FORSCHUNGSZENTRUM ROSSENDORF FZR

WISSENSCHAFTLICH-TECHNISCHE BERICHTE
FZR-296
Juli 2000
ISSN 1437-322X

Eberhard Altstadt and Thomas Moessner

Extension of the ANSYS® creep and damage simulation capabilities

Herausgeber:
FORSCHUNGSZENTRUM ROSSENDORF
Postfach 51 01 19
D-01314 Dresden
Telefon +49 351 26 00
Telefax +49 351 2 69 04 61
http://www.fz-rossendorf.de/

Als Manuskript gedruckt

FORSCHUNGSZENTRUM ROSSENDORF
WISSENSCHAFTLICH-TECHNISCHE BERICHTE
FZR-296
Juli 2000

Eberhard Altstadt and Thomas Moessner

Extension of the ANSYS® creep and damage simulation capabilities

Abstract

The user programmable features (UPF) of the finite element code ANSYS® are used to generate a customized ANSYS-executable including a more general creep behaviour of materials and a damage module. The numerical approach for the creep behaviour is not restricted to a single creep law (e.g. strain hardening model) with parameters evaluated from a limited stress and temperature range. Instead of this strain rate - strain relations can be read from external creep data files for different temperature and stress levels.

The damage module accumulates a damage measure based on the creep strain increment and plastic strain increment of the load step and the current fracture strains for creep and plasticity (depending on temperature and stress level). If the damage measure of an element exceeds a critical value this element is deactivated.

Examples are given for illustration and verification of the new program modules.

Contents

Nomenclature

Latin

$c_1 \ldots c_4$	coefficients of the time hardening creep equation
$d_1 \ldots d_4$	coefficients of the strain hardening creep equation
D	damage parameter
ΔD	damage increment
K	Kelvin
q	temperature exponent for non-linear interpolation
Pa	Pascal
r	stress exponent for non-linear interpolation; radius
R_v	triaxiality function
s	second
t	time
Δt	time increment
T	temperature
UPF	user programmable features
$w_1 \ldots w_8$	weighting factors for interpolation of the creep strain increment
x,y	coordinates

Greek

ε	strain
$\dot{\varepsilon}$	strain rate
ε^{cr}_{frac}	creep fracture strain
ε^{cr}_{eqv}	effective (equivalent) creep strain
$\Delta\varepsilon^{cr}_{eqv}$	effective creep strain increment
ε^{pl}_{frac}	plastic fracture strain
ε^{pl}_{eqv}	effective (equivalent) plastic strain
$\Delta\varepsilon^{cr}_{eqv}$	effective plastic strain increment
ν	Poisson's number
σ	mechanical stress
σ_{eqv}	von-Mises equivalent stress
σ_H	hydrostatic stress

Indices

cr	creep
eqv	equivalent
frac	fracture
H	high, hydrostatic, hole
L	low
max	maximum
min	minimum
pl	plastic
P	primary
ref	reference
S	secondary
T	tertiary
tan	tangential

1. Introduction

The creep behaviour of materials is usually described by analytical formulas (creep laws) with a number of free coefficients, e.g.:

$$\dot{\varepsilon} = d_1 \cdot \left[\frac{\sigma}{\sigma_{ref}} \right]^{d_2} \cdot \varepsilon^{d_3} \cdot \exp\left[-\frac{d_4}{T} \right] \tag{1}$$

The coefficients (c_1... c_4 in eq. 1) are used to adapt the creep laws to creep test results performed at constant load and temperature. However, in practice it is often difficult to achieve a satisfying adjustment for a wide range of temperatures and stresses with only one set of coefficients. Instead it appears that the coefficients itself are dependent on the temperature or on the stress level. Therefore a supplementary tool is developed which allows to describe the creep behaviour of a material for different stress and temperature levels independently. This is especially useful if strong stress and/or temperature gradients are present and if both primary and secondary creep are to be considered in the analysis. Additionally it is possible to calculate the creep damage and deactivate elements whose accumulated damage is greater or equal to one.

The user programmable features (UPF) available on the ANSYS® distribution media were used to develop the tools. The Compac® Visual Fortran Compiler (Rev. 6.1) was used for programming and for generating the customized ANSYS-executable on a Windows/NT® platform.

2. Input of the creep data of the materials

The creep behaviour of a material can be described by the strain hardening representation

$$\dot{\varepsilon}^{cr} = f(\varepsilon^{cr};\sigma;T) \tag{2}$$

The time hardening representation or the work hardening representation can in general be transformed into eq (2). The relation eq (2) is transferred into the ANSYS database by means of a number of discrete pairs of the form

$$\begin{bmatrix} \varepsilon_{(1)} & \dot{\varepsilon}_{(1)} \\ \vdots & \vdots \\ \varepsilon_{(n)} & \dot{\varepsilon}_{(n)} \end{bmatrix}_{T=const;\sigma=const} \tag{3}$$

Several of such sets for different temperature and stress levels can be combined. The complete creep data base then is as follows:

$$\begin{array}{ccc|c|ccc}
 & T_1 & & \cdots & & T_K & \\
\hline
\sigma_{1;1} & \cdots & \sigma_{1;M1} & \cdots & \sigma_{K;1} & \cdots & \sigma_{K;MK} \\
\hline
\varepsilon^{frac}_{1;1} & \cdots & \varepsilon^{frac}_{1;M1} & \cdots & \varepsilon^{frac}_{K;1} & \cdots & \varepsilon^{frac}_{K;MK} \\
\hline
N_{1;1} & \cdots & N_{M1} & \cdots & N_{K;1} & \cdots & N_{K;MK} \\
\hline
\left(\varepsilon_{1;1;1};\dot{\varepsilon}_{1;1;1}\right) & \cdots & \left(\varepsilon_{1;M1;1};\dot{\varepsilon}_{1;M1;1}\right) & \cdots & \left(\varepsilon_{K;1;1};\dot{\varepsilon}_{K;1;1}\right) & \cdots & \left(\varepsilon_{K;MK;1};\dot{\varepsilon}_{K;MK;1}\right) \\
\vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\left(\varepsilon_{1;1;N};\dot{\varepsilon}_{1;1;N}\right) & \cdots & \left(\varepsilon_{1;M1;N};\dot{\varepsilon}_{1;M1;N}\right) & \cdots & \left(\varepsilon_{K;1;N};\dot{\varepsilon}_{K;1;N}\right) & \cdots & \left(\varepsilon_{K;MK;N};\dot{\varepsilon}_{K;MK;N}\right)
\end{array} \tag{4}$$

The first index refers to the temperature, the second index to the stress and the third to the strain. K is the number of temperature levels, Mk the number of stress levels within the k-th temperature level and $N_{k;m}$ the number of strain rate-strain pairs for the m-th stress level within the k-th temperature level.

The UPF routine user01 /1/ is used to realise the creep data input into ANSYS. The data must be provided by the user as a set of ASCII files (for each temperature-stress level one file). The structure of a creep data file is demonstrated in the following example:

```
8.73150E+02     !! Temperature K
3.00000E+05     !! Stress kPa
3.50000E-01     !! creep fracture strain
1.00000E+02     !! number of strain sets in this file
0.00000E+00     6.00000E-05     !! set1: eps1 deps1
4.97257E-04     7.55093E-05     !! set2: eps2 deps2
1.16640E-03     8.85600E-05     !! set3: eps3 deps3
1.92061E-03     9.72162E-05     !! set4: eps4 deps4
2.73599E-03     1.03866E-04     !! set5: eps5 deps5
.....................................................
.....................................................
6.16071E-01     4.83579E-03     !! set98: eps98 deps98
6.56547E-01     5.16142E-03     !! set99: eps99 deps99
6.99738E-01     5.50632E-03     !! set100: eps100 deps100
```

Table 1: creep data file example

The creep data files must be named according to the pattern: *basename*.c?? where ?? stands for the sequent file number (01 .... nfiles). To read the creep data files use the following ANSYS commands /3,4/:

```
!! set command name associated with UPF user01
     /ucmd,usr1,1
!! specify the names and the number of the creep data files
     usr1,filename,nfiles,log_key,intp_key
!! instruct ANSYS to use the user defined creep law
     /prep7
     tb,creep,mat
     tbdata,1,1,,,,,100
```

Table 2: ANSYS input example for using the user defined creep data base

If *log_key*=1, a control output file <jobnam>.crlg is written. If the argument *intp_key*=1, the non-linear interpolation scheme is activated (section 3).

For the generation of the creep data files the supporting program CRPGEN is available (command reference in Annex 2).

3. Calculation of the creep strain increment

To realize the calculation of the creep strain increment according to the non-standard creep law, the UPF usercr.f was modified and linked to the customized ANSYS executable /1/. In this routine the scalar creep strain increment $\Delta\varepsilon^{cr} = \dot{\varepsilon}^{cr} \cdot \Delta t$ is determined from the creep data (input see section 2) by multi-linear interpolation:

$$\Delta\varepsilon^{cr} = [w_1 \cdot \dot{\varepsilon}_{L;L;L} + w_2 \cdot \dot{\varepsilon}_{L;L;H} + w_3 \cdot \dot{\varepsilon}_{L;H;L} + w_4 \cdot \dot{\varepsilon}_{L;H;H} + w_5 \cdot \dot{\varepsilon}_{H;L;L} + w_6 \cdot \dot{\varepsilon}_{H;L;H} + w_7 \cdot \dot{\varepsilon}_{H;H;L} + w_8 \cdot \dot{\varepsilon}_{H;H;H}] \cdot \Delta t \quad (5)$$

3.1 Linear interpolation

Assuming that the strain rate depends linearly on stress and temperature between two data base points, the weighting factors in eq. (5) are:

$$
\begin{aligned}
w_1 &= \frac{(T_H - T)\cdot(\sigma_{L;H} - \sigma)\cdot(\varepsilon_{L;L;H} - \varepsilon)}{(T_H - T_L)\cdot(\sigma_{L;H} - \sigma_{L;L})\cdot(\varepsilon_{L;L;H} - \varepsilon_{L;L;L})} \\
w_2 &= \frac{(T_H - T)\cdot(\sigma_{L;H} - \sigma)\cdot(\varepsilon - \varepsilon_{L;L;L})}{(T_H - T_L)\cdot(\sigma_{L;H} - \sigma_{L;L})\cdot(\varepsilon_{L;L;H} - \varepsilon_{L;L;L})} \\
w_3 &= \frac{(T_H - T)\cdot(\sigma - \sigma_{L;L})\cdot(\varepsilon_{L;H;H} - \varepsilon)}{(T_H - T_L)\cdot(\sigma_{L;H} - \sigma_{L;L})\cdot(\varepsilon_{L;H;H} - \varepsilon_{L;H;L})} \\
w_4 &= \frac{(T_H - T)\cdot(\sigma - \sigma_{L;L})\cdot(\varepsilon - \varepsilon_{L;H;L})}{(T_H - T_L)\cdot(\sigma_{L;H} - \sigma_{L;L})\cdot(\varepsilon_{L;H;H} - \varepsilon_{L;H;L})} \\
w_5 &= \frac{(T - T_L)\cdot(\sigma_{H;H} - \sigma)\cdot(\varepsilon_{H;L;H} - \varepsilon)}{(T_H - T_L)\cdot(\sigma_{H;H} - \sigma_{H;L})\cdot(\varepsilon_{H;L;H} - \varepsilon_{H;L;L})} \\
w_6 &= \frac{(T - T_L)\cdot(\sigma_{H;H} - \sigma)\cdot(\varepsilon - \varepsilon_{H;L;L})}{(T_H - T_L)\cdot(\sigma_{H;H} - \sigma_{H;L})\cdot(\varepsilon_{H;L;H} - \varepsilon_{H;L;L})} \\
w_7 &= \frac{(T - T_L)\cdot(\sigma - \sigma_{H;L})\cdot(\varepsilon_{H;H;H} - \varepsilon)}{(T_H - T_L)\cdot(\sigma_{H;H} - \sigma_{H;L})\cdot(\varepsilon_{H;H;H} - \varepsilon_{H;H;L})} \\
w_8 &= \frac{(T - T_L)\cdot(\sigma - \sigma_{H;L})\cdot(\varepsilon - \varepsilon_{H;H;L})}{(T_H - T_L)\cdot(\sigma_{H;H} - \sigma_{H;L})\cdot(\varepsilon_{H;H;H} - \varepsilon_{H;H;L})} \\
&\left(\sum_{i=1}^{8} w_i = 1\right)
\end{aligned}
\qquad \textbf{(6)}
$$

The quantities without index ε, σ, T are the actual values of the current element integration point. The indexed quantities are the values from the creep data base eq. (3). They form the smallest intervals which the actual quantities are enclosed in. The meaning of the indices is L: low bound (largest data base value which is smaller than the actual integration point value) and H: high bound (smallest data base value which is greater than the actual integration point value). The first index refers to the temperature, the second index to the stress and the third to the strain.

$$
\begin{aligned}
&T_L \le T \le T_H \\
&\{\sigma_{L;L}, \sigma_{H;L}\} \le \sigma \le \{\sigma_{L;H}, \sigma_{H;H}\} \\
&\{\varepsilon_{L;L;L}, \varepsilon_{L;H;L}, \varepsilon_{H;L;L}, \varepsilon_{H;H;L}\} \le \varepsilon \le \{\varepsilon_{L;L;H}, \varepsilon_{L;H;H}, \varepsilon_{H;L;H}, \varepsilon_{H;H;H}\}
\end{aligned}
\qquad \textbf{(7)}
$$

All stresses and strains in eq. (4-6) are equivalent values. The components of the creep strain tensor increment, $\Delta\varepsilon_{kl}^{cr}$, are calculated according to the Prandtl-Reuss flow rule /2,6/.

