraph 3.3.

3.6 Constitutive equations for multiaxial model

The relation between the elastic strain components and the stress components is given in equation (25). The elastic strains are according to equation (26) and the creep (plastic) strains are according to equation (27).
Note that components $\varepsilon_{t,0,ij}$ are the creep strains at the start of the increment and should not be confused with ε_{t0} , which is the projection back to zero time of the secondary strain rate curve according to equation (3).

$$\begin{pmatrix} \varepsilon_{elas,xx} \\ \varepsilon_{elas,yy} \\ \varepsilon_{elas,zz} \\ \gamma_{elas,xy} \\ \gamma_{elas,yz} \\ \gamma_{elas,zx} \end{pmatrix} = \frac{1}{E} \cdot \begin{pmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{pmatrix} \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{pmatrix} \tag{25}$$

$$\begin{pmatrix} \varepsilon_{elas,xx} \\ \varepsilon_{elas,yy} \\ \varepsilon_{elas,zz} \\ \gamma_{elas,xy} \\ \gamma_{elas,yz} \\ \gamma_{elas,zx} \end{pmatrix} = \begin{pmatrix} \varepsilon_{xx} - \varepsilon_{t,xx} - \varepsilon_{th} \\ \varepsilon_{yy} - \varepsilon_{t,yy} - \varepsilon_{th} \\ \varepsilon_{zz} - \varepsilon_{t,zz} - \varepsilon_{th} \\ \gamma_{xy} - \gamma_{t,xy} \\ \gamma_{yz} - \gamma_{t,yz} \\ \gamma_{zx} - \gamma_{t,zx} \end{pmatrix} \quad (26)$$

$$\begin{pmatrix} \varepsilon_{t,xx} \\ \varepsilon_{t,yy} \\ \varepsilon_{t,zz} \\ \gamma_{t,xy} \\ \gamma_{t,yz} \\ \gamma_{t,xz} \end{pmatrix} = \begin{pmatrix} \varepsilon_{t,0,xx} \\ \varepsilon_{t,0,yy} \\ \varepsilon_{t,0,zz} \\ \gamma_{t,0,xy} \\ \gamma_{t,0,yz} \\ \gamma_{t,0,xz} \end{pmatrix} + \frac{1}{2}\Delta t \cdot \begin{pmatrix} \dot{\varepsilon}_{t,xx} + \dot{\varepsilon}_{t,0,xx} \\ \dot{\varepsilon}_{t,yy} + \dot{\varepsilon}_{t,0,yy} \\ \dot{\varepsilon}_{t,zz} + \dot{\varepsilon}_{t,0,zz} \\ \dot{\gamma}_{t,xy} + \dot{\gamma}_{t,0,xy} \\ \dot{\gamma}_{t,yz} + \dot{\gamma}_{t,0,yz} \\ \dot{\gamma}_{t,xz} + \dot{\gamma}_{t,0,xz} \end{pmatrix} \quad (27)$$

The strain rates at the start of the step, $\dot{\varepsilon}_{t,0,ij}$ can be determined directly as the strain and stress components at the start of the step ($\varepsilon_{0,ij}$ and $\sigma_{0,ij}$) are known. Substitution of equations (27) and (26) in equation (25) results in 6 equations with 12 unknown parameters, being $\dot{\varepsilon}_{t,ij}$ and σ_{ij}. The other 6 equations are given by substitution of equation (27) in equation (20).

Elaboration of these 12 equations gives the following relations for the individual stress components:

$$\sigma_{xx} = \frac{1 - CA^2 \cdot CB - \left(CA - CA^2\right)\left(CC + CD\right)}{1 - 3 \cdot CA^2 + 2 \cdot CA^3} \quad (28)$$

$$\sigma_{yy} = \frac{1 - CA^2 \cdot CC - \left(CA - CA^2\right)\left(CB + CD\right)}{1 - 3 \cdot CA^2 + 2 \cdot CA^3} \quad (29)$$

$$\sigma_{zz} = \frac{1 - CA^2 \cdot CD - \left(CA - CA^2\right)\left(CB + CC\right)}{1 - 3 \cdot CA^2 + 2 \cdot CA^3} \quad (30)$$

$$\sigma_{xy} = \frac{E\left(\varepsilon_{xy} - \varepsilon_{t,0,xy} - \frac{1}{2}\Delta t \cdot \dot{\varepsilon}_{t,0,xy}\right)}{2 + 2\nu + 3E \cdot C2 \cdot \Delta t} \quad (31)$$

$$\sigma_{yz} = \frac{E\left(\varepsilon_{yz} - \varepsilon_{t,0,yz} - \frac{1}{2}\Delta t \cdot \dot{\varepsilon}_{t,0,yz}\right)}{2 + 2\nu + 3E \cdot C2 \cdot \Delta t} \quad (32)$$

$$\sigma_{zx} = \frac{E\left(\varepsilon_{zx} - \varepsilon_{t,0,zx} - \frac{1}{2}\Delta t \cdot \dot{\varepsilon}_{t,0,zx}\right)}{2 + 2\nu + 3E \cdot C2 \cdot \Delta t} \tag{33}$$

$$CA = \frac{-\nu - \frac{1}{2}\Delta t \cdot C2 \cdot E}{1 + \Delta t \cdot C2 \cdot E} \tag{34}$$

$$CB = \frac{E\left(\varepsilon_{xx} - \varepsilon_{t,0,xx} - \frac{1}{2}\Delta t \cdot \dot{\varepsilon}_{cr,0,xx} - \varepsilon_{th}\right)}{1 + \Delta t \cdot C2 \cdot E} \tag{35}$$

$$CC = \frac{E\left(\varepsilon_{yy} - \varepsilon_{t,0,yy} - \frac{1}{2}\Delta t \cdot \dot{\varepsilon}_{cr,0,yy} - \varepsilon_{th}\right)}{1 + \Delta t \cdot C2 \cdot E} \tag{36}$$

$$CD = \frac{E\left(\varepsilon_{zz} - \varepsilon_{t,0,zz} - \frac{1}{2}\Delta t \cdot \dot{\varepsilon}_{cr,0,zz} - \varepsilon_{th}\right)}{1 + \Delta t \cdot C2 \cdot E} \tag{37}$$

3.7 Basic routine for multiaxial model

The unknown parameters in equations (28) up to (37) are the stress components and parameter C2, i.e. 7 unknown parameters and 6 equations. The set of equations is completed by considering the relation between parameter C2 and the Von Mises stress in equations 20 b. and 20 c. This set of equations is solved numerically in a routine in which the parameter C2 is updated.

First, maximum and minimum values for C_2 are determined. It is noted that parameter C_2 increases for increasing stress. The minimum value for C_2 is zero (i.e. the strain rate is zero): $C_{2,min} = 0$. The maximum possible value for C_2 (denoted as $C_{2,max}$) is found for the maximum possible Von Mises stress.

The routine consists of the following steps

1. An upper limit for the Von Mises stress $\sigma_{vm,max}$ is obtained according to the equations in Annex A.
2. This Von Mises stress $\sigma_{vm,max}$ and the plastic strain at the start of the increment ε^*_{VM}, are applied in equations (20 b.) and (20 c.) in order to obtain an upper limit for $C_{2,max}$
3. A first estimation of parameter C_2 (denoted as $C_{2,est}$) is obtained by taking the average value of the maximum and minimum values: $C_{2,est} = C_{2,max} + C_{2,min}$;
4. The stress components $\sigma_{est,ij}$ are determined using equations (28) up to (37).
5. The Von Mises stress $\sigma_{VM,est}$ is determined from the stress components $\sigma_{est,ij}$;
6. The creep strain increments $\Delta\varepsilon_{t,ij}$ are determined, being the total strain increments minus the elastic part of this increment. The elastic strain increments are calculated from the individual stress components $\sigma_{est,ij}$;

7. The creep strains at the end of the iteration, $\varepsilon^*_{est,ij}$, are determined by adding the creep strain increments to the creep strains at the start of the increment: $\varepsilon^*_{est,ij}$ = $\varepsilon^*_{est,ij}$ + $\Delta\varepsilon_{t,ij}$;
8. The effective value of the creep strain at the end of the increment, $\varepsilon^*_{eff,est}$, is determined (equation (13));
9. A new value for parameter C_2 is determined by calculating equations (20 b.) and (20 c.), using $\sigma_{VM,est}$ and $\varepsilon^*_{eff,est}$,
10. The new value for C_2, obtained in step 9, is compared with the estimated value of step 3: $C_2 \leftrightarrow C_{2,est}$
    - In case the new value of C_2 is smaller than the estimated value, the maximum possible value $C_{2,max}$ is replaced by the estimated creep strain ($C_2 < C_{2,est} \rightarrow C_{2,max} = C_{2,est}$) and the procedure is repeated from step 3 onwards;
    - In case the new value of C_2 is larger than the estimated value, the minimum possible value $C_{2,min}$ is replaced by the estimated creep strain $C_2 > C_{2,est} \rightarrow C_{2,min} = C_{2,est}$) and the procedure is repeated from step 3 onwards;
    - In case the new value of C_2 is equal to the estimated value $C_{2,est}$, the required values of the creep strain increments and the stresses are determined and the procedure is finished.

The routine results in the stresses σ_{ij} corresponding with the strain $\varepsilon_{ij,0} + \Delta\varepsilon_{ij}$.

3.8 Tangent stiffness matrix in the multiaxial model

The tangent stiffness matrix D was determined numerically by running the procedure again with a slightly different value of the strain increment of one strain component ($\varepsilon_{err,kl}$). The procedure then results in slightly different values for the stresses ($\sigma_{err,ij}$). A numerical estimation of the tangential stiffness is obtained by:

$$\frac{d\sigma_{ij}}{d\varepsilon_{kl}} \approx \frac{\sigma_{ij} - \sigma_{err,ij}}{\varepsilon_{kl} - \varepsilon_{err,kl}} \qquad (38)$$

The value used for $\varepsilon_{err,kl}$ is equal as in the uniaxial case (paragraph 3.4).
Annex B gives the fortran code of the multiaxial model.

