Data structure `sd_cham_mater`

Summarized:

The data structure sd_cham_mater here `is described` (produced by the command `AFFE_MATERIAU`).

One also describes the data structure `sd_cham_mater_code` which is a temporary data structure used in the commands of computation.

Contents

1 General information

the data structure sd_cham_mater are produced by the operator AFFE_MATERIAU . It contains essential" and compulsory card an "containing the name of the materials (sd_mater) affected on meshes of the mesh.

For a rapid access with the characteristics of the materials in the routines te00xx , one introduced the notion of "coded material" (sd_mater_code).
Consequently, it was necessary to create the sd_cham_mater_code which is a card in which the materials were replaced by "coded materials".

The data structures sd_mater and sd_mater_code are described in [D4.06.18]

2 Tree structures

```
sd_cham_mater (K8)
   (O)    ".CHAMP_MAT"       :       sd_carte (NOMMATER)
   (F)    ".TEMPE_REF"       :       sd_carte (TEMP_R)
   (F)    "$VIDE"           :       sd_cham_mater_varc
   (F)    ".COMPOR"          :       sd_carte (COMPOR)

sd_cham_mater_code (K19)
   (O)    "$VIDE"            :       sd_carte (ADRSJEVE)

sd_cham_mater_varc (K8)
   (O)    ".CVRCNOM"         :       OJB   S   V   K8   long=nbcvrc
   (O)    ".CVRCGD"          :       OJB   S   V   K8   long=nbcvrc
   (O)    ".CVRCVARC"        :       OJB   S   V   K8   long=nbcvrc
   (O)    ".CVRCCMP"         :       OJB   S   V   K8   long=nbcvrc
```

+ 2*nbvarc sd_carte whose name is deduced from the contents of object .CVRCVARC

3 Contained of the objects Jeveux

3.1 sd_cham_mater

.CHAMP_MAT

This sd_carte contains the name of (or of) the sd_mater affected (S) on each mesh of the mesh.

In the general case, each mesh is affected only by one sd_mater. But sometimes, it is necessary to indicate a list of sd_mater (when the nonlinear structural mechanics behavior is obtained by the command DEFI_COMPOR [U4.43.06]).

On each, mesh, one can affect up to 28 sd_mater different.

Note: particular management of the reference temperature:

> *The transformation* `sd_cham_mater` → `sd_cham_mater_code` *obliges to keep a correspondence (* `sd_mater` → *reference temperature). The reason is the transformation of the function* $alpha(T)$ *starting from 2 temperatures* `TREF` *and* `TDEF` *(see routine* `alfint.f`)*.* *This correspondence is not automatically assured any more since key word AFFE* `/TEMP_REF` *was replaced by AFFE* `_VARC/VALE_REF. One`
> *restores this correspondence at the end of operator AFFE_MATERIAU* (`routine` *cmtref.f*). `For` *that, one writes the reference temperature affected in sd_carte* `.CHAMP` *_MAT.* `One` *stores the reference temperature by writing 2 sd_mater* `additional` *(and fictitious): ("ACIER* `", "TREF⇒", "20.50")`. *The temperature*
> *is written with the F8.2 format .* `.COMPOR`

This

sd_carte `contains` information of assignment of the key word factor AFFE_COMPOR `(routine` rccomp.f `). The quantity`
associated with this card is COMPOR. `The 7` components
used are: RELCOM, `NBVARI, DEFORM, INCELA, C_PLAN, and` 2 sd `_cham_`

3.2 mater_code This sd

_carte `is` a copy of sd_carte `.CHAMP_MAT . The difference` between these 2 sd_carte `is that` the values of the card of the coded materials are addresses of sd_mater `_code instead of` names of the sd_mater `. Note:`

This sd

> *_carte* `is created` *on the Volatile basis at the beginning of the operators of computation (routine rcmfmc.f). As it contains addresses JEVEUX, it cannot have an unlimited life duration. sd_cham_mater*