The creep data base eq. (3) has to be provided in such a way that the actual temperature and the equivalent stress of the elements do not exceed the maximum values of the creep data base. If the actual temperature or stress values are smaller than the smallest values of eq. (4), the creep strain increment is zero for this step.

3.2 Non-linear interpolation

If the temperature steps and/or the stress steps in the creep data base are not sufficiently small, the creep strain increment might be overestimated, since the strain rate in general depends over-proportionally on the stress and on the temperature respectively. That's why a non-linear calculation of the weighting coefficients is useful.

Assuming that the strain rate depends exponentially on the stress, i.e.

$$\dot{\varepsilon} = A \cdot \sigma^{r} \tag{8}$$

the according interpolation between two points is:

$$\dot{\varepsilon} = \frac{\sigma_2^r - \sigma^r}{\sigma_2^r - \sigma_1^r} \cdot \dot{\varepsilon}_1 + \frac{\sigma^r - \sigma_1^r}{\sigma_2^r - \sigma_1^r} \cdot \dot{\varepsilon}_2 \tag{9}$$

the exponent r can be estimated from

$$r = \frac{\ln(\dot{\varepsilon}_2) - \ln(\dot{\varepsilon}_1)}{\ln(\sigma_2) - \ln(\sigma_1)} \tag{10}$$

Similarly, if it is assumed that the strain rate dependency on the temperature can be described by

$$\dot{\varepsilon} = K \cdot e^{-q/T} \tag{11}$$

the appropriate interpolation between two points is

$$\dot{\varepsilon} = \frac{e^{-q/T_2} - e^{-q/T}}{e^{-q/T_2} - e^{-q/T_1}} \cdot \dot{\varepsilon}_1 + \frac{e^{-q/T} - e^{-q/T_1}}{e^{-q/T_2} - e^{-q/T_1}} \cdot \dot{\varepsilon}_2 \tag{12}$$

The parameter q can be estimated from

$$q = \frac{\ln(\dot{\varepsilon}_2) - \ln(\dot{\varepsilon}_1)}{\frac{1}{T_1} - \frac{1}{T_2}} \tag{13}$$

Of course, the parameter q can change with the stress level and the accumulated creep strain as

the parameter r may be dependent on temperature and strain.
Applying the above equations to the calculation of the weighting coefficients, one obtains:

$$
\begin{aligned}
w_1 &= \frac{(e^{-q_L/T_H} - e^{-q_L/T})\cdot(\sigma^{r_L}_{L;H} - \sigma^{r_L})\cdot(\varepsilon_{L;L;H} - \varepsilon)}{(e^{-q_L/T_H} - e^{-q_L/T_L})\cdot(\sigma^{r_L}_{L;H} - \sigma^{r_L}_{L;L})\cdot(\varepsilon_{L;L;H} - \varepsilon_{L;L;L})} \\
w_2 &= \frac{(e^{-q_L/T_H} - e^{-q_L/T})\cdot(\sigma^{r_L}_{L;H} - \sigma^{r_L})\cdot(\varepsilon - \varepsilon_{L;L;L})}{(e^{-q_L/T_H} - e^{-q_L/T_L})\cdot(\sigma^{r_L}_{L;H} - \sigma^{r_L}_{L;L})\cdot(\varepsilon_{L;L;H} - \varepsilon_{L;L;L})} \\
w_3 &= \frac{(e^{-q_H/T_H} - e^{-q_H/T})\cdot(\sigma^{r_L} - \sigma^{r_L}_{L;L})\cdot(\varepsilon_{L;H;H} - \varepsilon)}{(e^{-q_H/T_H} - e^{-q_H/T_L})\cdot(\sigma^{r_L}_{L;H} - \sigma^{r_L}_{L;L})\cdot(\varepsilon_{L;H;H} - \varepsilon_{L;H;L})} \\
w_4 &= \frac{(e^{-q_H/T_H} - e^{-q_H/T})\cdot(\sigma^{r_L} - \sigma^{r_L}_{L;L})\cdot(\varepsilon - \varepsilon_{L;H;L})}{(e^{-q_H/T_H} - e^{-q_H/T_L})\cdot(\sigma^{r_L}_{L;H} - \sigma^{r_L}_{L;L})\cdot(\varepsilon_{L;H;H} - \varepsilon_{L;H;L})} \\
w_5 &= \frac{(e^{-q_L/T} - e^{-q_L/T_L})\cdot(\sigma^{r_H}_{H;H} - \sigma^{r_H})\cdot(\varepsilon_{H;L;H} - \varepsilon)}{(e^{-q_L/T_H} - e^{-q_L/T_L})\cdot(\sigma^{r_H}_{H;H} - \sigma^{r_H}_{H;L})\cdot(\varepsilon_{H;L;H} - \varepsilon_{H;L;L})} \\
w_6 &= \frac{(e^{-q_L/T} - e^{-q_L/T_L})\cdot(\sigma^{r_H}_{H;H} - \sigma^{r_H})\cdot(\varepsilon - \varepsilon_{H;L;L})}{(e^{-q_L/T_H} - e^{-q_L/T_L})\cdot(\sigma^{r_H}_{H;H} - \sigma^{r_H}_{H;L})\cdot(\varepsilon_{H;L;H} - \varepsilon_{H;L;L})} \\
w_7 &= \frac{(e^{-q_H/T} - e^{-q_H/T_L})\cdot(\sigma^{r_H} - \sigma^{r_H}_{H;L})\cdot(\varepsilon_{H;H;H} - \varepsilon)}{(e^{-q_H/T_H} - e^{-q_H/T_L})\cdot(\sigma^{r_H}_{H;H} - \sigma^{r_H}_{H;L})\cdot(\varepsilon_{H;H;H} - \varepsilon_{H;H;L})} \\
w_8 &= \frac{(e^{-q_H/T} - e^{-q_H/T_L})\cdot(\sigma^{r_H} - \sigma^{r_H}_{H;L})\cdot(\varepsilon - \varepsilon_{H;H;L})}{(e^{-q_H/T_H} - e^{-q_H/T_L})\cdot(\sigma^{r_H}_{H;H} - \sigma^{r_H}_{H;L})\cdot(\varepsilon_{H;H;H} - \varepsilon_{H;H;L})}
\end{aligned}
\tag{14}
$$

with the parameters

$$
\begin{aligned}
q_L &= \frac{\ln(\dot\varepsilon_{H;L;L}) - \ln(\dot\varepsilon_{L;L;L})}{\frac{1}{T_L} - \frac{1}{T_H}}; \quad q_H = \frac{\ln(\dot\varepsilon_{H;H;L}) - \ln(\dot\varepsilon_{L;H;L})}{\frac{1}{T_L} - \frac{1}{T_H}} \\
r_L &= \frac{\ln(\dot\varepsilon_{L;H;L}) - \ln(\dot\varepsilon_{L;L;L})}{\ln(\sigma_{L;H}) - \ln(\sigma_{L;L})}; \quad r_H = \frac{\ln(\dot\varepsilon_{H;H;L}) - \ln(\dot\varepsilon_{H;L;L})}{\ln(\sigma_{H;H}) - \ln(\sigma_{H;L})}
\end{aligned}
\tag{15}
$$

It should be noted that in the above equations the index of $q_\bullet$ always corresponds to the second index of $\varepsilon_{\bullet\bullet\bullet}$ (stress level) and the index of $r_\bullet$ always corresponds to the first index of $\varepsilon_{\bullet\bullet\bullet}$ (temperature level).

4. Damage module

4.1 Damage model

The material damage due to significant creep strains is modelled by a damage measure which is incrementally accumulated at the end of a load step or substep. This damage includes also the prompt plastic deformation of the structure. The damage increment is:

$$\Delta D = \left[\frac{\Delta \varepsilon_{eqv}^{cr}}{\varepsilon_{frac}^{cr}(\sigma, T)} + \frac{\Delta \varepsilon_{eqv}^{pl}}{\varepsilon_{frac}^{pl}(T)} \right] \cdot R_v \qquad (16)$$

with ε_{frac}^{cr} being the creep fracture strain of the uniaxial creep test at constant stress and temperature and ε_{frac}^{pl} the plastic fracture strain (true strain) of the tensile test. R_v is a function which considers the damage behaviour in dependance on the triaxiality of the stress tensor /8/:

$$R_v = \frac{2}{3} \cdot (1 + \nu) + 3 \cdot (1 - 2\nu) \cdot \left(\frac{\sigma_H}{\sigma_{eqv}} \right)^2 \qquad (17)$$

where σ_H is the hydrostatic stress and σ_{eqv} is the von-Mises equivalent stress. Alternatively, the triaxiality function can be set to 1.
The damage increment is calculated for each element by averaging its nodal equivalent creep and plastic strains. The accumulated damage is

$$D = \sum_{i=1}^{ldstep} \Delta D_i \qquad (18)$$

If the element damage reaches the value of D=1, the element is killed by setting its death flag to 1 (refer to the „element birth and death" section of /3/).

4.2 Calling the damage procedure

The creep damage module is realized in the subroutine dmg_01 and a number of supporting routines. The invocation of this module is initialized by the user routine USER02. Once this initialization has been done the creep damage routine is automatically called after each substep or after each loadstep, depending on the settings with the **outres** command. This is realized by the UPFs USOLBEG, USSFIN, and USOLFIN respectively. If all substeps are written to the result file (**outres,all,all**), the damage routine is called after each substep. Otherwise the damage routine is called after each loadstep (default). If the analysis is a restart analysis, the previously accumulated damage values are resumed from the binary file <jobnam>.dsav.

Notes:

- It is *not* necessary to invoke the **usrcal** command, the initialization for USOLBEG, USSFIN, ULDFIN, USOLFIN and USEROU is done automatically.

- It is *not* recommended to issue outres specifications other than outres,**all**,*freq*, since element- and DOF-solutions need to be stored on the result file for creep damage evaluation

To activate the damage accumulation and the element killing, use the following commands:

```
!! set command name associated with UPFs (begin level)
    /ucmd,usr2,2
    /ucmd,usr3,3
!! calculate creep damage
    /solu
    ....

    !! set the output frequency (optinal)
    outres,all,freq

    !! initialize the damage procedure
    !! the calling frequency depends on the outres settings
    usr2,dmgcrit
    !! set fracture strains
    usr3,epfr_key,epbr0,c1tem,c1sig

    ....
    solv
    ....
```

Table 3: ANSYS input example for the use of the creep damage module

The argument *dmgcrit* of the **usr2**-command is used to select a damage criterion (*dmgcrit*=2: triaxiality according to eq. 17; *dmgcrit*=1: $R_v = 1$, *dmgcrit*=0: no damage calculation).