4 Validation of the routines

The results of simulations with the routines for the uniaxial and the multiaxial model are compared with tests and with hand calculations.

4.1 Description of the models

For this purpose, analyses are carried out of uniaxial truss elements L2TRU with the uniaxial model (model consisting of 1 element and model consisting of 10 elements), three-dimensional solid elements CHX60 with the multiaxial model (1 element and 10 x 10 x 10 elements) and three-dimensional shell elements CQ40S with the multiaxial model (1 element and 10 x 10 elements). In case of shell elements, the stress component parallel to the normal of the element σ_{zz} is by definition equal to zero. Nonetheless, the multiaxial model as described in chapter 3 can be applied. In an internal procedure following the user supplied subroutine, DIANA reformulates stresses and strains such as to result in σ_{zz} being equal to zero.

Material tests were only carried out with a uniaxial stress condition. Therefore, uniaxial stress conditions were applied in the simulations in order to check the routines. In case of the multiaxial model, all stress and strain components were compared with tests and hand calculations for the uniaxial stress condition. Besides, some basic three-dimensional cases were simulated, such as a uniform tensile stress in all directions.

Simulations are carried out with the material parameters determined in tensile tests for alloys 5083-H111 and 6060-T66. The parameters are summarised in Table 4.1.

Table 4.1 – Values for the material dependent parameters

Parameter	**Alloy 5083-H111**	**Alloy 6060-T66**
Q	152000 [J/mol K]	195000 [J/mol K]
A	$1{,}112 \cdot 10^{9}$ [/s]	$3{,}333 \cdot 10^{12}$ [/s]
α	0,025	0,019
n	3	4
D	$3{,}9 \cdot 10^{-10}$	$2{,}0 \cdot 10^{-18}$
m	3,4	7,45
ε_{lim}	--- (> 2 %)	0,002
EA	71000 [N/mm^2]	69000 [N/mm^2]
EB	-10 [N/mm^2 K]	-10 [N/mm^2 K]
EC	-0.21 [N/mm^2 K^2]	-0,21 [N/mm^2 K^2]
αA	$1{,}7 \cdot 10^{-5}$ [/ K]	$1{,}7 \cdot 10^{-5}$ [/ K]
αB	$2 \cdot 10^{-8}$ [/ K^2]	$2 \cdot 10^{-8}$ [/ K^2]

4.2 Simulation of creep tests

Creep tests were simulated with hand calculations and with the numerical model in DIANA. The mechanical strain according to the DIANA models (both uniaxial and multiaxial) was equal to that of the hand calculations. Figure 4-1 gives an example of a creep test with a constant temperature and a stepwise varied stress. In this figure, and in all other figures in this chapter, the mechanical strain is displayed, which is obtained by subtraction of the thermal strain from the total strain.

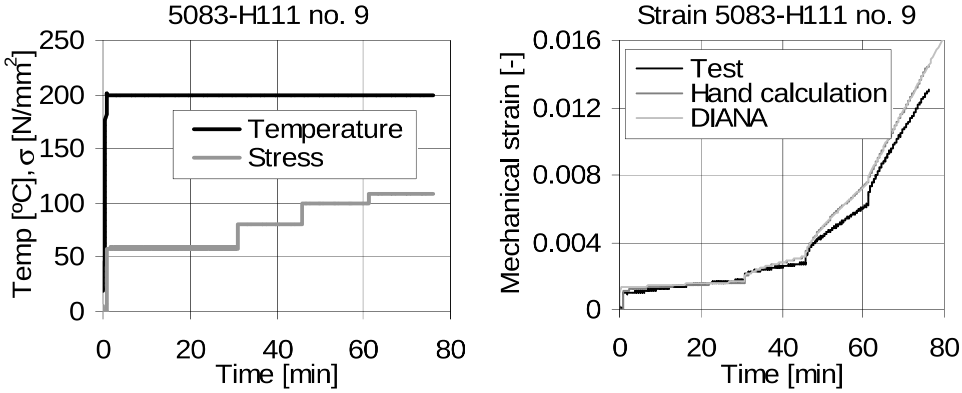

Figure 4-1 – Example of simulation of a creep test with stepwise increased stress (material 5083-H111)

Figure 4-2 gives an example of the simulation of a test with a tensile stress and subsequently a compression stress.

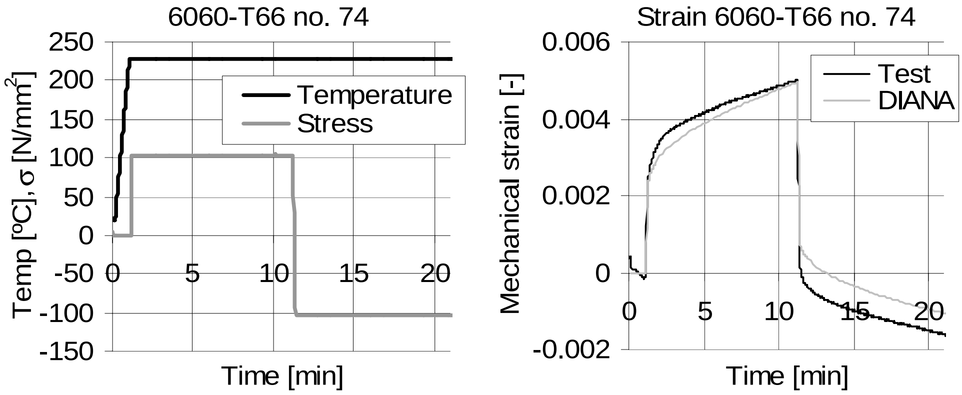

Figure 4-2 – Example of simulation of a creep test with a positive and subsequently a negative stress (material 6060-T66)

Calculations with a hydraulic stress state (σ_{xx} = σ_{yy} = σ_{zz} and shear components are zero) result, as required, in only elastic strain and no plastic strain.

4.3 Simulation of transient state tests

Transient state tests were simulated with the model. The mechanical strain according to the DIANA models (both uniaxial and multiaxial) was equal to that of the hand calculations. Figure 4-3 gives an example of a transient state test with a constant stress in time. Figure 4-4 gives an example of a transient state test with an increasing stress in time.

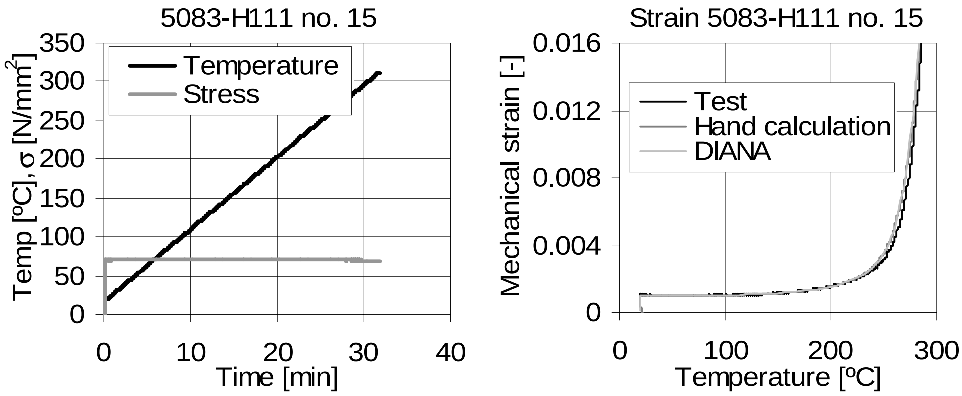

Figure 4-3 – Example of simulation of a transient state test with constant stress in time (material 5083-H111)

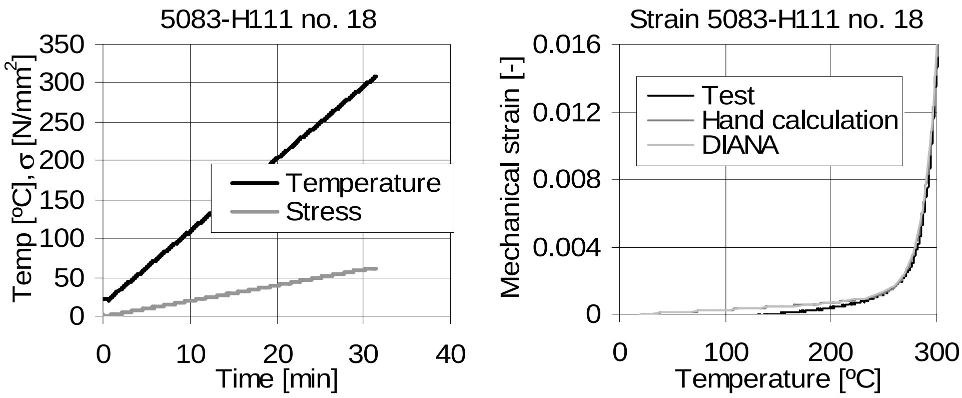

Figure 4-4 – Example of simulation of a transient state test with linear increasing stress in time (material 5083-H111)

4.4 Simulation of steady-state tests

Steady state tests, with constant temperature and constant strain rate, were carried out. The tests were simulated with DIANA. The measured specimen temperature and the strain rate measured with the strain gages were used as input parameters for the simulation. Output of the simulation is the stress. As equation (1) gives the strain as a function of the stress, these steady-state tests could not be simulated with a simple hand calculation.

Figure 4-5 gives $f_{u,test}$ and $f_{u,simulation}$ for alloy 5083-H111. There is a good agreement between measured and simulated values of the ultimate tensile strength (ratio $f_{u,simulation}$ / $f_{u,test}$ = 0,91 to 1,07, average value = 1,00).