3.3 _varc Vocabulary

, definitions One calls

"CVRC" (scalar command variable) a scalar real variable which influences the mechanical constitutive laws. Examples: temperature, hydration,… One calls

logically "VARC" (vectorial command variable) a set of "CVRC" connected between them. Example: metallurgical

phases of steel: percentages of ferrite, pearlite, of bainite,… the VARC and

CVRC are named (K8). To simplify, each isolated CVRC is attached to of a the same VARC `name`. The access to a scalar command variable (CVRC) is thus done logically by giving the name of the VARC and `the name of` the CVRC. Examples :

VARC=' TEMP'

`CVRC=' TEMP'      => temperature` (isolated CVRC `) VARC` = ' M_ACIER
`"CVRC=' PBAINITE  " => proportion` of bainite for the metallurgy of steel Note:

even if certain

> *CVRC are connected logically by VARC, it is* `necessary` *that the names all of the CVRC are distinct. The reason is that they are often the parameters of certain functions of the materials*

(sd_mater). When, *for example, a Young modulus is defined like a function of "PBAINITE "*, this name *must* *have an "absolute" meaning. One will call*

nbcvrc the number of affected CVRC (even partially) in the sd_cham_mater . If for example , L" user wrote: chmat= AFFE_MATERIAU

```
(… AFFE_VARC =  (_F (NOM_VARC
                = ' TEMP", GROUP_MA=' GM1',…) _F (NOM_VARC
                = ' M_ACIER', GROUP_MA=' GM2',…) The number of
```

CVRC (nbcvrc) is worth 8 (1 for VARC "TEMP " + 7 for VARC "M_ACIER ") even if all the CVRC are not affected on all the model. Object .CVRCNOM

This vector

gives the name of all the affected CVRC (same partially) on the model. L" order of the CVRC in this vector is L" order which is also used in the 4 other objects below. Object .CVRCVARC

This vector

gives the name of the VARC corresponding to the CVRC. For each

VARC (of name novarc), there exist 2 named cards: CART1 = sd_

```
cham_mater (1:8)/". " //novarc (1:8)/".1" CART2 = sd_
cham_mater (1:8)/". " //novarc (1:8)/".2" CART1 (sd_carte
```

(NEUT_R)) the command variable contains the values of reference (VALE_REF) affected for VARC. CART 2 (sd _carte

(NEUT_K16)) contains the necessary information to evaluate VARC the command variable. This information

is a "tuple" of 7 values (varc, tysd, nomsd, nomsym, proldr, prolga, finst) varc: name
of the VARC tysd: type
of the SD affected: "EVOL" /"FIELD" if tysd= "

FIELD ": nomsd: name

- of the field (presumedly steady) affected nomsym = proldr
- = prolga = finst = ""if tysd= "

EVOL ": nomsd: name

- of the affected sd_evol_xxx nomsym: symbolic name
- of the field to be used in the sd nomsd proldr : prolongation
- "on the right" C" is to be final moment said beyond L" L `evol_xxx ("EXCLUDED" /"CONSTANT"/"LINEAIRE"/"") prolga: prolongation
- "on the left" C" is to be said in on this side L" urgent initial of L `evol_xxx ("EXCLUDED" /"CONSTANT"/"LINEAIRE"/"") finst: name
- of the sd_fonction (or sd_formule) allowing to transform the "time of mechanical computation" into "times of L `evol_xxx". If finst=" " , the function "identity" is used. INST_EVOL =
  finst (INST_CALC) Object .CVRCGD

: name of

$V(k)$ the quantity associated with the field (or L `evol_xxxx) affected for kème CVRC. Object .CVRCCMP

: name of

$V(k)$ the component of the quantity associated with the field (or L `evol_xxxx) affected for kème CVRC. Example One

could

, for example, to find in these 4 objects: .CVRCNOM =

```
"TEMP" "SECH" "EPSXX    " "EPSYY    "…   .CVRCVARC   =
"TEMP" "SECH" "EPSA     " "EPSA     "…   .CVRCGD     =
"TEMP_R"  "TEMP_R""  EPSI_R" "  EPSI_R"…  .CVRCCMP   =
"TEMP" "TEMP" "EPSXX    " "EPSYY    "…
```