The **usr3**-command provides an additional possibility to calculate the creep fracture strain or the plastic fracture strain according to:

$$\begin{aligned} &\text{epfr\_key} = 0: \quad \varepsilon^{cr}_{frac} = \text{epbr0} + \text{c1tem}\cdot T + \text{c1sig}\cdot\sigma \\ &\text{epfr\_key} = 1: \quad \varepsilon^{pl}_{frac} = \text{epbr0} + \text{c1tem}\cdot T \end{aligned} \tag{19}$$

If the **usr3**-command is not entered (or entered with all arguments zero), the current creep fracture strain is calculated from the creep data base values (see creep file example, table 1) by multi-linear interpolation:

$$\varepsilon^{cr}_{frac}(T,\sigma) = \frac{(T_H - T)\cdot(\sigma_{L;H} - \sigma)\cdot\varepsilon^{frac}_{L;L}}{(T_H - T_L)\cdot(\sigma_{L;H} - \sigma_{L;L})} + \frac{(T_H - T)\cdot(\sigma - \sigma_{L;L})\cdot\varepsilon^{frac}_{L;H}}{(T_H - T_L)\cdot(\sigma_{L;H} - \sigma_{L;L})} + \frac{(T - T_L)\cdot(\sigma_{H;H} - \sigma)\cdot\varepsilon^{frac}_{H;L}}{(T_H - T_L)\cdot(\sigma_{H;H} - \sigma_{H;L})} + \frac{(T - T_L)\cdot(\sigma - \sigma_{H;H})\cdot\varepsilon^{frac}_{H;H}}{(T_H - T_L)\cdot(\sigma_{H;H} - \sigma_{H;L})} \quad \mathbf{(20)}$$

For the meaning of the indices see sections 2 and 3. The plastic fracture strain ε^{pl}_{frac} is calculated from the last strain-stress point of the MISO-table or of the MKIN table (interpolation between the temperatures). Refer to the **tb, tbpt, tbtemp** commands /3,4/.

The **usr2**- and **usr3**-commands can also be used with an ANSYS standard creep law (c6<100), however, in this case the creep fracture strain must be entered via **usr3** since it is otherwise not available.

After the solv command the scalar ANSYS parameter „**dmgmax**" is available which represents the maximum creep damage of all elements at the current loadstep.

Note:
If an element is killed, the stress distribution may promptly and significantly change. So it is recommended to observe the process of element killing after each load step using the standard ANSYS commands (esel, *get). If one or more elements are killed during the load step, the time increment for the next step should be sufficiently small.

4.3 Plotting the damage

The damage D is calculated for each element whenever the cr_dmg-routine is invoked (section 4.2). To make this quantity available for the postprocessing it is written it to the NMISC-records of the elements (refer to /5/ for the description of the standard non-summable miscellaneous element output - NMISC). The UPF USEROU was employed to realize the additional nmisc-output. The damage value is automatically stored on the first place after the last standard-NMISC output. The graphical output during the postprocessing can easily be realized with the commands **etable, pletab, esol** /4/. The output frequency to the result file depends on the settings of the **outres** command /4/. Table 4 shows an ANSYS input example.

```
!! post-processing
    /post1
    set,...
    etable,crdmg,nmisc,26    !! 26 if element is plane42
    pletab,crdmg,avg
```

Table 4: ANSYS input example for plotting of the creep damage

Note: The creep damage calculated with the stresses and strains of the n-th result set is stored in the (n+1)-st result set of the result file. This unintended shift is due to the fact that the USEROU routine is called *before* USSFIN (damage calculation after a substep) and ULDFIN (damage calculation after a loadstep). This cannot be influenced by the UPF programmer. However, in the additional file ***<jobnam>.dmg*** (see section 5) the assignment between the element results and the calculated damage is correct. The command-macro „pldmg.mac" can be used as a work around for this problem.

5. Additional files created

5.1 Log-File of creep data base input

To verify the input of the creep data files the log-file ***<jobnam>.crlg*** can be created (*jobnam* defaults to „file"). This ASCII file contains the base name of creep data files, the number of temperature levels, the number of stress levels per temperature, the number of strain rate - strain pairs for each level and the first twenty of such pairs. The file is written, if the third argument of the ucri-command is set to 1: (for example: `usr1,testcrp,5,1`)

5.2 Log-File of the damage calls

This ASCII file is named ***<jobnam>.dmg*** and contains information on current load step, substep, time, number of elements and number of nodes. Moreover, for each element the average temperature, stresses, creep strains, plastic strains and accumulated damage is recorded. The log-information is appended each time the dmg-routine is called. The file is newly created in the first loadstep, if the analysis is not a restart.

5.3 Damage backup file

This binary file is named ***<jobnam>.dsav*** and contains the accumulated damage. This file is necessary if a restart analysis is performed. In the case of a restart the creep data base must be input in the same way as in the first analysis, since the creep data base is not stored in the db-file (cf. usr1-command). Additionally, the previously accumulated damage must be input from the dsav-file. This is automatically initialized if a restart analysis is performed.

6 Examples

6.1 Tensile bar

For verification of the creep module a simple model which consists of only one element was used (**Fig. A1**). The creep data base is generated according to equation (1) with the coefficients:

$$\begin{aligned} d_1 &= 6.9372E-23\,s^{-1} \quad & d_2 &= 3.367 \\ d_3 &= -0.459 & d_4 &= 22392\,K \\ & & \sigma_{ref} &= 1\,Pa \end{aligned} \tag{21}$$

For this „hypothetical" material a reference solution can be obtained using the ANSYS standard

creep equation. So the purpose of this example is
- to show that user creep module works correctly
- to demonstrate the influence of the interpolation method (cf. section 3)

The load for the model in **Fig. A1** is a constant stress of $\sigma = 200$ MPa and the temperature is 1000K. The creep data base is generated for the following temperature and stress levels:

$$\begin{array}{lll} T_L = 973.15\,K & \sigma_{LL} = 150\,MPa & \sigma_{LH} = 250\,MPa \\ T_H = 1073.15\,K & \sigma_{HL} = 150\,MPa & \sigma_{HH} = 250\,MPa \end{array} \quad \textbf{(22)}$$

The creep curves are calculated for a time range of 10 s with an integration step width of 0.025s. **Figure A2** shows the results for linear interpolation (cf. eq. 6) and **Fig. A3** for non-linear interpolation (cf. eq. 14). With linear interpolation the creep strain is too high (what is clear in view of eqs. 1 and 21), whereas the results with non-linear interpolation meet exactly the reference solution.

6.2 Disk with centre hole

The model is shown in **Fig. A4.** The disk is loaded at its free end by a tensile stress in x-direction. **Fig. A5** shows the tangential stress of the well known elastic solution exhibiting a stress maximum of $\sigma_{\varphi,max} = 3\sigma_0$ at the 90° position.

6.2.1 Material behaviour

The material of the disk is assumed to be the French reactor pressure vessel steel 16MND5. The creep behaviour of this material at high temperatures was investigated within a extensive experimental program funded by the European Commission /9/. Based on the experimental data the ANSYS creep data base is generated.

The creep curves are subdivided into three sections (cf. **Fig. A6**):

I primary creep range ($t < t_P$)
II secondary creep range ($t_P < t < t_S$)
III tertiary creep range ($t_S < t < t_T$)

The creep curves were generated at different temperature and stress levels according to the following equations:

$$\begin{array}{lll} \text{I:} & \dot{\varepsilon} = C_{1P}(T)\cdot\sigma^{c_{2P}(T)}\cdot t^{c_{3P}(T)} & \varepsilon = \dfrac{C_{1P}(T)}{1+c_{3P}}\cdot\sigma^{c_{2P}(T)}\cdot t^{c_{3P}(T)+1} \\ \text{II:} & \dot{\varepsilon} = C_{1S}(T)\cdot\sigma^{c_{2S}(T)}\cdot t^{c_{3S}(T)} & \varepsilon = \varepsilon_P + \dfrac{C_{1S}(T)}{1+c_{3S}}\cdot\sigma^{c_{2S}(T)}\cdot[t^{c_{3S}(T)+1} - t_P^{c_{3S}(T)+1}] \\ \text{III:} & \dot{\varepsilon} = C_{1T}(T)\cdot\sigma^{c_{2T}(T)}\cdot t^{c_{3T}(T)} & \varepsilon = \varepsilon_S + \dfrac{C_{1T}(T)}{1+c_{3T}}\cdot\sigma^{c_{2T}(T)}\cdot[t^{c_{3T}(T)+1} - t_S^{c_{3T}(T)+1}] \end{array} \quad \textbf{(23)}$$

For the primary creep stage an equivalent strain hardening representation is:

$$\dot{\varepsilon} = D_{1P}(T)\cdot\sigma^{d_{2P}(T)}\cdot\varepsilon^{d_{3P}(T)} \quad \text{with}$$

$$D_{1P} = \left[C_{1P}\cdot(1+c_{3P})^{c_{3P}}\right]^{\frac{1}{1+c_{3P}}} \qquad d_{2P} = \frac{c_{2P}}{1+c_{3P}} \qquad d_{3P} = \frac{c_{3P}}{1+c_{3P}} \tag{24}$$

The coefficients of the primary creep stage were taken from the work of IKONEN /9/ who summarized a couple of creep test performed in a temperature range of 600-1300°C. According to the experimental data in /9/ it was assumed that the transition between primary and secondary creep takes place at a creep strain of $\varepsilon = \varepsilon_P = 0.1$, and the transition between secondary and tertiary creep at a creep strain of $\varepsilon = \varepsilon_S = 0.2$. The fracture strain is about $\varepsilon_{frac} = \varepsilon_T = 0.8$. The times for the creep stages can be calculated by (cf eq. 23):

$$\begin{aligned}
t_P &= \left[\frac{(1+c_{3P})\cdot\varepsilon_P}{C_{1P}\cdot\sigma^{c_{2P}}}\right]^{\frac{1}{1+c_{3P}}} \\
t_S &= \left[\frac{(1+c_{3S})\cdot(\varepsilon_S-\varepsilon_P)}{C_{1S}\cdot\sigma^{c_{2S}}} + t_P^{c_{3S}+1}\right]^{\frac{1}{1+c_{3S}}} \\
t_T &= \left[\frac{(1+c_{3T})\cdot(\varepsilon_T-\varepsilon_S)}{C_{1T}\cdot\sigma^{c_{2T}}} + t_S^{c_{3T}+1}\right]^{\frac{1}{1+c_{3T}}}
\end{aligned} \tag{25}$$

To get a smooth transition between the stages of the creep curve the coefficients C_{1S} and C_{1T} must fulfill the relations:

$$\begin{aligned}
C_{1S} &= C_{1P}\cdot t_P^{(c_{3P}-c_{3S})} \\
C_{1T} &= C_{1S}\cdot t_S^{(c_{3S}-c_{3T})} \\
c_{2P} &= c_{2S} = c_{2T}
\end{aligned} \tag{26}$$

Table 5 shows the parameters for the creep curves. The coefficients D_{1P}, d_{2P} and d_{3P} are parameters fitted to the experiments by IKONEN /9/, the other coefficients follow from these fitted parameters by using the above equations. The coefficients are related to the stress unit MPa and the time unit sec.

T[K]	873.15	973.15	1073.15	1173.15	1273.15	1373.15	1473.15	1573.15
D_{1P}	3.890E-14	2.394E-12	1.412E-11	5.672E-11	2.540E-10	1.890E-09	1.571E-08	2.954E-07
d_{2P}	3.418	3.411	3.316	3.366	3.367	3.340	3.346	3.620
d_{3P}	-0.201	-0.218	-0.454	-0.455	-0.459	-0.470	-0.476	-0.436
C_{1P}	6.621E-12	2.778E-10	3.069E-08	8.072E-08	2.354E-07	1.028E-06	4.553E-06	2.542E-05
c_{2P}	2.846	2.800	2.281	2.313	2.308	2.272	2.267	2.521
c_{3P}	-0.167	-0.179	-0.312	-0.313	-0.315	-0.320	-0.322	-0.304
C_{1S}	eqs. (3,4)	eqs. (3,4)	eqs. (3,4)	eqs. (3,4)	eqs. (3,4)	eqs. (3,4)	eqs. (3,4)	eqs. (3,4)
c_{2S}	2.846	2.800	2.281	2.313	2.308	2.272	2.267	2.521
c_{3S}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C_{1T}	eq. (3,4)	eqs. (3,4)	eqs. (3,4)	eqs. (3,4)	eqs. (3,4)	eqs. (3,4)	eqs. (3,4)	eqs. (3,4)
c_{2T}	2.846	2.800	2.281	2.313	2.308	2.272	2.267	2.521
c_{3T}	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
ε_P	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
ε_S	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
ε_T	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8

Table 5: Parameters of the creep curves for the steel 16MND5

Figure A7 shows the creep strain and the creep strain rate at T=1373 K and $\sigma = 10\,\text{MPa}$ as an example.