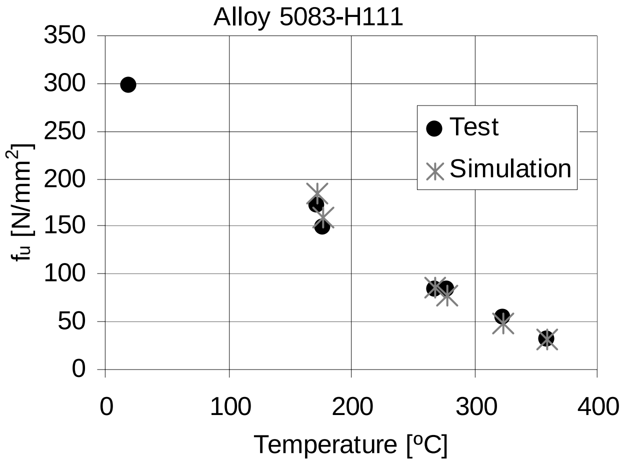

Figure 4-5 – Ultimate tensile strength as a function of the temperature of alloy 5083-H111, according to steady-state tests and according to simulation with model

In case of alloy 6060, simulations were made with model parameters based on creep test series 2005. Figure 4-6 and Figure 4-7 give $f_{u,test}$ and $f_{u,simulation}$ for the square hollow section tests and the angle tests of alloy 6060-T66, respectively. There is a worse agreement between measured and simulated values of the ultimate tensile strength as in case of alloy 5083-H111 (ratio $f_{u,simulation}$ / $f_{u,test}$ = 1,05 to 1,42, average value = 1,22). One reason for the fact that the simulated strain is larger than the measured strain, especially at higher temperatures, is that the material model is based on tests with a constant elevated temperature for approximately 15 minutes, while the period with constant temperature in the steady-state tensile tests was longer (approximately 40 minutes at f_u). This might have influenced the temper. Another reason for the difference between the measured and simulated strain is that the strain measurement with the strain gage was inaccurate, so that the strain rate applied as input parameter in the material model is inaccurate.

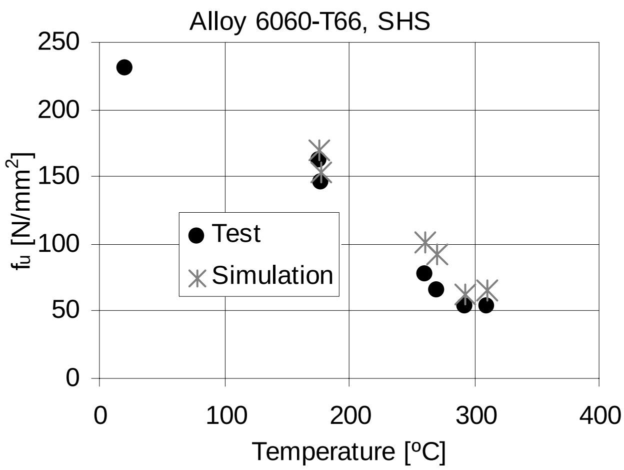

Figure 4-6 – Ultimate tensile strength as a function of the temperature of extruded square hollow sections of alloy 6060-T66, according to steady-state tests and according to simulation with model

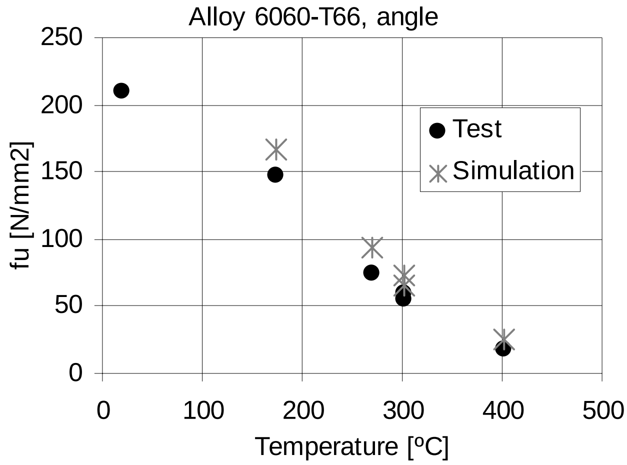

Figure 4-7 – Ultimate tensile strength as a function of the temperature of extruded angles of alloy 6060-T66, according to steady-state tests and according to simulation with model

4.5 Simulation of stress relaxation

Simulations were made with DIANA models in which the element nodes were restrained in each direction. In one direction, a residual stress was applied of 83 N/mm2, while the stress components in the other directions were zero at the start of the analysis. The residual stresses in a specimen relax when the specimen is exposed to an increasing temperature, due to a decreasing modulus of elasticity and the influence of creep. Figure 4-8 shows that the resulting relaxation of the stress in the DIANA models is equal to that of a hand calculation.

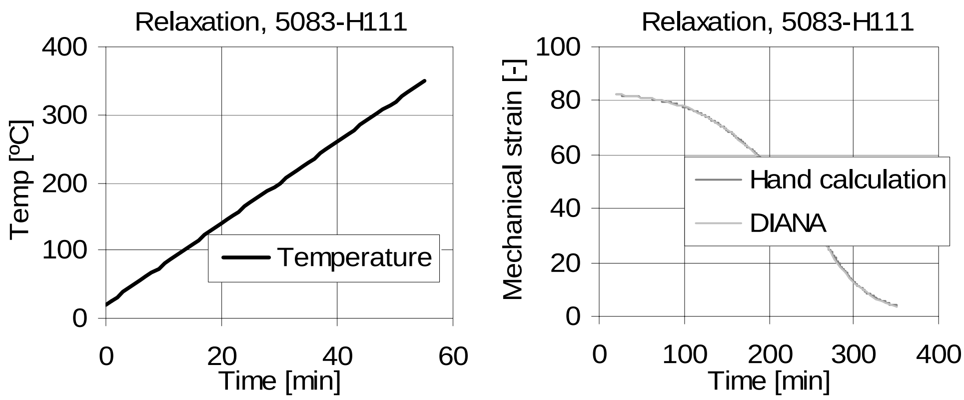

Figure 4-8 – Stress as a function of temperature in a simulation with initial stress σ_{ini} = 83 N/mm^2 while the strain is completely restrained (material 5083-H111)

4.6 Simulations of multiaxial stress conditions

Some basic analysis were carried out with multiaxial stress conditions (using the multiaxial model).

1. A creep analysis was carried out with shear stresses equal to zero and all normal stresses equal to one another (σ). This gives a hycrostatic stress condition, so that

the Von Mises stress is equal to zero. The analyses therefore results in creep strains equal to zero, as required;

2. A creep analysis was carried out with shear stresses equal to zero, one normal stress equal to zero and the other two normal stresses equal to one another (σ). The Von Mises stress is then equal to $\sigma_{VM} = \sigma$. The resulting strain was, as required, equal to that of a uniaxial test and the strain direction was in conformity with equation (20 a.): $\varepsilon_{xx} = \varepsilon_{yy}$ and $\varepsilon_{zz} = -(\varepsilon_{xx} + \varepsilon_{yy})$;
3. A creep analysis was carried out with a shear stress τ. The Von Mises stress is then equal to $\sigma_{VM} = \tau\sqrt{3}$. The strain direction corresponded to the stress direction. The resulting plastic strain corresponded to that of a hand calculation;
4. A creep analysis was carried with $\sigma_{xx} = \sigma$ and $\sigma_{yy} = ½\ \sigma$. The Von Mises stress is then equal to $\sigma_{VM} = ½\sqrt{3}\ \sigma$. According to equation (20 a.) the resulting creep strains are $\varepsilon_{yy,pl} = 0$ and $\varepsilon_{xx,pl} = -\ \varepsilon_{zz,pl}$. The size and direction of the creep strain corresponded to that of a hand calculation;
5. A creep analysis was carried with a normal stress in one direction ($\sigma_{xx} = \sigma$). Later in the calculation, the size and the direction of the stress changed ($\sigma_{xx} = \sigma$, $\sigma_{yy} = \sigma$). Direction and size of the creep strain increments corresponded to that of a hand calculation.

5 Conclusions

The Dorn-Harmathy material model, which is able to describe the relationship between stresses and strains in case of fire exposed aluminium alloys, is implemented in the finite element program DIANA. The model was implemented both for uniaxial and multiaxial situations (or elements).

The routines are implemented in such a way that thermal expansion and residual stresses can be applied together with the constitutive model.
The tangent stiffness matrix was estimated numerically.

Convergence of the routines was sufficiently fast (for a model consisting of a single element: 3 iterations maximum for a time increment of 2 minutes in a transient state situation).

The routines were checked with uniaxial tests and hand calculations with creep, transient state and steady state conditions. The routines were also checked with hand calculations on stress relaxation and multiaxial stress conditions. The routines gave equal results in terms of stresses and strains as the hand calculations.

References

[1] Maljaars, J. Background report no. 4. Local buckling of slender aluminium sections exposed to fire Mechanical properties at elevated temperature
[2] Kraus, H. *Creep analysis*, Wiley, 1980, ISBN 0-471-06255-3