In some load cases creep and prompt plasticity occur simultaneously. Therefore, the temperature dependent $\sigma - \varepsilon$ curves are shown in **Figure A8**. A multi-linear isotropic hardening is assumed and the flow rule of von-Mises is used.

6.2.2 Pure creep at constant temperature

At first the development of stresses and damage is studied assuming pure creep (no prompt plastic deformation) at a constant temperature of 1370 K. The creep deformation leads to a relatively quick release of the stress concentration near the hole. **Figures A9 and A10** show the tangential stress and the accumulated damage after 1000 s. **Figures A11 and A12** show the same quantities after 9900s. The elements which were killed in the meantime are not shown. It is remarkable that in spite of reduced cross section area the maximum stress is still lower than it was in the beginning of the calculation. This is a consequence of the stress relocation. The development of the damage and the stress over the time for some locations on the y-axis is shown in the **Figures A13 and A14**. In **Fig. A14** the effects of stress relocation and of the element killing can be clearly seen. **Figure A15** shows the creep strains of the two points $(0;r_H)$ and $(0;1.5r_H)$ over the time. The element at $(0;1.5r_H)$ fails at a slightly lower total creep strain which is a consequence of the higher triaxiality factor at this location.

6.2.3 Creep and plasticity

In this load case creep and plasticity are combined. As before the temperature is constant (1370 K) and the stress load is $\sigma_0 = 10\,\text{MPa}$. The **Figures A16 and A17** show the tangential stress and the accumulated damage after 1000 s. **Figures A18 and A19** show the same quantities after 4800s. The development of the damage and the stress over the time for some locations on the y-axis is shown in the **Figures A20 and A21. Figure A22** shows the equivalent creep strain and the equivalent plastic strain of the two points (0;r_H) and (0;1.5r_H) over the time.

The stress near the hole exhibits a slightly different behaviour compared to the pure creep case (ref. **Fig. A21**, red curve). It starts with a lower value since a prompt plastic deformation occurs and the maximum stress is limited to the initial yield stress. After that the stress is increasing due to the plastic hardening of the material. After some 800s a stress relocation starts to develop as a consequence of the increasing creep deformation.

The progress of damage looks also somewhat different than in the pure creep case (ref. **Fig. A20**). The damage value at t = 0 s is not zero but up to 0.29. This is the result of the prompt plastic deformation. The contribution of the plastic strain to the damage leads to a shorter time at which the first element fails (4300 s vs. 8400 s in the pure creep case, Fig. A13). The failure of the first element causes a prompt increase of the damage in the neighbouring elements since the stress relocation leads to another prompt plastic strain increment. This effect can also be seen in the strain curves **(Fig. A22)**. Contrary, in the pure creep case the killing of the first element only causes a steeper gradient in the damage curves of the neighbouring elements (Fig. A13).

Comparing the damage **(Fig. A20)** with the plastic strains and creep strains **(Fig. A22)** it can be stated that the relative contribution of the plastic strain to the damage is larger than that of the creep strain. This is due to the fact that in the case of the 16MND5 steel the plastic fracture strain is smaller than the creep fracture strain (see also eq. 16).

References

/1/ ANSYS Programmer's Manual. ANSYS, Inc., 1998

/2/ ANSYS User's Manual - Theory (Rev. 5.5). ANSYS, Inc., 1998

/3/ ANSYS User's Manual - Analysis guide (Rev. 5.5). ANSYS, Inc., 1998

/4/ ANSYS User's Manual - Command reference (Rev. 5.5). ANSYS, Inc., 1998

/5/ ANSYS User's Manual - Elements manual (Rev. 5.5). ANSYS, Inc., 1998

/6/ Becker, A.A.: Background to Material Non-Linear Benchmarks. NAFEMS-report R0049 (International Association for the Engineering Analysis Community)

/7/ Azodi, D., P. Eisert, U. Jendrich, W.M. Kuntze: GRS-Report GRS-A-2264

/8/ Lemaitre, J.: A Course on Damage Mechanics. ISBN 3-540-60980-6, 2nd edition Springer-Verlag Berlin, Heidelberg, New York, 1996

/9/ Ikonen, K., 1999, "Creep Model Fitting Derived from REVISA Creep, Tensile and Relaxation Measurements", Technical Report MOSES-4/99, VTT-Energy, Espoo, Finland.

Appendix 1: Figures

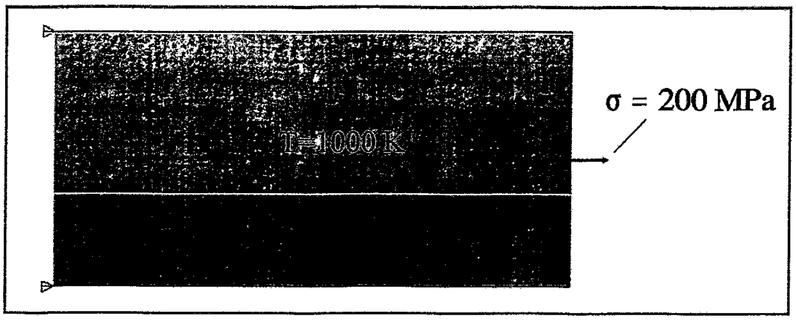

Fig. A1: Model of the tensile bar

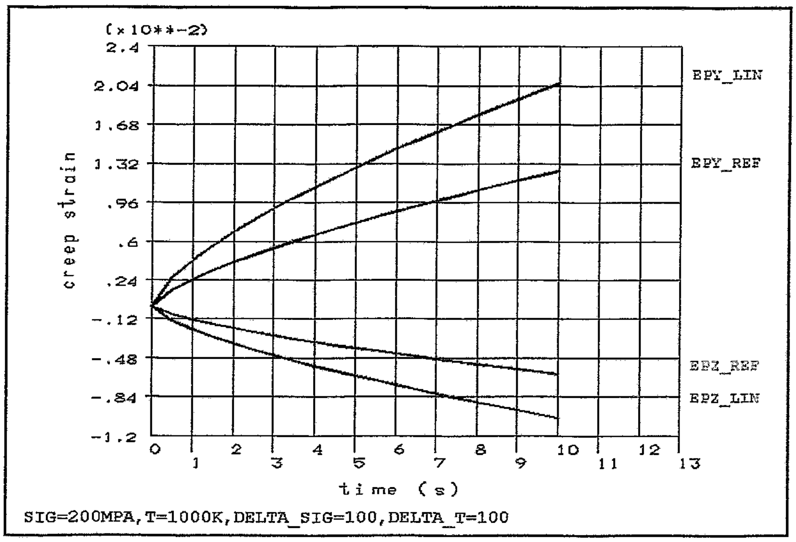

Fig. A2: Tensile bar, creep curves (axial strain and lateral strain) vs. time, linear interpolation and reference solution.

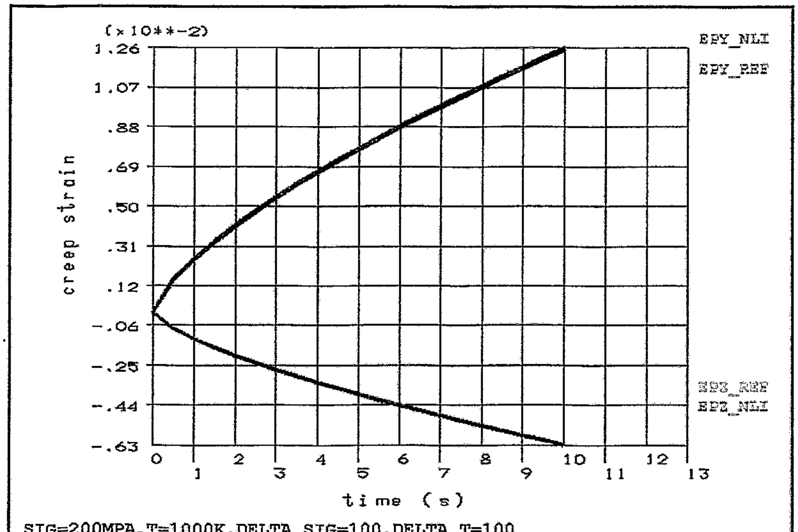

Fig. A3: Tensile bar, creep curves (axial strain and lateral strain) vs. time, non-linear interpolation and reference solution

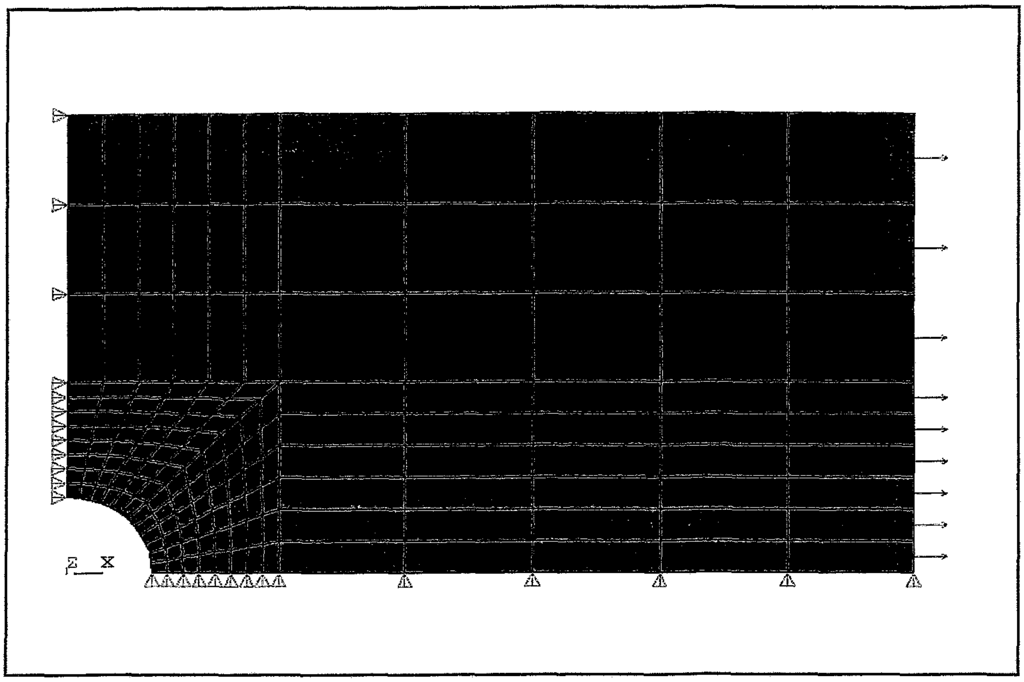

Fig. A4: Model of the rectangular disk with a centre hole; uniform stress at x=a (red); symmetry conditions at x=0 and y=0 (blue)

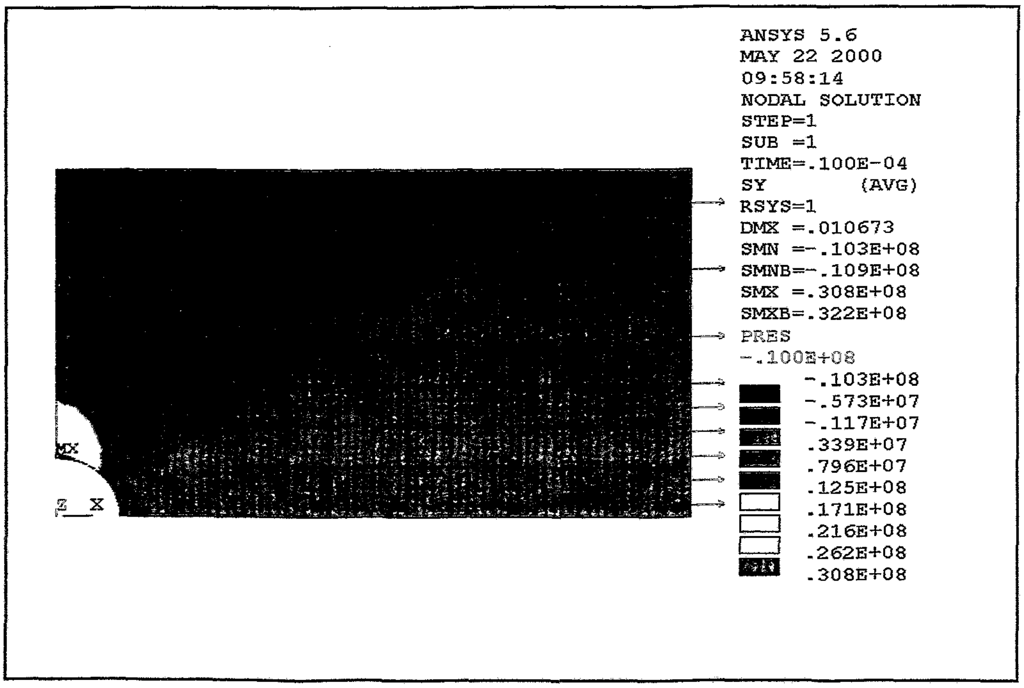

Fig. A5: Tangential stress (elastic solution)

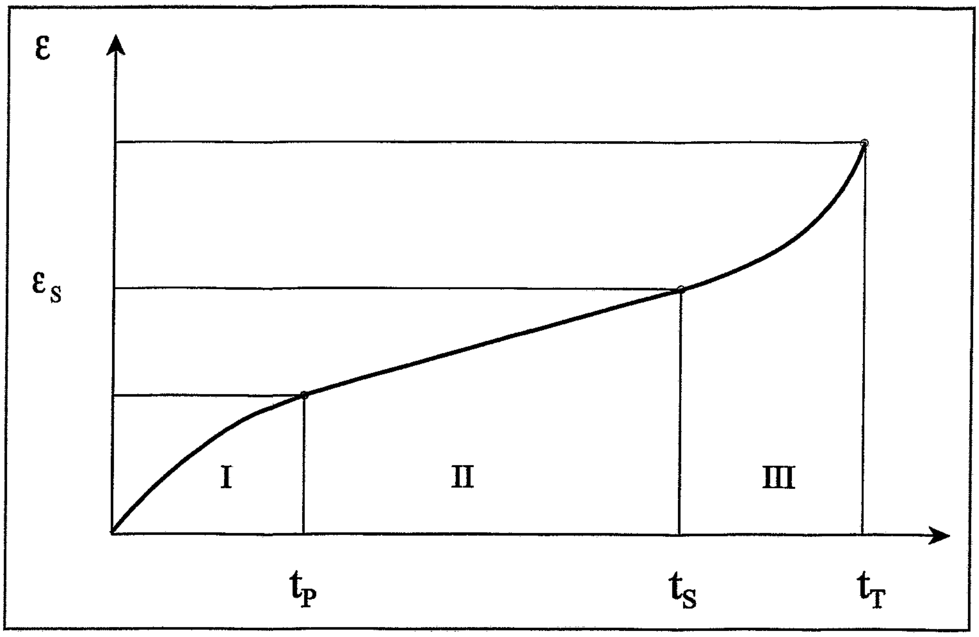

Fig. A6: Scheme of a creep curve with primary, secondary and tertiary creep stage

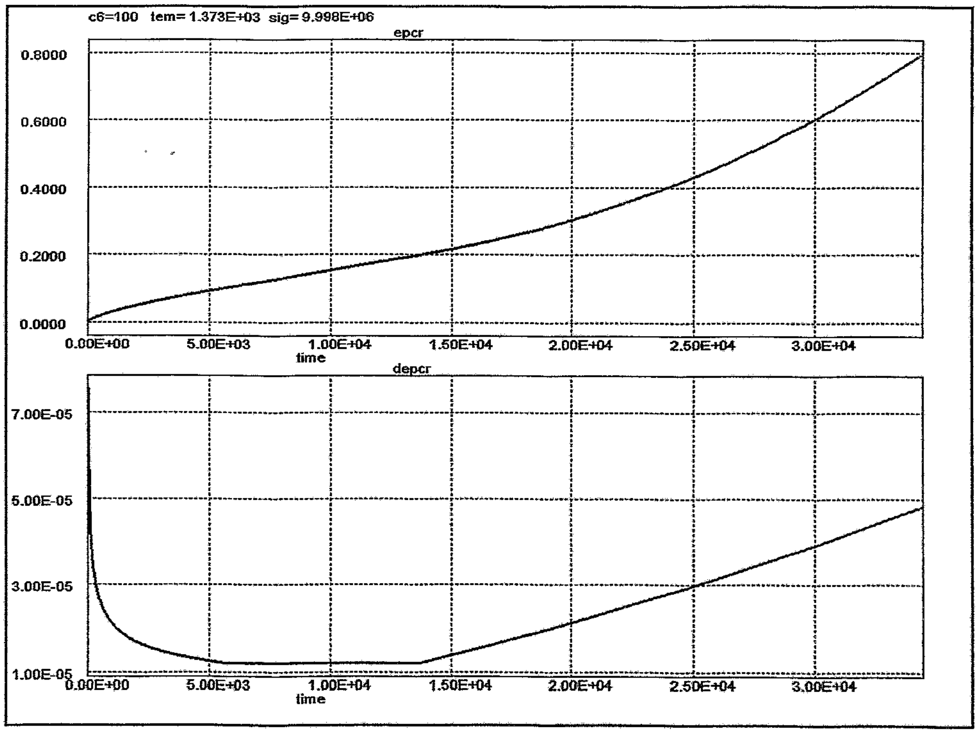

Fig. A7: Creep curves (creep strain and creep strain rate over the time) for 16MND5 at $T = 1373K; \sigma = 10MPa$.

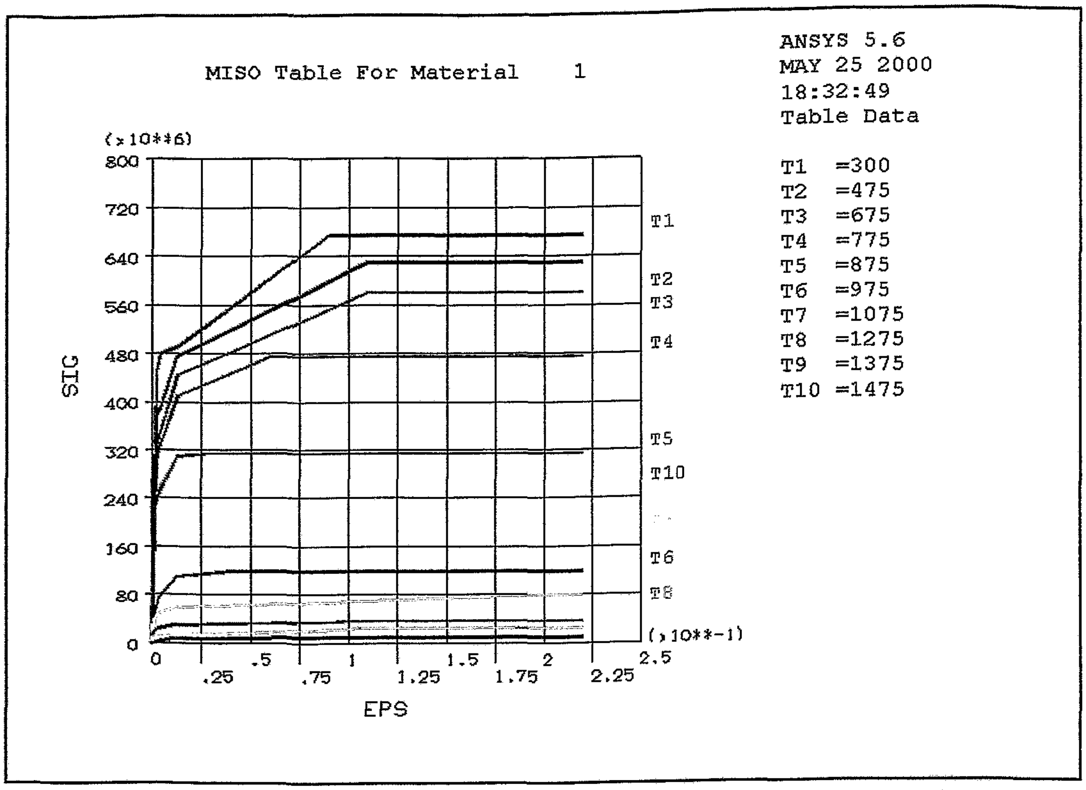

Fig. A8: Stress-strain-curves of 16MND5 for the temperature range from 300K to 1475 K (true stress and strain) based on /9/.

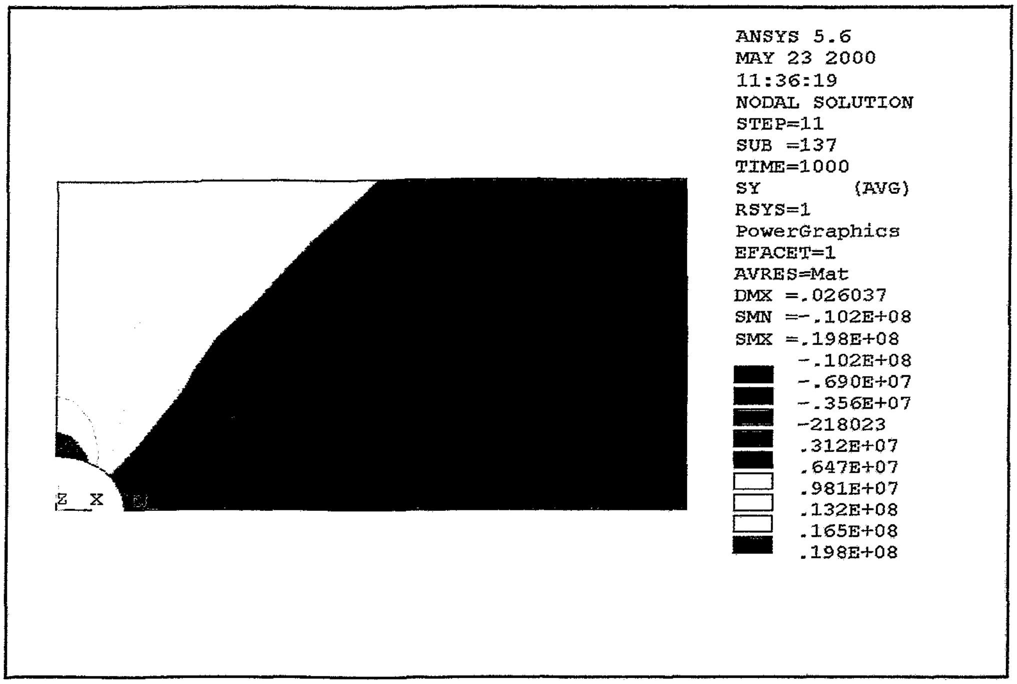

Fig. A9: Disk with hole (Material 16MND5). Pure creep at T=1373 K and $\sigma_0 = 10\text{MPa}$. Tangential stress after 1000 s.

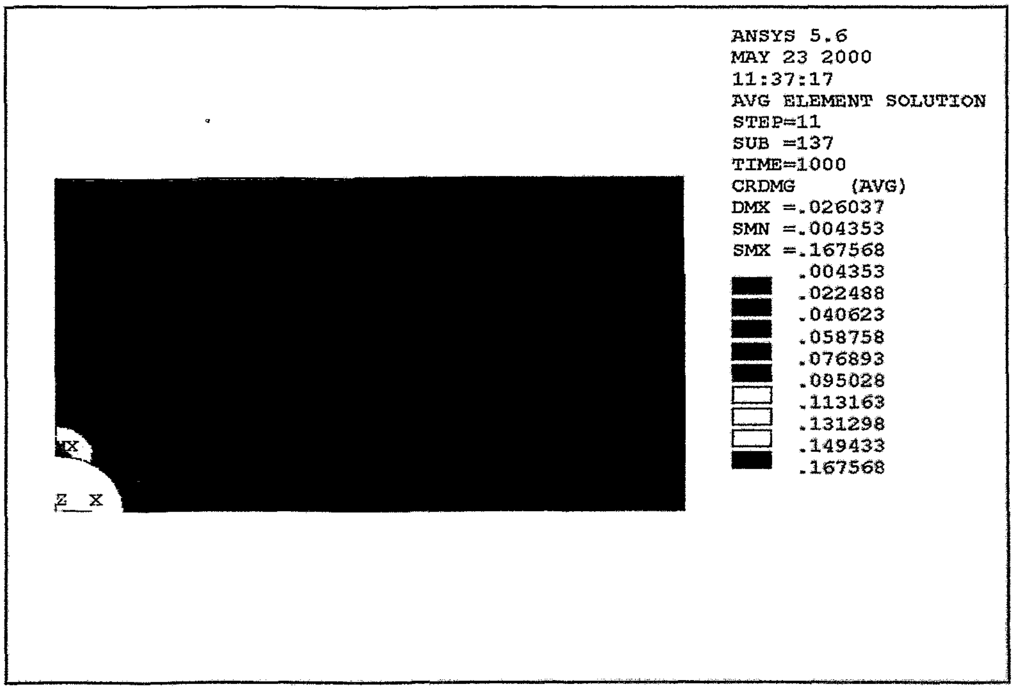

Fig. A10: Disk with hole (Material 16MND5). Pure creep at T=1373 K and $\sigma_0 = 10\text{MPa}$. Damage after 1000 s.