A Fortran code for uniaxial model

```
C$DIVA ARG1:READ ARG2:READ ARG3:READ ARG4:READ ARG5:READ
C$DIVA ARG6:READ ARG7:READ ARG8:READ ARG9:READ ARG10:READ
C$DIVA ARG11:READ ARG12:READ ARG13:READ ARG14:READ ARG15:READ
C$DIVA ARG16:READWRITE ARG17:READ ARG18:READWRITE ARG19:READ
C$DIVA ARG20:READWRITE ARG21:READWRITE
      SUBROUTINE USRMAT( EPS0, DEPS, NS, AGE0, DTIME, TEMP0, DTEMP,
     $                   ELEMEN, INTPT, COORD, SE, ITER, USRMOD, USRVAL,
     $                   NUV, USRSTA, NUS, USRIND, NUI, SIGMA, STIFF )
C.....................................Copyright (c) 2005 TNO DIANA B.V.
C...   NL/LB41/USRLIB
C...
C$DDOC PURPOSE:
C$DDOC   User-supplied subroutine for general nonlinear behaviour.
C$DDOC   Return updated stress and tangential stiffness matrix.
C$DDOC
C$DDOC ARGUMENTS:
C$DDOC   EPS0   D() In     - Strain vector at start of increment.
C$DDOC   DEPS   D() In     - Total strain increment.
C$DDOC   NS     I   In     - Number of stress components
C$DDOC   AGE0   D   In     - Age of element.
C$DDOC   DTIME  D   In     - Total time increment.
C$DDOC   TEMP0  D   In     - Temperature.
C$DDOC   DTEMP  D   In     - Total temperature increment.
C$DDOC   ELEMEN I   In     - Current element number.
C$DDOC   INTPT  I   In     - Current integration point number.
C$DDOC   COORD  D() In     - Coordinates of integration point.
C$DDOC   SE     D() In     - Elastcity matrix.
C$DDOC   ITER   I   In     - Current iteration number.
C$DDOC   USRMOD C   In     - User model name.
C$DDOC   USRVAL D() In     - User parameters.
C$DDOC   NUV    I   In     - Number of user parameters.
C$DDOC   USRSTA D() InOut  - User state variables at start of increment.
C$DDOC                       Should be updated at output.
C$DDOC   NUS    I   In     - Number of user state variables.
C$DDOC   USRIND I() InOut  - User indicators at start of increment.
C$DDOC                       Should be updated at output.
C$DDOC   NUI    I   In     - Number of user state indicators.
C$DDOC   SIGMA  D() InOut  - Total stress at start of increment.
C$DDOC                       Current stress at output.
C$DDOC   STIFF  D() InOut  - Previous tangent stiffness.
C$DDOC                       Current tangent stiffness at output.
C...
C...   DORN-HARMATHY MODEL IMPLEMENTATION
C...
C...   LITERATURE REFERENCE:
C...   Johan Maljaars - TNO
C...
C...   DEFINITION OF USER-DEFINED MATERIAL PARAMETERS:
C...
C...   USRVAL(1)   EMODA    : PAR A IN EQUATION FOR YOUNG'S MODULUS
C...   USRVAL(2)   EMODB    : PAR B IN EQUATION FOR YOUNG'S MODULUS
C...   USRVAL(3)   EMODC    : PAR C IN EQUATION FOR YOUNG'S MODULUS
C...   USRVAL(2)   ATHEA    : PAR A IN EQUATION FOR THERMAL EXPANSION
C...   USRVAL(3)   ATHEB    : PAR B IN EQUATION FOR THERMAL EXPANSION
```

```
C...   USRVAL(4)   PARQ     : PARAMETER Q - ACTIVATION ENERBY
C...   USRVAL(5)   PARA     : PARAMETER A - CONSTANT IN Z-FUNCTION
C...   USRVAL(6)   ALPHA    : PARAMETER ALPHA - MULTIPLICATION IN Z-FUNC
C...   USRVAL(7)   PARN     : PARAMETER N - EXPONENT IN Z-FUNCTION
C...   USRVAL(8)   PARD     : PARAMETER D - CONSTANT IN ET0-FUNCTION
C...   USRVLA(9)   PARM     : PARAMETER M - EXPONENT IN ET0-FUNCTION
C...   USRVAL(10)  ECRLIM   : LIMIT CREEP STRAIN
C...   USRVAL(11)  CR       : GAS CONSTANT
C...
C...   DEFINTION OF INTERNAL MATERIAL PARAMETERS
C...
C...   USRSTA(1)   EPSCR    : CREEP STRAIN
C.......................................................................
C
      INTEGER          MSTR
      PARAMETER      ( MSTR=1 )
C
      CHARACTER*6      USRMOD
      INTEGER          NS, NUV, NUS, NUI, ELEMEN, INTPT,
     $                 ITER
      DOUBLE PRECISION EPS0(NS), DEPS(NS), AGE0, DTIME, TEMP0, DTEMP,
     $                 COORD(3), SE(NS,NS), USRVAL(NUV), USRSTA(NUS),
     $                 SIGMA(NS), STIFF(NS,NS)
      INTEGER          USRIND(NUI)
C
      DOUBLE PRECISION EPSCR, EMODA, EMODB, EMODC,
     $                 YOUNG, ATHEA, ATHEB, PARQ, PARA,
     $                 ALPHA, PARN, PARD, PARM, ECRLIM, CR, TEMP,
     $                 TCEL, SIG0, EPSCR0, C20, C21,
     $                 DEPSCR, EPS, DDEPS, DEPSTH, ALFATH, EPSTH,
     $                 EPSTH0, DDEPSE, DECRE, SIGE, EPSE,
     $                 EVM, DEVM, ECRVM0, SIGVM0, VECRV0, C21E,
     $                 EVME, VET0XX, DEVME, ESTARP, ESTARN, ESTAR0,
     $                 ESTVM0
C
C...     CHECK DORN-HARMATHY MODEL DEFINTION
      IF ( USRMOD .NE. 'DORHAR'  ) CALL PRGERR( 'DORHAR', 1 )
      IF ( NS     .NE. MSTR      ) CALL PRGERR( 'DORHAR', 2 )
C
      EPSCR  = USRSTA(1)
      ESTARP = USRSTA(2)
      ESTARN = USRSTA(3)
      ESTAR0 = USRSTA(4)
      EPSTH  = USRSTA(5)
C
      EMODA  = USRVAL(1)
      EMODB  = USRVAL(2)
      EMODC  = USRVAL(3)
      ATHEA  = USRVAL(4)
      ATHEB  = USRVAL(5)
      PARQ   = USRVAL(6)
      PARA   = USRVAL(7)
      ALPHA  = USRVAL(8)
      PARN   = USRVAL(9)
      PARD   = USRVAL(10)
      PARM   = USRVAL(11)
      ECRLIM = USRVAL(12)
      CR     = USRVAL(13)
C
```

```
      EPS     = EPS0(1) + DEPS(1)
      DDEPS   = DEPS(1)
      TEMP    = TEMP0 + DTEMP
      TCEL    = TEMP - 273
      YOUNG   = EMODA*TCEL**2 + EMODB*TCEL + EMODC
      EPSCR0 = EPSCR
      EPSTH0 = EPSTH
      SIG0    = SIGMA(1)
      ALFATH = (TEMP0 + TEMP) / 2 * ATHEA + ATHEB
      DEPSTH = DTEMP * ALFATH
      EPSTH   = EPSTH0 + DEPSTH
      EVM     = SQRT(EPS**2)
      DEVM    = SQRT(DDEPS**2)
      ECRVM0 = SQRT(EPSCR0**2)
      SIGVM0 = SQRT(SIG0**2)
      ESTVM0 = SQRT(ESTAR0**2)
C
C...     CALCULATE CREEP STRAIN RATE AT BEGINNING OF INCREMENT
      CALL XXDOHA( ESTVM0, SIGVM0, TEMP0, PARQ, PARA, ALPHA,
     $             PARN, PARD, PARM, ECRLIM, CR, VECRV0, C20 )
C
C...     CALCULATE CREEP STRAIN RATE AT END OF INCREMENT
      CALL XXDOC2( DEVM, YOUNG, SIGVM0, ECRVM0, ESTVM0, EVM, EPSTH,
     $             TEMP, PARQ, PARA, ALPHA, PARN, PARD, PARM,
     $             ECRLIM, CR, DTIME, VECRV0, C21 )
C
      VET0XX     = C20 * 2 * SIG0
      SIGMA(1)   = ( EPS - EPSCR0 - 0.5*DTIME*VET0XX - EPSTH ) *
     $             YOUNG / ( 1 + DTIME * C21 * YOUNG )
      DEPSCR     = 0.5*DTIME*(VET0XX + C21*2*SIGMA(1))
C
      IF ( DEPSCR.LT.0.D0 ) THEN
         DDEPSE = DDEPS - 1.D-7
         EPSE   = EPS - 1.D-7
      ELSE
         DDEPSE = DDEPS + 1.D-7
         EPSE   = EPS + 1.D-7
      END IF
      DEVME   = SQRT(DDEPSE**2)
      EVME    = SQRT(EPSE**2)
C
C...     CALCULATE CREEP STRAIN AT END OF INCREMENT FOR 'WRONG' STRAIN
      CALL XXDOC2( DEVME, YOUNG, SIGVM0, ECRVM0, ESTVM0, EVME, EPSTH,
     $             TEMP, PARQ, PARA, ALPHA, PARN, PARD, PARM,
     $             ECRLIM, CR, DTIME, VECRV0, C21E )
C
      SIGE       = ( EPSE - EPSCR0 - 0.5*DTIME*VET0XX - EPSTH ) *
     $             YOUNG / ( 1 + DTIME * C21E * YOUNG )
      DECRE      = 0.5*DTIME*(VET0XX + C21E*2*SIGE)
      STIFF(1,1) = ( SIGE - SIGMA(1) ) / ( EPSE - EPS )
C
      CALL PRIVAL(DDEPS, 'DDEPS  ')
      CALL PRIVAL(DEPSCR, 'DEPSCR ')
      CALL PRIVAL(SIGMA(1), 'SIGMA  ')
      CALL PRIVAL((DDEPS-DDEPSE), 'DIFEPS ')
      CALL PRIVAL((DEPSCR-DECRE), 'DIFECR ')
      CALL PRIVAL((SIGMA(1)-SIGE), 'DIFSIG ')
      CALL PRIVAL(STIFF(1,1), 'STIFF  ')
      CALL PRIVAL(YOUNG, 'YOUNG  ')
```

```
C
      USRSTA(1)  = EPSCR0 + DEPSCR
      USRSTA(2)  = MAX ( 0.D0 , ( ESTARP + DEPSCR ))
      USRSTA(3)  = MAX ( 0.D0 , ( ESTARN - DEPSCR ))
      IF ( DEPSCR.LT.0.D0 ) THEN
         USRSTA(4) = MAX ( 0.D0 , ( ESTARN - DEPSCR ))
      ELSE
         USRSTA(4) = MAX ( 0.D0 , ( ESTARP + DEPSCR ))
      END IF
      USRSTA(5)  = EPSTH
C
      END
C
      SUBROUTINE XXDOC2( DDEPS, YOUNG, SIG0, EPSCR0, ESTAR0, EPS,
     $                   EPSTH, TEMP, PARQ, PARA, ALPHA, PARN,
     $                   PARD, PARM, ECRLIM, CR, DTIME, VEPCR0,
     $                   C21 )
C..................................Copyright (c) 2007 TNO
C...
C$DDOC PURPOSE:
C$DDOC   Calculate creep strain at end of increment
C$DDOC
C$DDOC ARGUMENTS:
C$DDOC   DDEPS  D   In     - Strain increment
C$DDOC   YOUNG  D   In     - Modulus of elasticity
C$DDOC   SIG0   D   In     - Stress at start of increment
C$DDOC   EPSCR0 D   In     - Creep strain at start of increment
C$DDOC   TEMP   D   In     - Temperature
C$DDOC   PARQ   D   In     - Parameter Q
C$DDOC   PARA   D   In     - Parameter A
C$DDOC   ALPHA  D   In     - Parameter alpha
C$DDOC   PARN   D   In     - Parameter n
C$DDOC   PARD   D   In     - Parameter d
C$DDOC   PARM   D   In     - Parameter m
C$DDOC   ECRLIM D   In     - Creep strain at which tertiary stage starts
C$DDOC   CR     D   In     - Gas Constant
C$DDOC   DTIME  D   In     - Total time increment
C$DDOC   VEPCR0 D   In     - Creep strain rate at start of increment
C$DDOC   DEPSCR D   Out    - Creep strain increment
C...
C.......................................................................
C
      DOUBLE PRECISION DDEPS, YOUNG, SIG0, EPSCR0, ESTAR0, EPS,
     $                 EPSTH, TEMP, PARQ, PARA, ALPHA, PARN, PARD,
     $                 PARM, ECRLIM, CR, DTIME, VEPCR0, C21
C
      INTEGER          IT, NTRIAL
      DOUBLE PRECISION TOL
      PARAMETER      ( NTRIAL=50, TOL=1.D-20 )
C
      DOUBLE PRECISION DECR1, SIG1, EPSCR1, DECRMI, DECRMA,
     $                 VEPCR1, VEPCAV, DEPSCR, Z, EPST0, POW, ESTAR1
C
      DECRMI = 0.D0
      DECRMA = DDEPS
C
      DO 100, IT = 1, NTRIAL
C
         DECR1 = ( DECRMI + DECRMA ) / 2.D0
```

```
         EPSCR1 = EPSCR0 + DECR1
         ESTAR1 = ESTAR0 + DECR1
         SIG1   = MAX ( 0.D0 , ( YOUNG * ( EPS - EPSCR1 - EPSTH )))
C
C...        CALCULATE CREEP STRAIN RATE AT END OF INCREMENT
         CALL XXDOHA( ESTAR1, SIG1, TEMP, PARQ, PARA, ALPHA,
     $                PARN, PARD, PARM, ECRLIM, CR, VEPCR1, C21 )
C
         VEPCAV = ( VEPCR0 + VEPCR1 ) / 2.D0
         DEPSCR = VEPCAV * DTIME
C
         IF ( ABS( DEPSCR - DECR1 ) .LT. TOL*ABS(DEPSCR) ) THEN
            GOTO 200
         ELSE IF ( DEPSCR .LT. DECR1 ) THEN
            DECRMA = DECR1
         ELSE IF ( DEPSCR .GT. DECR1 ) THEN
            DECRMI = DECR1
         END IF
C
  100 CONTINUE
  200 CONTINUE
C
      END
C
      SUBROUTINE XXDOHA( ESTAR, SIGMA, TEMP, PARQ, PARA, ALPHA,
     $                   PARN, PARD, PARM, ECRLIM, CR, VEPSCR, C2 )
C.....................................Copyright (c) 2007 TNO
C...
C$DDOC PURPOSE:
C$DDOC   Calculate creep strain rate
C$DDOC
C$DDOC ARGUMENTS:
C$DDOC   EPSCR  D   In     - Total Creep Strain
C$DDOC   SIGMA  D   In     - Stress
C$DDOC   TEMP   D   In     - Temperature
C$DDOC   PARQ   D   In     - Parameter Q
C$DDOC   PARA   D   In     - Parameter A
C$DDOC   ALPHA  D   In     - Parameter alpha
C$DDOC   PARN   D   In     - Parameter n
C$DDOC   PARD   D   In     - Parameter D
C$DDOC   PARM   D   In     - Parameter m
C$DDOC   ECRLIM D   In     - Limit Creep Strain
C$DDOC   CR     D   In     - Gas Constant
C$DDOC   VEPSCR D   Out    - Tangential creep strain
C...
C.......................................................................
C
      DOUBLE PRECISION ESTAR, SIGMA, TEMP, PARQ, PARA, ALPHA,
     $                 PARN, PARD, PARM, ECRLIM, CR, VEPSCR, C2
C
      DOUBLE PRECISION EPCR, Z, EPST0, POW
C
      EPCR = ESTAR
C...  IF ( ABS( EPCR ) .LT. 1.D-7 ) EPCR = 1.D-7
C
      Z     = PARA * ( SINH( ALPHA*SIGMA ) )**PARN
      EPST0 = PARD * SIGMA**PARM
      POW   = - PARQ / ( CR*TEMP )
C
```

```
      IF ( EPCR .LE. ( ECRLIM + EPST0 ) ) THEN
         VEPSCR = Z * EXP( POW )  /
     $            ( TANH( ( EPCR + 1.D-7 ) / ( EPST0 + 1.D-7 )))**2.
      ELSE
         VEPSCR = Z * EXP( POW ) * ( EPCR / ( ECRLIM + EPST0 ) ) /
     $            ( TANH( ( EPCR + 1.D-7 ) / ( EPST0 + 1.D-7 )))**2.
      END IF
      IF ( SIGMA .EQ. 0 ) THEN
         C2 = 0
      ELSE
         C2 = 0.5 * VEPSCR / SIGMA
      END IF
C
      END
```