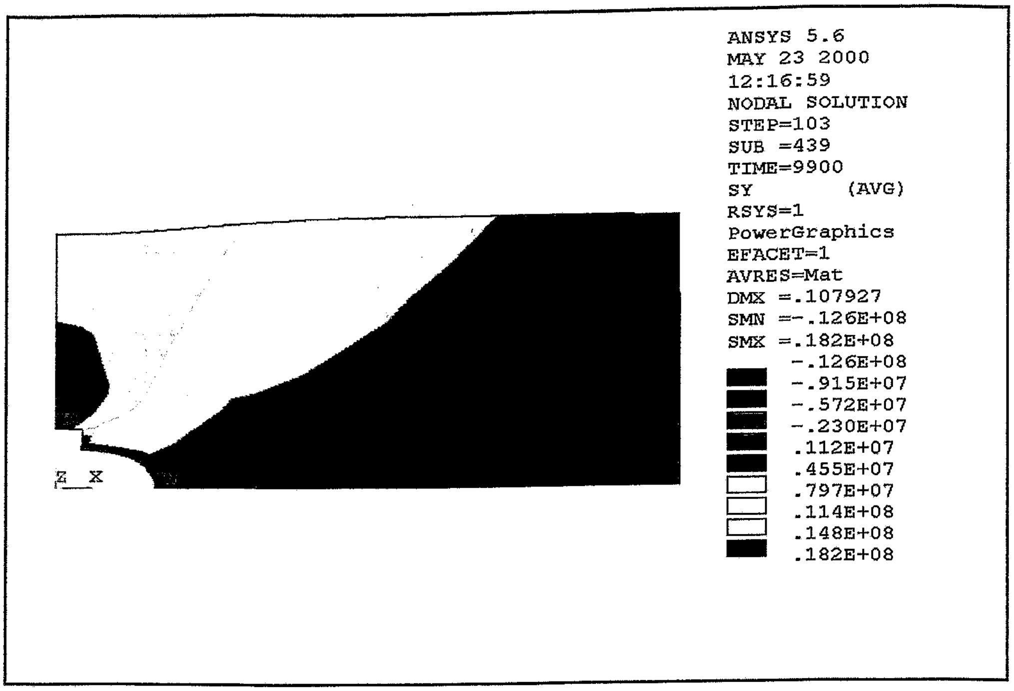

Fig. A11: Disk with hole (Material 16MND5). Pure creep at T=1373 K and $\sigma_0 = 10\text{MPa}$. Tangential stress after 9900 s.

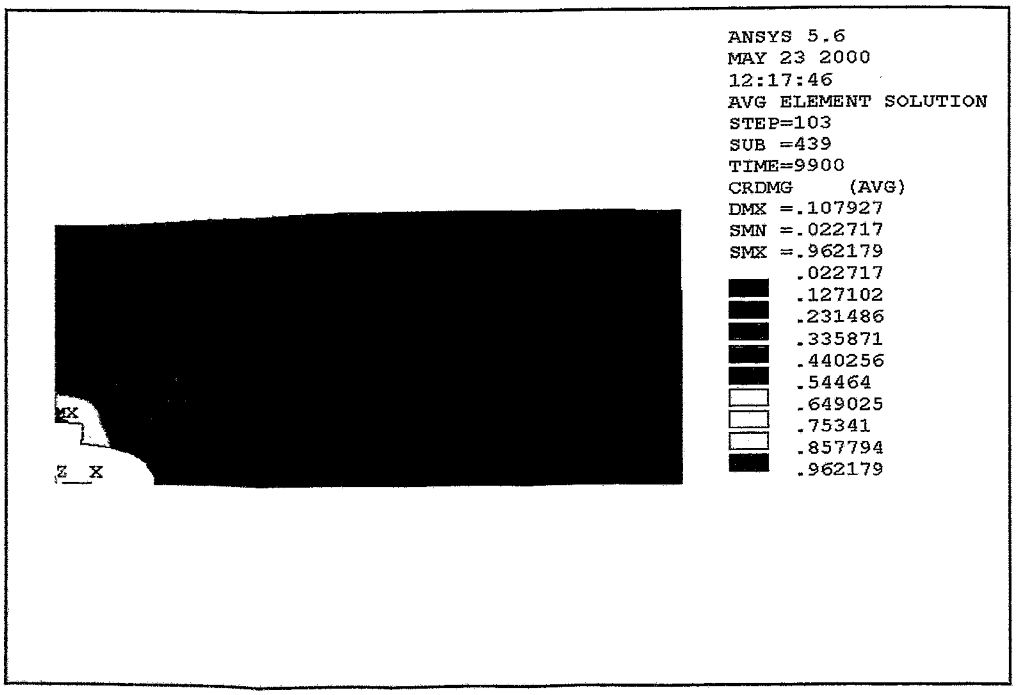

Fig. A12: Disk with hole (Material 16MND5). Pure creep at T=1373 K and $\sigma_0 = 10\text{MPa}$. Damage after 9900 s.

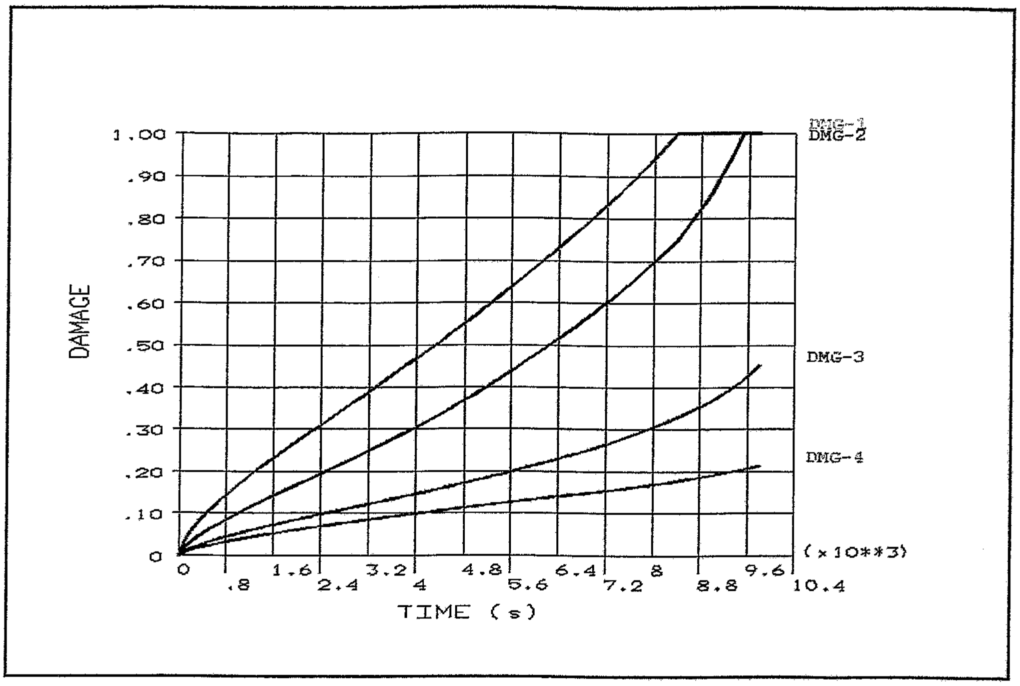

Fig. A13: Disk with hole (Material 16MND5). Pure creep at T=1373 K and $\sigma_0 = 10\text{MPa}$. Damage over time at four points: DMG-1 at (x=0; y=r_H), DMG-2 at (0; 1.5 r_H), DMG-3 at (0; 2.5 r_H), DMG-4 at (0; b).

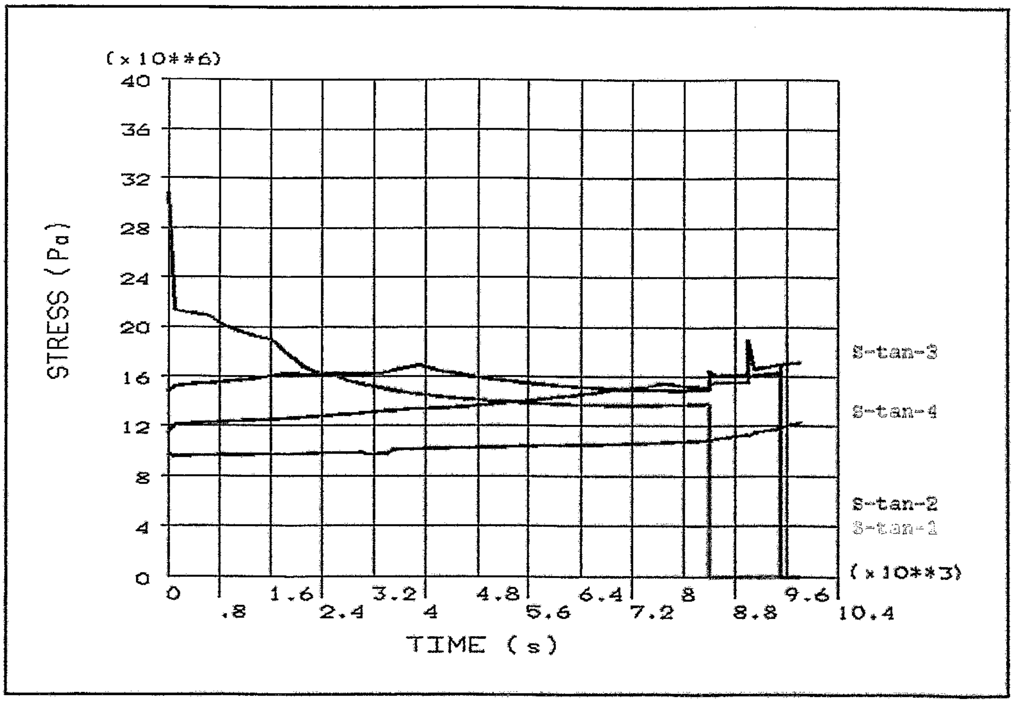

Fig. A14: Disk with hole (Material 16MND5). Pure creep at T=1373 K and $\sigma_0 = 10\text{MPa}$. Tangential stress over time at four points (see Figure A13)

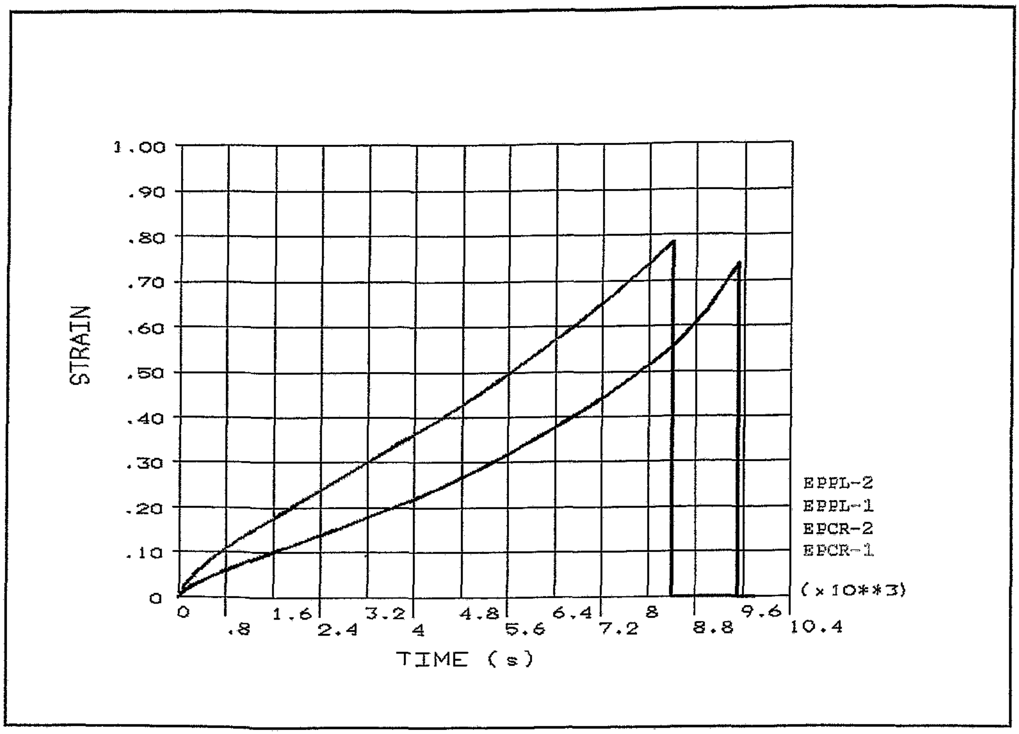

Fig. A15: Disk with hole (Material 16MND5). Pure creep at T=1373 K and $\sigma_0 = 10\text{MPa}$. Equivalent creep strain over time at points 1 and 2 (see Figure A13)