B Fortran code for multiaxial model

```
C$DIVA ARG1:READ ARG2:READ ARG3:READ ARG4:READ ARG5:READ
C$DIVA ARG6:READ ARG7:READ ARG8:READ ARG9:READ ARG10:READ
C$DIVA ARG11:READ ARG12:READ ARG13:READ ARG14:READ ARG15:READ
C$DIVA ARG16:READWRITE ARG17:READ ARG18:READWRITE ARG19:READ
C$DIVA ARG20:READWRITE ARG21:READWRITE
      SUBROUTINE USRMNL( EPS0, DEPS, NS, AGE0, DTIME, TEMP0, DTEMP,
     $                   ELEMEN, INTPT, COORD, SE, ITER, USRMOD, USRVAL,
     $                   NUV, USRSTA, NUS, USRIND, NUI, SIGMA, STIFF )
C.....................................Copyright (c) 2005 TNO DIANA B.V.
C...   NL/LB41/USRLIB
C...
C$DDOC PURPOSE:
C$DDOC   User-supplied subroutine for general nonlinear behaviour.
C$DDOC   Return updated stress and tangential stiffness matrix.
C$DDOC
C$DDOC ARGUMENTS:
C$DDOC   EPS0   D() In     - Strain vector at start of increment.
C$DDOC   DEPS   D() In     - Total strain increment.
C$DDOC   NS     I   In     - Number of stress components
C$DDOC   AGE0   D   In     - Age of element.
C$DDOC   DTIME  D   In     - Total time increment.
C$DDOC   TEMP0  D   In     - Temperature.
C$DDOC   DTEMP  D   In     - Total temperature increment.
C$DDOC   ELEMEN I   In     - Current element number.
C$DDOC   INTPT  I   In     - Current integration point number.
C$DDOC   COORD  D() In     - Coordinates of integration point.
C$DDOC   SE     D() In     - Elastcity matrix.
C$DDOC   ITER   I   In     - Current iteration number.
C$DDOC   USRMOD C   In     - User model name.
C$DDOC   USRVAL D() In     - User parameters.
C$DDOC   NUV    I   In     - Number of user parameters.
C$DDOC   USRSTA D() InOut  - User state variables at start of increment.
C$DDOC                       Should be updated at output.
C$DDOC   NUS    I   In     - Number of user state variables.
C$DDOC   USRIND I() InOut  - User indicators at start of increment.
C$DDOC                       Should be updated at output.
C$DDOC   NUI    I   In     - Number of user state indicators.
C$DDOC   SIGMA  D() InOut  - Total stress at start of increment.
C$DDOC                       Current stress at output.
C$DDOC   STIFF  D() InOut  - Previous tangent stiffness.
C$DDOC                       Current tangent stiffness at output.
C...
C...   DORN-HARMATHY MODEL IMPLEMENTATION
C...
C...   LITERATURE REFERENCE:
C...   Johan Maljaars - TNO
C...
C...   DEFINITION OF USER-DEFINED MATERIAL PARAMETERS:
C...
C...   USRVAL(1)   EMODA    : PAR A IN EQUATION FOR YOUNG'S MODULUS
C...   USRVAL(2)   EMODB    : PAR B IN EQUATION FOR YOUNG'S MODULUS
C...   USRVAL(3)   EMODC    : PAR C IN EQUATION FOR YOUNG'S MODULUS
C...   USRVAL(2)   ATHEA    : PAR A IN EQUATION FOR THERMAL EXPANSION
C...   USRVAL(3)   ATHEB    : PAR B IN EQUATION FOR THERMAL EXPANSION
```

```
C...   USRVAL(4)   PARQ     : PARAMETER Q - ACTIVATION ENERBY
C...   USRVAL(5)   PARA     : PARAMETER A - CONSTANT IN Z-FUNCTION
C...   USRVAL(6)   ALPHA    : PARAMETER ALPHA - MULTIPLICATION IN Z-FUNC
C...   USRVAL(7)   PARN     : PARAMETER N - EXPONENT IN Z-FUNCTION
C...   USRVAL(8)   PARD     : PARAMETER D - CONSTANT IN ET0-FUNCTION
C...   USRVLA(9)   PARM     : PARAMETER M - EXPONENT IN ET0-FUNCTION
C...   USRVAL(10)  ECRLIM   : LIMIT CREEP STRAIN
C...   USRVAL(11)  CR       : GAS CONSTANT
C...
C...   DEFINTION OF INTERNAL MATERIAL PARAMETERS
C...
C...   USRSTA(1)   EPSCR    : CREEP STRAIN
C.....................................................................
C
      INTEGER           MSTR
      PARAMETER       ( MSTR=6 )
C
      CHARACTER*6       USRMOD
      INTEGER           NS, NUV, NUS, NUI, ELEMEN, INTPT, NDIR, NDIR2,
     $                  ITER, TEL
      DOUBLE PRECISION EPS0(NS), DEPS(NS), AGE0, DTIME, TEMP0, DTEMP,
     $                  COORD(3), SE(NS,NS), USRVAL(NUV), USRSTA(NUS),
     $                  SIGMA(NS), STIFF(NS,NS)
      INTEGER           USRIND(NUI)
C
      DOUBLE PRECISION TEMP, TCEL, EMODA, EMODB, EMODC, YOUNG, NU,
     $                  ATHEA, ATHEB, ALFATH, DEPSTH, EPSTH0, EPSTH,
     $                  PARQ, PARA, ALPHA, PARN, PARD, PARM, ECRLIM, CR,
     $                  C20, C21, SIG0(NS),SIGE(NS), EEL0(NS),
     $                  ECR0(NS), DECR(NS), EPS(NS), EPSER, VECR0(NS),
     $                  ESTARP(NS), ESTARN(NS), EST0(NS),
     $                  EVM, DEVM, ECRVM0, SIGVM0, VECRV0, ESTVM0,
     $                  SIGVM, DECRVM, TIME0, TIME, EPSPRE(NS)
C
C...      CHECK DORN-HARMATHY MODEL DEFINTION
C      IF ( USRMOD .NE. 'DORHAR'  ) CALL PRGERR( 'DORHAR', 1 )
      IF ( NS     .NE. MSTR      ) CALL PRGERR( 'DORHAR', 2 )
C
      EPSTH     = USRSTA(1)
      TIME0     = USRSTA(2)
      TEL = 2
      DO NDIR=1,6
         TEL = TEL + 1
         ECR0(NDIR) = USRSTA(TEL)
         TEL = TEL + 1
         ESTARP(NDIR) = USRSTA(TEL)
         TEL = TEL + 1
         ESTARN(NDIR) = USRSTA(TEL)
         TEL = TEL + 1
         EST0(NDIR)  = USRSTA(TEL)
         TEL = TEL + 1
         EPSPRE(NDIR) = USRSTA(TEL)
      END DO
C
      EMODA    = USRVAL(1)
      EMODB    = USRVAL(2)
      EMODC    = USRVAL(3)
      NU       = USRVAL(4)
      ATHEA    = USRVAL(5)
```

```
      ATHEB   = USRVAL(6)
      PARQ    = USRVAL(7)
      PARA    = USRVAL(8)
      ALPHA   = USRVAL(9)
      PARN    = USRVAL(10)
      PARD    = USRVAL(11)
      PARM    = USRVAL(12)
      ECRLIM  = USRVAL(13)
      CR      = USRVAL(14)
C
      TEMP   = TEMP0 + DTEMP
      TCEL   = TEMP - 273
      IF ( TCEL.LT.20.D0 ) THEN
         YOUNG = EMODA*20.D0**2.D0 + EMODB*20.D0 + EMODC
      ELSE
         YOUNG = EMODA*TCEL**2.D0 + EMODB*TCEL + EMODC
      END IF
      EPSTH0 = EPSTH
      ALFATH = (TEMP0 + TEMP) / 2.D0 * ATHEA + ATHEB
      DEPSTH = DTEMP * ALFATH
      EPSTH  = EPSTH0 + DEPSTH
C
      DO NDIR=1,6
         SIG0(NDIR) = SIGMA(NDIR)
      END DO
C
      IF ( TIME0 .EQ. 0.D0 ) THEN
         DO NDIR=1,6
            EPSPRE(NDIR) = SIG0(NDIR) / YOUNG
         END DO
      END IF
      TIME = TIME0 + DTIME
C
      DO NDIR=1,6
         EPS(NDIR) = EPS0(NDIR) + DEPS(NDIR) + EPSPRE(NDIR)
         EEL0(NDIR) = EPS0(NDIR) - ECR0 (NDIR)
      END DO
C
C...     CALCULATE VON MISES VALUES FOR STRESS AND STRAINS
      CALL XXEVM( NS, EPS, EVM )
      CALL XXEVM( NS, DEPS, DEVM )
      CALL XXEVM( NS, ECR0, ECRVM0 )
      CALL XXEVM( NS, EST0, ESTVM0 )
      CALL XXSVM( NS, SIG0, SIGVM0 )
C
C...     CALCULATE CREEP STRAIN RATE AT BEGINNING OF INCREMENT
      CALL XXDOHA( ESTVM0, SIGVM0, TEMP0, PARQ, PARA, ALPHA,
     $             PARN, PARD, PARM, ECRLIM, CR, C20 )
C
C...     CALCULATE STRAIN RATES AT BEGINNING OF INCREMENT
      VECR0(1) = C20 * ( 2.D0 * SIG0(1) - SIG0(2) - SIG0(3) )
      VECR0(2) = C20 * ( 2.D0 * SIG0(2) - SIG0(1) - SIG0(3) )
      VECR0(3) = C20 * ( 2.D0 * SIG0(3) - SIG0(2) - SIG0(1) )
      VECR0(4) = C20 * 6.D0 * SIG0(4)
      VECR0(5) = C20 * 6.D0 * SIG0(5)
      VECR0(6) = C20 * 6.D0 * SIG0(6)
C
C...     CALCULATE STRESSES AND CREEP STRAINS AT END OF INCREMENT
      CALL XXDOC2( NS, YOUNG, NU, EEL0, ECR0, EST0, VECR0, EPS, DEPS,
```

```
     $              EPSTH, TEMP, PARQ, PARA, ALPHA, PARN, PARD, PARM,
     $              ECRLIM, CR, DTIME, DECR, SIGMA,
     $              ESTARP, ESTARN )
C
c.      DO NDIR=1,3
C...        DETERMINE STRESSES FOR 'WRONG' STRAIN (EPSER)
c.         CALL XXEPSE( NS, NDIR, YOUNG, NU, DEPS, EPS, EEL0, ECR0,
VECR0,
c.     $                EST0, EPSTH, TEMP, PARQ, PARA, ALPHA, PARN,
c.     $                PARD, PARM, ECRLIM, CR, DTIME, SIGE, EPSER,
c.     $                ESTARP, ESTARN )
C
c.         DO NDIR2=1,3
c.            STIFF(NDIR2,NDIR) = ( SIGMA(NDIR2) - SIGE(NDIR2) ) /
c.     $                          ( EPS(NDIR) - EPSER )
c.         END DO
c.         DO NDIR2=4,6
c.            STIFF(NDIR2,NDIR) = 0.D0
c.         END DO
c.      END DO
C
c.      DO NDIR=4,6
C...        DETERMINE STRESSES FOR 'WRONG' STRAIN (EPSER)
c.         CALL XXEPSE( NS, NDIR, YOUNG, NU, DEPS, EPS, EEL0, ECR0,
VECR0,
c.     $                EST0, EPSTH, TEMP, PARQ, PARA, ALPHA, PARN,
c.     $                PARD, PARM, ECRLIM, CR, DTIME, SIGE, EPSER,
c.     $                ESTARP, ESTARN )
C
c.         DO NDIR2=1,6
c.            STIFF(NDIR2,NDIR) = 0.D0
c.         END DO
c.         STIFF(NDIR,NDIR) = ( SIGMA(NDIR) - SIGE(NDIR) ) /
c.     $                      ( EPS(NDIR) - EPSER )
c.      END DO
C
      DO NDIR=1,6
C...        DETERMINE STRESSES FOR 'WRONG' STRAIN (EPSER)
         CALL XXEPSE( NS, NDIR, YOUNG, NU, DEPS, EPS, EEL0, ECR0, VECR0,
     $                EST0, EPSTH, TEMP, PARQ, PARA, ALPHA, PARN,
     $                PARD, PARM, ECRLIM, CR, DTIME, SIGE, EPSER,
     $                ESTARP, ESTARN )
C
         DO NDIR2=1,6
            STIFF(NDIR2,NDIR) = ( SIGMA(NDIR2) - SIGE(NDIR2) ) /
     $                          ( EPS(NDIR) - EPSER )
         END DO
      END DO
C
c.      STIFF(1,1) = YOUNG*(1.D0 - NU)/((1.D0 + NU)*(1.D0 - 2.D0 * NU))
c.      STIFF(1,2) = YOUNG*NU / ((1.D0 + NU) * (1.D0 - 2.D0 * NU))
c.      STIFF(1,3) = YOUNG*NU / ((1.D0 + NU) * (1.D0 - 2.D0 * NU))
c.      STIFF(2,1) = YOUNG*NU / ((1.D0 + NU) * (1.D0 - 2.D0 * NU))
c.      STIFF(2,2) = YOUNG*(1.D0 - NU)/((1.D0 + NU)*(1.D0 - 2.D0 * NU))
c.      STIFF(2,3) = YOUNG*NU / ((1.D0 + NU) * (1.D0 - 2.D0 * NU))
c.      STIFF(3,1) = YOUNG*NU / ((1.D0 + NU) * (1.D0 - 2.D0 * NU))
c.      STIFF(3,2) = YOUNG*NU / ((1.D0 + NU) * (1.D0 - 2.D0 * NU))
c.      STIFF(3,3) = YOUNG*(1.D0 - NU)/((1.D0 + NU)*(1.D0 - 2.D0 * NU))
c.      STIFF(4,4) = YOUNG / (2.D0 + 2.D0*NU)
```

```
c.      STIFF(5,5) = YOUNG / (2.D0 + 2.D0*NU)
c.      STIFF(6,6) = YOUNG / (2.D0 + 2.D0*NU)
C
      USRSTA(1) = EPSTH
      USRSTA(2) = TIME
      TEL = 2
      DO NDIR=1,6
         TEL = TEL + 1
         USRSTA(TEL) = ECR0(NDIR) + DECR(NDIR)
         TEL = TEL + 1
         USRSTA(TEL) = MAX ( 0.D0 , ( ESTARP(NDIR) + DECR(NDIR) ))
         TEL = TEL + 1
         USRSTA(TEL) = MIN ( 0.D0 , ( ESTARN(NDIR) + DECR(NDIR) ))
         TEL = TEL + 1
         IF ( (DECR(NDIR)).LT.(0.D0) ) THEN
            USRSTA(TEL) = MIN ( 0.D0 , ( ESTARN(NDIR) + DECR(NDIR) ))
         ELSE
            USRSTA(TEL) = MAX ( 0.D0 , ( ESTARP(NDIR) + DECR(NDIR) ))
         END IF
         TEL = TEL + 1
         USRSTA(TEL) = EPSPRE(NDIR)
      END DO
C
      END
C
      SUBROUTINE XXEPSE( NS, NDIR, YOUNG, NU, DEPS, EPS, EEL0, ECR0,
     $                   VECR0, EST0, EPSTH, TEMP, PARQ, PARA, ALPHA,
     $                   PARN, PARD, PARM, ECRLIM, CR, DTIME, SIGE,
     $                   EPSER,
     $                   ESTARP, ESTARN )
C
C...........................................Copyright (c) 2007 TNO
C...
C$DDOC PURPOSE:  Calculate stresses in each direction
C$DDOC ARGUMENTS:
C$DDOC   NS()    D   In     - Number of stress components
C$DDOC   NDIR()  D   In     - Number of directions
C$DDOC   YOUNG   D   In     - Modulus of elasticity
C$DDOC   NU      D   In     - Poisson ratio
C$DDOC   DEPS()  D   In     - Von Mises strain increment
C$DDOC   EPS()   D   In     - Total strain at end of increment
C$DDOC   NDIR    D   In     - Direction for changing strain
C$DDOC   EEL0()  D   In     - Elastic strain at start of increment
C$DDOC   VECR0()D    In     - Creep strain rate at start of increment
C$DDOC   EST0()  D   In     - Modified creep strain at start of increment
C$DDOC   EPSTH   D   In     - Thermal strain
C$DDOC   TEMP    D   In     - Temperature
C$DDOC   PARQ    D   In     - Parameter Q
C$DDOC   PARA    D   In     - Parameter A
C$DDOC   ALPHA   D   In     - Parameter alpha
C$DDOC   PARN    D   In     - Parameter n
C$DDOC   PARD    D   In     - Parameter d
C$DDOC   PARM    D   In     - Parameter m
C$DDOC   ECRLIM D    In     - Creep strain at which tertiary stage starts
C$DDOC   CR      D   In     - Gas Constant
C$DDOC   DTIME   D   In     - Time increment
C$DDOC   SIGE()  D   Out    - Stress for 'wrong' strain increment
C$DDOC   EPSER   D   Out    - Total strain for wrong strain incremen
C$DDOC   ESTARP()D   In     - Modified creep strain at end of increment
```

```
C$DDOC   ESTARM()D   In     - Modified creep strain at end of increment
C...
C...........................................................................
C
      INTEGER          NDIR, NS
      DOUBLE PRECISION YOUNG, NU, DEPS(NS), EPS(NS), EEL0(NS), ECR0(NS),
     $                 VECR0(NS), EST0(NS), EPSTH, TEMP, PARQ, PARA,
     $                 ALPHA, PARN, PARD, PARM, ECRLIM, CR, DTIME,
     $                 SIGE(NS), EPSER, ESTARP(NS), ESTARN(NS)
C
      INTEGER          NDIR2
      DOUBLE PRECISION DEPSE(NS), EPSE(NS), DECRE(NS)
C
      DO NDIR2=1,6
         DEPSE(NDIR2) = DEPS(NDIR2)
         EPSE(NDIR2) = EPS(NDIR2)
      END DO
      IF ( DEPS(NDIR).LT.0.D0 ) THEN
         DEPSE(NDIR) = MIN(DEPS(NDIR)*(1.D0+1.D-2), DEPS(NDIR)-1.D-5)
      ELSE
         DEPSE(NDIR) = MAX(DEPS(NDIR)*(1.D0+1.D-2), DEPS(NDIR)+1.D-5)
      END IF
      EPSE(NDIR) = EPS(NDIR) - DEPS(NDIR) + DEPSE(NDIR)
      EPSER = EPSE(NDIR)
C
      CALL XXDOC2( NS, YOUNG, NU, EEL0, ECR0, EST0, VECR0, EPSE,
     $             DEPSE, EPSTH, TEMP, PARQ, PARA, ALPHA, PARN,
     $             PARD, PARM, ECRLIM, CR, DTIME, DECRE, SIGE,
     $             ESTARP, ESTARN )
C
      END
C
      SUBROUTINE XXDOC2( NS, YOUNG, NU, EEL0, ECR0, ESTAR0, VECR0, EPS,
     $                   DEPS, EPSTH, TEMP, PARQ, PARA, ALPHA, PARN,
     $                   PARD, PARM, ECRLIM, CR, DTIME, DECR, SIG,
     $                   ESTARP, ESTARN )
C
C.............................................Copyright (c) 2007 TNO
C...
C$DDOC PURPOSE:     Calculate creep strain rate at end of increment
C$DDOC ARGUMENTS:
C$DDOC   NS     D   In     - Number of stress components
C$DDOC   YOUNG  D   In     - Modulus of elasticity
C$DDOC   NU     D   In     - Poisson ratio
C$DDOC   EEL0() D   In     - Elastic strain at start of increment
C$DDOC   ECR0() D   In     - Creep strain at start of increment
C$DDOC   ESTAR0()D  In     - Modified creep strain at start of increment
C$DDOC   VECR0()D   In     - Strain rate at start of increment
C$DDOC   EPS()  D   In     - Total strain at end of increment
C$DDOC   DEPS() D   In     - Strain increment
C$DDOC   EPSTH  D   In     - Thermal strain
C$DDOC   TEMP   D   In     - Temperature
C$DDOC   PARQ   D   In     - Parameter Q
C$DDOC   PARA   D   In     - Parameter A
C$DDOC   ALPHA  D   In     - Parameter alpha
C$DDOC   PARN   D   In     - Parameter n
C$DDOC   PARD   D   In     - Parameter d
C$DDOC   PARM   D   In     - Parameter m
C$DDOC   ECRLIM D   In     - Creep strain at which tertiary stage starts
```

```
C$DDOC   CR      D   In     - Gas Constant
C$DDOC   DTIME   D   In     - Time increment
C$DDOC   DECR()  D   Out    - Creep strain increment
C$DDOC   SIG()   D   Out    - Stress at end of increment
C$DDOC   ESTARP()D   Out    - Modified creep strain
C$DDOC   ESTARN()D   Out    - Modified creep strain
C...
C.......................................................................
C
      INTEGER           NS
C
      DOUBLE PRECISION YOUNG, NU, EEL0(NS), ECR0(NS), ESTAR0(NS),
     $                 VECR0(NS), EPS(NS), DEPS(NS),  EPSTH, TEMP, PARQ,
     $                 PARA, ALPHA, PARN, PARD, PARM, ECRLIM, CR, DTIME,
     $                 DECR(NS), SIG(NS),
     $                 ESTARP(NS), ESTARN(NS)
C
      INTEGER           IT, NTRIAL, NDIR
      DOUBLE PRECISION TOL
      PARAMETER       ( NTRIAL=100, TOL=1.D-50 )
C
      DOUBLE PRECISION EMAXSQ(NS), SVMMAX, SIGVM1, ESTVM0, ESTVM1,
     $                 DECR1(NS), C21MAX, C21, C21MIN, C2, SIG1(NS),
     $                 ESTAR1(NS)
C
      DO NDIR=1,6
         EMAXSQ(NDIR) = MAX( (EEL0(NDIR))**2.D0,
     $                    ((EEL0(NDIR)+DEPS(NDIR))**2.D0 ) )
      END DO
      SVMMAX = YOUNG * DSQRT( EMAXSQ(1) + EMAXSQ(2) + EMAXSQ(3) +
     $         75.D-2 * ( EMAXSQ(4) + EMAXSQ(5) + EMAXSQ(6) ) )
C...     CALCULATE VON MISES VALUE FOR ESTAR0