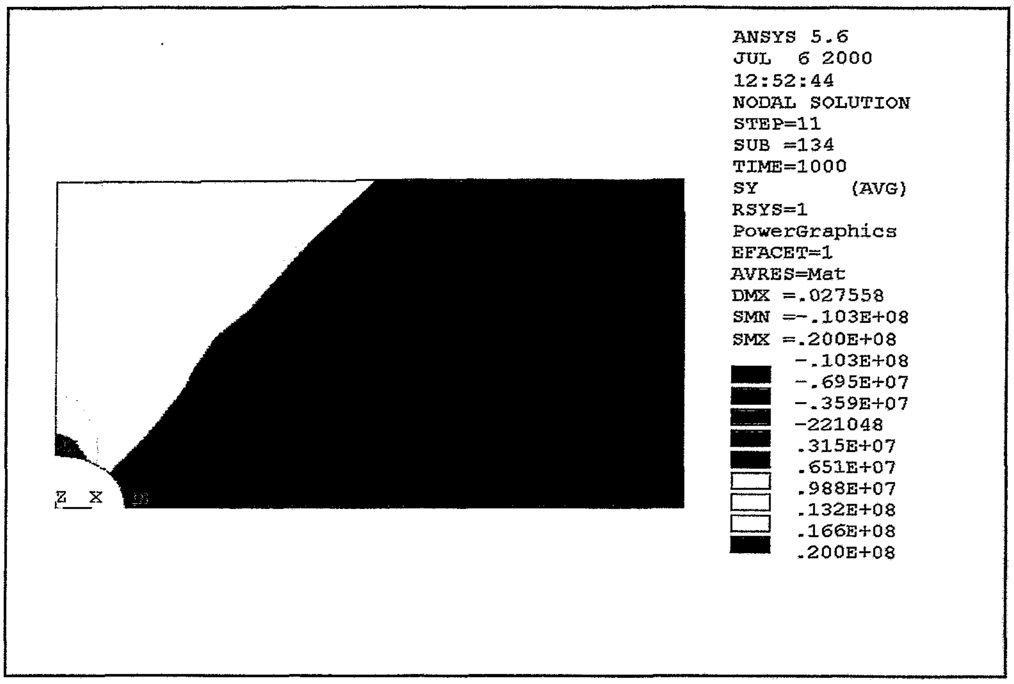

Fig. A16: Disk with hole (Material 16MND5). Creep and plasticity at T=1373 K and $\sigma_0 = 10\text{MPa}$. Tangential stress after 1000 s.

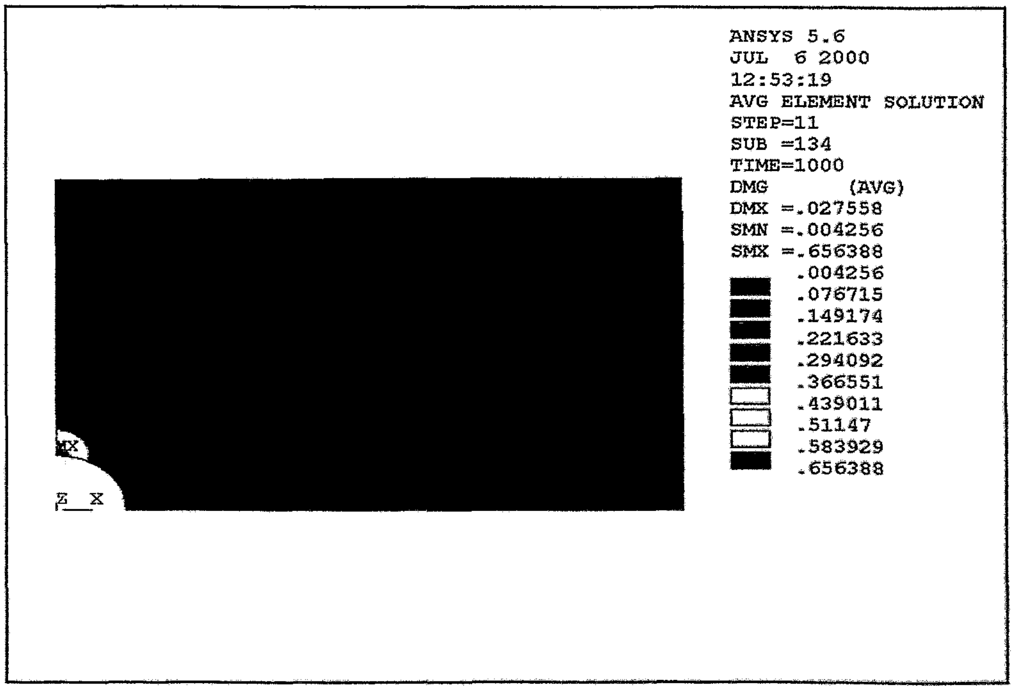

Fig. A17: Disk with hole (Material 16MND5). Creep and plasticity at T=1373 K and $\sigma_0 = 10\text{MPa}$. Damage after 1000 s.

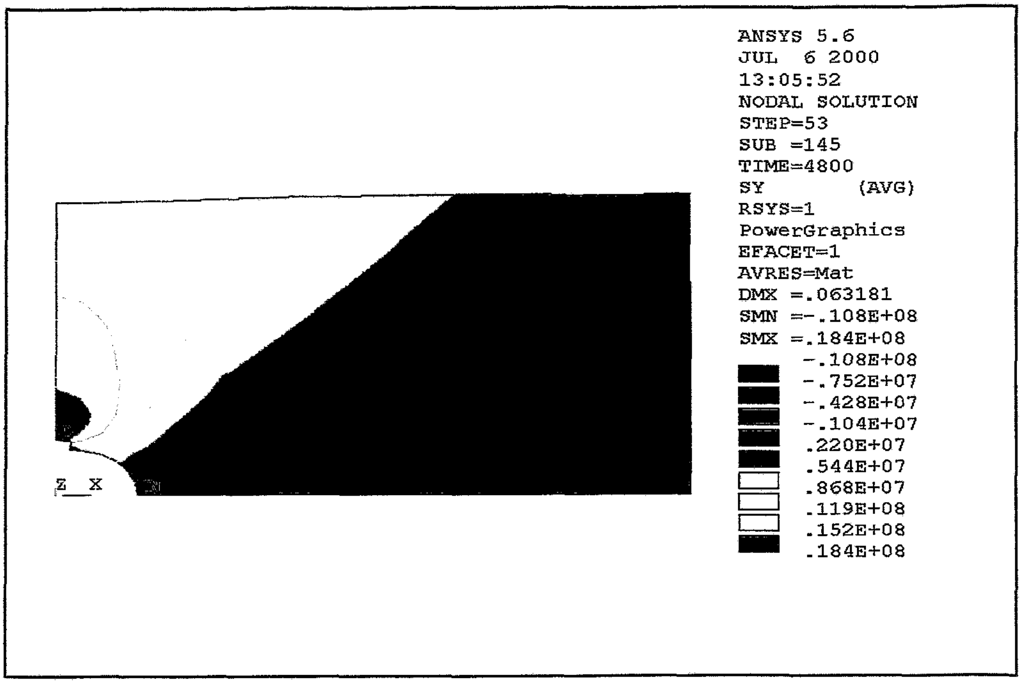

Fig. A18: Disk with hole (Material 16MND5). Creep and plasticity at T=1373 K and $\sigma_0 = 10\text{MPa}$. Tangential stress after 4800 s.

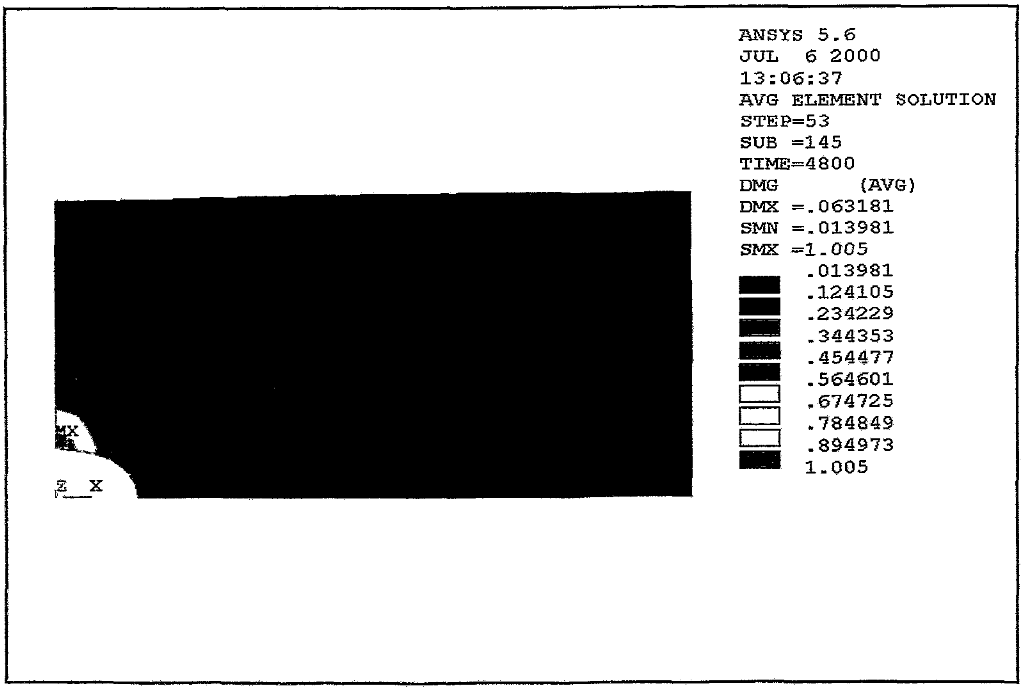

Fig. A19: Disk with hole (Material 16MND5). Creep and plasticity at T=1373 K and $\sigma_0 = 10\text{MPa}$. Damage after 4800 s.

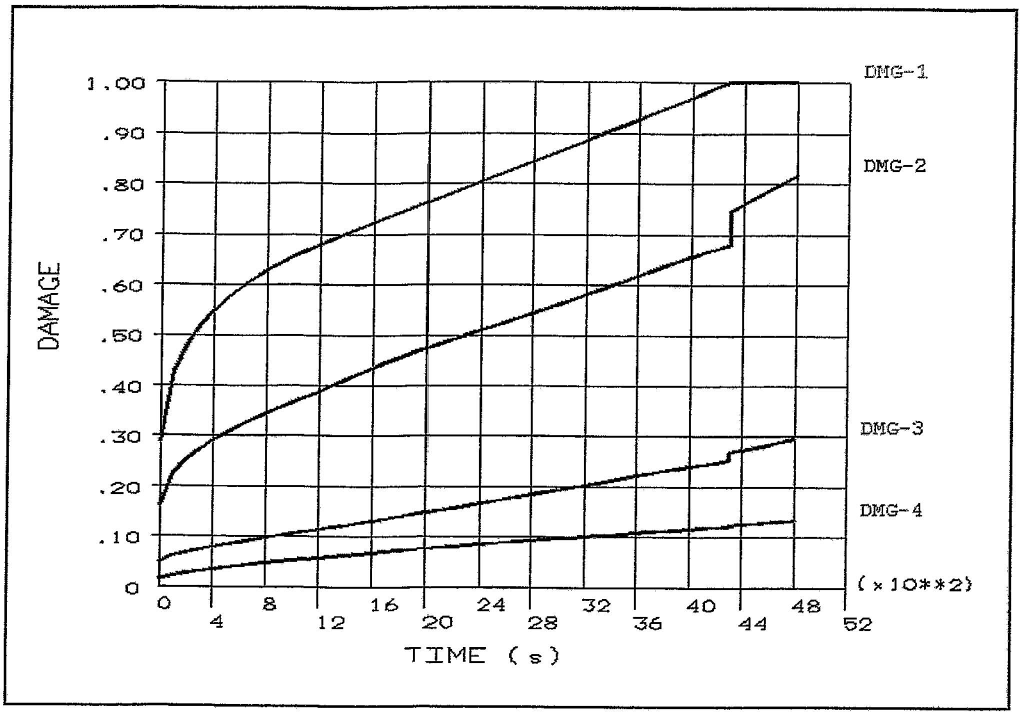

Fig. A20: Disk with hole (Material 16MND5). Creep and plasticity at T=1373 K and $\sigma_0 = 10 \mathrm{MPa}$. Damage over time at four points: DMG-1 at (x=0; y=r_H), DMG-2 at (0; 1.5 r_H), DMG-3 at (0; 2.5 r_H), DMG-4 at (0; b).