      CALL XXEVM( NS, ESTAR0, ESTVM0 )
C...        CALCULATE MAXIMUM CREEP STRAIN RATE AT END OF INCREMENT
      CALL XXDOHA( ESTVM0, SVMMAX, TEMP, PARQ, PARA, ALPHA, PARN, PARD,
     $            PARM, ECRLIM, CR, C21MAX )
      C21MIN = 0
C
      DO 100, IT = 1, NTRIAL
C
         C21 = ( C21MAX + C21MIN ) / 2.D0
C...        CALCULATE STRESSES AT END OF INCREMENT
         CALL XXSIGM( NS, VECR0, EPS, EPSTH, ECR0, NU, C21, YOUNG,
     $                DTIME, SIG1 )
C...        CALCULATE VON MISES STRESS
         CALL XXSVM( NS, SIG1, SIGVM1 )
C
         DECR1(1) = 5.D-1 * DTIME * ( VECR0(1) +
     $              C21*(2.D0*SIG1(1)-SIG1(2)-SIG1(3)) )
         DECR1(2) = 5.D-1 * DTIME * ( VECR0(2) +
     $              C21*(2.D0*SIG1(2)-SIG1(1)-SIG1(3)) )
         DECR1(3) = 5.D-1 * DTIME * ( VECR0(3) +
     $              C21*(2.D0*SIG1(3)-SIG1(2)-SIG1(1)) )
         DECR1(4) = 5.D-1 * DTIME * ( VECR0(4) + 6.D0 * C21 * SIG1(4) )
         DECR1(5) = 5.D-1 * DTIME * ( VECR0(5) + 6.D0 * C21 * SIG1(5) )
         DECR1(6) = 5.D-1 * DTIME * ( VECR0(6) + 6.D0 * C21 * SIG1(6) )
C
         DO NDIR=1,6
            ESTAR1(NDIR) = ESTAR0(NDIR) + DECR1(NDIR)
```

```
c.             IF ( (DECR1(NDIR)).LT.(0.D0) ) THEN
c.                ESTAR1(NDIR) = MIN( 0.D0 , ( ESTARN(NDIR)+DECR1(NDIR)
))
c.             ELSE
c.                ESTAR1(NDIR) = MAX( 0.D0 , ( ESTARP(NDIR)+DECR1(NDIR)
))
c.             END IF
         END DO
C
C...         CALCULATE VON MISES VALUE FOR ESTAR
         CALL XXEVM( NS, ESTAR1, ESTVM1)
C
C...         CALCULATE CREEP STRAIN RATE AT END OF INCREMENT
         CALL XXDOHA( ESTVM1, SIGVM1, TEMP, PARQ, PARA, ALPHA,
     $                PARN, PARD, PARM, ECRLIM, CR, C2 )
C
C.       IF ( ABS( C2 - C21 ) .LT. MAX((TOL * C2), (TOL * C21)) ) THEN
         IF( ( C2 - C21 ) .EQ. TOL ) THEN
            GOTO 200
         ELSE IF ( C2 .LT. C21 ) THEN
            C21MAX = C21
         ELSE IF ( C2 .GT. C21 ) THEN
            C21MIN = C21
         END IF
C
  100 CONTINUE
  200 CONTINUE
C
      DO NDIR=1,6
         SIG(NDIR) = SIG1(NDIR)
         DECR(NDIR) = DECR1(NDIR)
      END DO
C
      END
C
      SUBROUTINE XXDOHA( ESTAR, SIGMA, TEMP, PARQ, PARA, ALPHA,
     $                  PARN, PARD, PARM, ECRLIM, CR, C2 )
C..................................................Copyright (c) 2007 TNO
C...
C$DDOC PURPOSE:     Calculate creep strain rate
C$DDOC ARGUMENTS:
C$DDOC   EPSCR  D   In     - Total Von Mises creep strain
C$DDOC   SIGMA  D   In     - Von Mises stress
C$DDOC   TEMP   D   In     - Temperature
C$DDOC   PARQ   D   In     - Parameter Q
C$DDOC   PARA   D   In     - Parameter A
C$DDOC   ALPHA  D   In     - Parameter alpha
C$DDOC   PARN   D   In     - Parameter n
C$DDOC   PARD   D   In     - Parameter D
C$DDOC   PARM   D   In     - Parameter m
C$DDOC   ECRLIM D   In     - Limit creep strain
C$DDOC   CR     D   In     - Gas constant
C$DDOC   VEPSCR D   Out    - Tangential Von Mises creep strain
C$DDOC   C2     D   Out    - Ratio between Von Mises creep strain rate
C$DDOC                       and Von Mises stress
C...
C.......................................................................
C
      DOUBLE PRECISION ESTAR, SIGMA, TEMP, PARQ, PARA, ALPHA,
```

```
     $                  PARN, PARD, PARM, ECRLIM, CR, C2
C
      DOUBLE PRECISION EPCR, Z, EPST0, POW, VEPSCR
C
      EPCR = ESTAR
C...  IF ( ABS( EPCR ) .LT. 1.D-7 ) EPCR = 1.D-7
C
      Z     = PARA * ( SINH( ALPHA*SIGMA ) )**PARN
      EPST0 = PARD * SIGMA**PARM
      POW   = - PARQ / ( CR*TEMP )
C
      IF ( EPCR .LE. ( ECRLIM + EPST0 ) ) THEN
         VEPSCR = Z * EXP( POW ) /
     $            ( TANH( ( EPCR + 1.D-7 ) / ( EPST0 + 1.D-7 )))**2.
      ELSE
         VEPSCR = Z * EXP( POW ) * ( EPCR / ( ECRLIM + EPST0 ) ) /
     $            ( TANH( ( EPCR + 1.D-7 ) / ( EPST0 + 1.D-7 )))**2.
      END IF
C
      IF ( SIGMA .EQ. 0 ) THEN
         C2 = 0
      ELSE
         C2 = 5.D-1 * VEPSCR / SIGMA
      END IF
C
      END
C
      SUBROUTINE XXSIGM( NS, VECR0, EPS, EPSTH, ECR0, NU, C21,
     $                    YOUNG, DTIME, SIG )
C
C.................................................Copyright (c) 2007 TNO
C...
C$DDOC PURPOSE:  Calculate stress in each direction
C$DDOC ARGUMENTS:
C$DDOC   NS()    D   In     - Number of stress components
C$DDOC   VECR0() D   In     - Creep strain rate at start of increment
C$DDOC   EPS()   D   In     - Total strain at end of increment
C$DDOC   EPSTH   D   In     - Thermal strain
C$DDOC   ECR0()  D   In     - Creep strain at start of increment
C$DDOC   NU      D   In     - Poisson ratio
C$DDOC   C21     D   In     - Ratio between Von Mises creep strain rate
C$DDOC                        and Von Mises stress at end of increment
C$DDOC   YOUNG   D   In     - Modulus of elasticity
C$DDOC   DTIME   D   In     - Time increment
C$DDOC   SIG()   D   Out    - Stress
C...
C......................................................................
C
      INTEGER          NS
      DOUBLE PRECISION VECR0(NS), EPS(NS), EPSTH, ECR0(NS), NU, C21,
$                  YOUNG, DTIME, SIG(NS)
C
      DOUBLE PRECISION CA, CB, CC, CD
C
      CA = ( -NU - 5.D-1*DTIME*C21*YOUNG ) / ( 1.D0 + DTIME*C21*YOUNG )
      CB = ( EPS(1) - ECR0(1) - 5.D-1 * DTIME * VECR0(1) - EPSTH)
     $     * YOUNG / ( 1.D0 + DTIME * YOUNG * C21)
      CC = ( EPS(2) - ECR0(2) - 5.D-1 * DTIME * VECR0(2) - EPSTH)
     $     * YOUNG / ( 1.D0 + DTIME * YOUNG * C21)
```

```
      CD = ( EPS(3) - ECR0(3) - 5.D-1 * DTIME * VECR0(3) - EPSTH)
     $      * YOUNG / ( 1.D0 + DTIME * YOUNG * C21)
C
      SIG(1) = (( 1.D0 - (CA)**2 ) * CB - ( CA - (CA)**2 )* ( CC + CD ))
     $          / ( 1.D0 - 3.D0 * (CA)**2 + 2.D0 * (CA)**3 )
      SIG(2) = (( 1.D0 - (CA)**2 ) * CC - ( CA - (CA)**2 )* ( CB + CD ))
     $          / ( 1.D0 - 3.D0 * (CA)**2 + 2.D0 * (CA)**3 )
      SIG(3) = (( 1.D0 - (CA)**2 ) * CD - ( CA - (CA)**2 )* ( CB + CC ))
     $          / ( 1.D0 - 3.D0 * (CA)**2 + 2.D0 * (CA)**3 )
      SIG(4) = YOUNG * (EPS(4) - ECR0(4) - 5.D-1 * DTIME * VECR0(4)) /
     $          ( 2.D0 + 2.D0 * NU + 3.D0 * YOUNG * C21 * DTIME )
      SIG(5) = YOUNG * (EPS(5) - ECR0(5) - 5.D-1 * DTIME * VECR0(5)) /
     $          ( 2.D0 + 2.D0 * NU + 3.D0 * YOUNG * C21 * DTIME )
      SIG(6) = YOUNG * (EPS(6) - ECR0(6) - 5.D-1 * DTIME * VECR0(6) ) /
     $          ( 2.D0 + 2.D0 * NU + 3.D0 * YOUNG * C21 * DTIME )
C
      END
C
      SUBROUTINE XXEVM( NS, EPS, EVM )
C..........................................Copyright (c) 2007 TNO
C...
C$DDOC PURPOSE:     Calculate Von Mises strain
C$DDOC
C$DDOC ARGUMENTS:
C$DDOC   NS()   D   In     - Number of stress components
C$DDOC   EPS()  D   In     - Strain per direction
C$DDOC   EVM    D   Out    - Von Mises strain
C...
C.............................................................
C
      INTEGER          NS
      DOUBLE PRECISION EPS(NS), EVM
      DOUBLE PRECISION EVM2
C
      EVM2 = (EPS(1))**2.D0 + (EPS(2))**2.D0 + (EPS(3))**2.D0 -
     $       EPS(1)*EPS(2) - EPS(2)*EPS(3) - EPS(1)*EPS(3) +
     $       75.D-2 * ( (EPS(4))**2.D0 ) + 75.D-2 * ( (EPS(5))**2.D0 ) +
     $       75.D-2 * ( (EPS(6))**2.D0 )
      EVM  = 2.D0 / 3.D0 * DSQRT(EVM2)
C
      END
C
      SUBROUTINE XXSVM( NS, SIG, SVM )
C..........................................Copyright (c) 2007 TNO
C...
C$DDOC PURPOSE:     Calculate Von Mises stress
C$DDOC
C$DDOC ARGUMENTS:
C$DDOC   NS()   D   In     - Number of stress components
C$DDOC   SIG()  D   In     - Stress per direction
C$DDOC   SVM    D   Out    - Von Mises stress
C...
C.............................................................
C
      INTEGER          NS
      DOUBLE PRECISION SIG(NS), SVM
      DOUBLE PRECISION SVM2
C
      SVM2 = (SIG(1))**2.D0 + (SIG(2))**2.D0 + (SIG(3))**2.D0 -
```

```
     $       SIG(1)*SIG(2) - SIG(2)*SIG(3) - SIG(1)*SIG(3) +
     $       3.D0*( (SIG(4))**2.D0 ) + 3.D0*( (SIG(5))**2.D0 ) +
     $       3.D0*( (SIG(6))**2.D0 )
      SVM  = DSQRT(SVM2)
C
      END
C
```