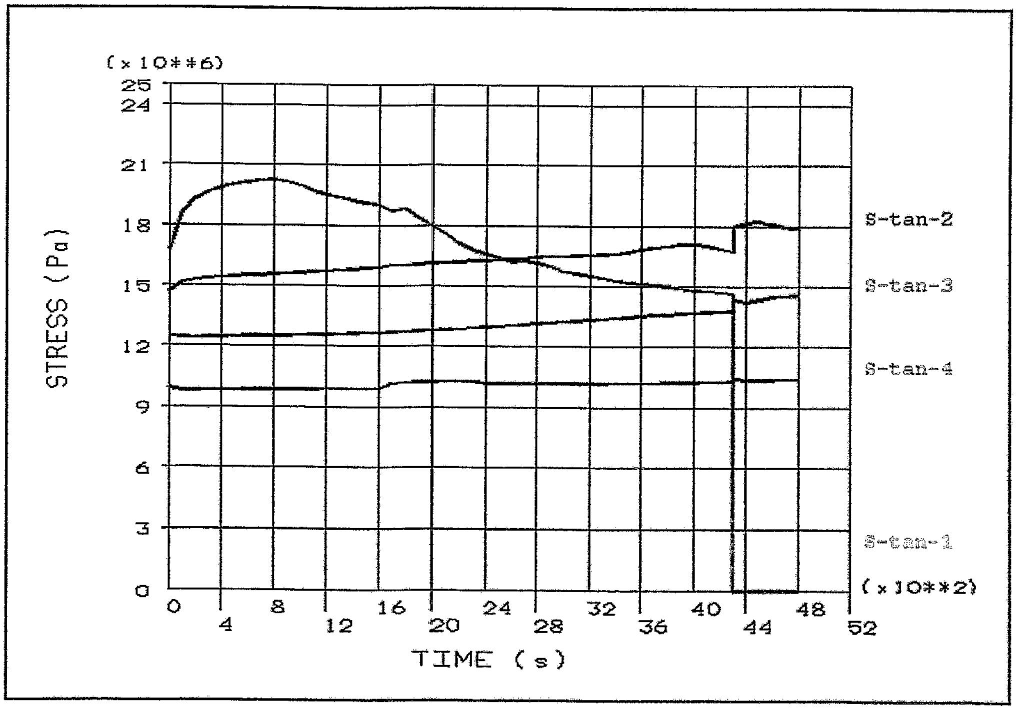

Fig. A21: Disk with hole (Material 16MND5). Creep and plasticity at T=1373 K and $\sigma_0 = 10 \mathrm{MPa}$. Tangential stress over time at four points (see Figure A13)

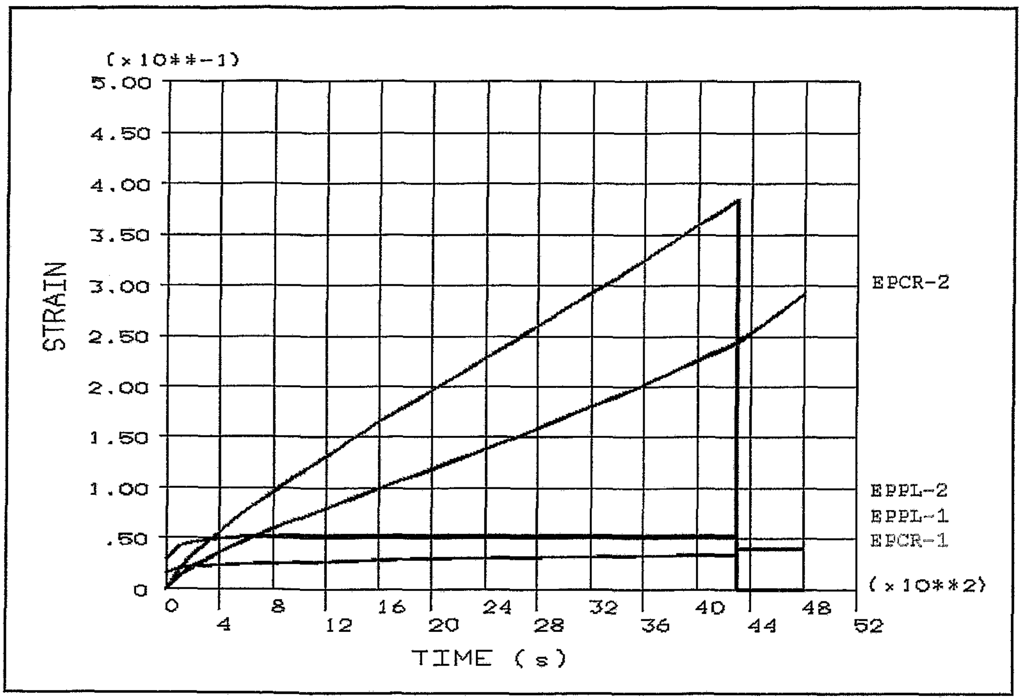

Fig. A22: Disk with hole (Material 16MND5). Creep and plasticity at T=1373 K and $\sigma_0 = 10\text{MPa}$. Equivalent creep strain and equivalent plastic strain over time at points 1 and 2 (see Figure A13)

Appendix 2: CRPGEN command reference

CRPGEN for WIN NT / 9x (Revision 1.4)
Command Reference

clear

Delete all preceding settings of the **crdat, time, epmax, sigma, temp** commands.

crdat,*stloc,c1,c2,c3,c4,c5,c6*

Set the creep parameters (c_1-c_5,c_7-c_{10}) and select creep law (c_6). *STLOC* refers to the first creep parameter to be input (e.g. *crdat,4,c1,c2,c3* reads the constants c_4, c_5 and c_6).
If $c_6 = 0$ use strain hardening model according to:

$$\dot{\varepsilon} = c_1 \cdot \sigma^{c_2} \cdot \varepsilon^{c_3} \cdot \exp\left[-\frac{c_4}{T}\right] \qquad (c_5 \text{ is unused})$$

If $c_6 = 1$ use time hardening model according to:

$$\dot{\varepsilon} = c_1 \cdot \sigma^{c_2} \cdot t^{c_3} \cdot \exp\left[-\frac{c_4}{T}\right] \qquad (c_5 \text{ is unused})$$

If $c_6 = 2$ use time hardening model according to:

$$\dot{\varepsilon} = c_1 \cdot \sigma^{c_2} \cdot [t^{c_3} + c_7 \cdot t^{c_8} + c_9 \cdot t^{c_{10}}] \cdot \exp\left[-\frac{c_4}{T}\right] \qquad (c_5 \text{ is unused})$$

cnvsig,*sigfac,[sigoff]*

Convert a creep function to another stress unit system (e.g. from Pa to MPa). This command can be used to change existing creep data files to use them in different engineering unit systems. The stress level in a creep data file is changed according to:
$\sigma_{new} = \text{sigfac} \cdot \sigma_{old} + \text{sigoff}$. The command should only be used in connection with **read** and **write.**
Example:

```
read,file1.c01
cnvsig,1.0e+06
write,file2.c01
```

epmax,*epmax*

Define the maximum strain up to which the creep curve $\dot{\varepsilon}(\varepsilon)$ is to be generated. Command is required for $c_6 = 0$ (see **crdat).**

exit

Terminate the program. The data base is not automatically saved. Use the **write** command.

plot,*plkey*

Plot creep curves. If *plkey* = 0, plot $\varepsilon(t)$ and $\dot{\varepsilon}(t)$, if *plkey* = 1 plot $\dot{\varepsilon}(\varepsilon)$.

prdat

Print creep parameters to standard output.

read,*fname*

Read a creep data file. *fname* is the ASCII file containing a creep curve

$$\dot{\varepsilon} = \dot{\varepsilon}(\varepsilon)\Big|_{\sigma=const;T=const}$$

rsolve

Continue the solution after the change of creep parameters. This command can be used to generate creep curves which are governed by different creep equations (e.g. primary, secondary and tertiary creep stage). A **solve** command must have been entered before the first **rsolve** command. An additional **time** or **epmax** command is also required, where the *endtime* or *epmax* argument must be greater than that of the previous **solve** or **rsolve** process. Stress and temperature must not be changed after the **solve** command.

Example:

```
sigma,100
temp,800
crdat,1,1.0e-16,2.2,-0.14,9860,0,1
time,3000
steps,200
solve
crdat,1,3.0e-15,2.2,0.01,9860,0,1
time,5600
steps,150
rsolve
write,crpdat.c01
```

sigma,*sigma*

Define the stress level for the creep curve.

solve

Start generation of creep data. Command requires creep parameters to be input (see **crdat, time, epmax, steps**).

steps,*nstep*

Define the number of time / strain steps; *nstep* pairs $[\varepsilon;\dot{\varepsilon}]$ are to be generated.

temp,*temp*

Define the temperature level for the creep curve to be generated.

time,*endtime*

Set maximum time for creep data generation to *endtime*. This command is required for $c_6 = 1$ (see **crdat**-command) and for plotting time-dependent creep curves (see **plot**,0)

write,*fame*

Write generated creep data to the ASCII file *fname*.

General Shell Commands (Macro-Language)

delvar

Clear all defined variables. See also: **Definition of variables, *stat**

*do,*dovar,dostart,dostop[,dostep]*

Execute a do-loop within a command file (see **/inp**). The command is not possible from standard input. The loop variables *dostart,dostop,dostep* can be numbers, variables or expressions (*dostart* <= *dostop*; *dostep* > 0). The loop must be closed by ***enddo** in the same command file where it was opened. The loop body must not refer to another command file. Up to 10 do-loops can recursively be opened at the same time.

echo,*key*

Switch on/off command echo to standard output

/eof

Terminate the input stream from the current input file and redirect it to the prior unit.

help

Open help file (Acrobat Reader required).

```
*if,expr1,op,expr2,then
        command-block
[*elseif,expr1,op,expr2,then]
        [command-block]
....
[*else]
        [command-block]
*endif
```

Branching within a command file (not possible from standard input). *expr1* and *expr2* are numerical values, variables or expressions; *op*={eq, ne, lt, gt, le, ge}. The syntax is similar to the FORTRAN programming language, but ***endif** and **then** are required. Recursive if-statements are possible.

/input,*fname*

Directs input stream to file *fname*. Recursive switches are possible.
If files with extention „.mac" (macros) are existing in the current working directory, these macros files can be input by simply typing the basename of the macro file (e.g. type „mac01" to input the command file „mac01.mac").

/out,*fname*

Switch program printouts from standard output to file *fname*. If f*name* is not input, switch back to standard output.

***stat**

Print all defined variables. See also: **Definition of variables, delvar**

Definition of variables

Variables can be defined by
var=value
where *value* can be a number, a name of a variable, a numerical expression or a character string. The name of the variable *var* must begin with a letter and can consist of 8 characters. Expressions can be assembled from numbers and numerical variables. Valid operators are:

left (opening) bracket: (	right (closing) bracket:)
plus (addition): +	minus (subtraction): -
multiplcation: *	division: **/**
power operation: **	

The defined variables are global ones. The character equivalents of variable values can be used in command strings by: *str1**%varname%**str2* (%varname% is replaced by its character value). For example the following commands are equivalent:

var1='xyz'	
read,file%var1%	**read**,filexyz
vnum=23	
siglab,1,s%vnum%b	**siglab**,1,s23b

Floating point variables are truncated to integer. If for example a=3.25E+01, then the result of %a% is „32".