

Structure of data sd_cham_mater

Summary:

One describes the structure of data here `sd_cham_mater` (produced by the order `AFFE_MATERIAU`).

One also describes the structure of data `sd_cham_mater_code` who is a temporary structure of data used in the orders of calculation.

Contents

1 General information.....	3
2 Tree structures.....	3
3 Contents of the Jeveux objects.....	3
3.1 sd_cham_mater.....	3
3.2 sd_cham_mater_code.....	4
3.3 sd_cham_mater_varc.....	4

1 General information

The structure of data `sd_cham_mater` is produced by the operator `AFPE_MATERIAU`. It contains essential and obligatory map an "containing the name of materials (`sd_mater`) affected on the meshes of the grid.

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

Structures of data `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 the object.CVRCVARC

3 Contents of the Jveux objects

3.1 `sd_cham_mater`

`.CHAMP_MAT`

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

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

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

Note: particular management of the temperature of reference:

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

.COMPOR

This `sd_carte` contains information of assignment of the keyword factor `AFFE_COMPOR` (routine `rccomp.f`).
The size associated with this map is `COMPOR`.
The 7 components used are: `RELCOM`, `NBVARI`, `DEFORM`, `INCELA`, `C_PLAN`, `XXXX1` and `XXXX2`

3.2 `sd_cham_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 map of coded materials are addresses of `sd_mater_code` instead of names of `sd_mater`.

Notice :

This `sd_carte` is created on the Volatile basis at the beginning of the operators of calculation (routine `rcmfmc.f`). As it contains addresses `JEVEUX`, it cannot have an unlimited lifetime.

3.3 `sd_cham_mater_varc`

Vocabulary, definitions

One calls "CVRC" (variable of scalar order) a scalar real variable which influences the mechanical laws of behavior. Examples: temperature, hydration,...

One calls "VARC" (variable of vectorial order) a set of "CVRC" connected between them logically.

Example: metallurgical phases of steel: percentages of ferrite, pearlite, bainite,...

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

Examples:

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

Note:

even if some `CVRC` are connected logically by `VARC`, it is necessary that the names of `CVRC` are all distinct. The reason is that they are often the parameters of some functions of materials (

`sd_mater`). When, for example, a Young modulus is defined like a function of `'PBAINITE'` , this name must have an "absolute" direction.

One will call `nbcvrc` the number of CVRC affected (even partially) in `sd_cham_mater` .
If for example, the 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 them CVRC are not affected on all the model.

Object `.CVRCNOM`

This vector gives the name of all them CVRC affected (same partially) on the model. The order of CVRC in this vector is the order which is also used in the 4 other objects below.

Object `.CVRCVARC`

This vector gives the name of VARC corresponding to 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)`) contains values of reference (`VALE_REF`) affected for the variable of order VARC .

CART2 (`sd_carte (NEUT_K16)`) contains the necessary information to evaluate the variable of order VARC.

This information is a "tuple" of 7 values (`varc`, `tysd`, `nomsd`, `nomsym`, `proldr`, `proлга`, `finst`)

`varc` : name of VARC

`tysd` : type of the affected SD: `'EVOL'` / `'FIELD'`

if `tysd='FIELD'`:

- `nomsd` : name of the field (presumably stationary) affected
- `Nomsym = proldr = proлга = finst = ''`

if `tysd='EVOL'`:

- `nomsd` : name of `sd_evol_xxx` affected
- `nomsym` : reference symbol of the field to be used in the `sd nomsd`
- `proldr` : prolongation "on the right" i.e. beyond the final moment of `evol_xxx` (`'EXCLUDED'` / `'CONSTANT'` / `'LINEAR'` / `'`)
- `proлга` : prolongation "on the left" i.e. in on this side initial moment of `evol_xxx` (`'EXCLUDED'` / `'CONSTANT'` / `'LINEAR'` / `'`)
- `finst` : name of `sd_fonction` (or `sd_formule`) allowing to transform the "time of mechanical calculation" into "time of `evol_xxx`". If `finst=''`, the function "identity" is used.

```
INST_EVOL = finst (INST_CALC)
```

Object .CVRCGD

$V(k)$: name of the size associated with the field (or with `evol_xxxx`) affected for $K^{\text{ème}}$ CVRC.

Object .CVRCCMP

$V(k)$: name of the component of the size associated with the field (or with `evol_xxxx`) affected for $K^{\text{ème}}$ 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'      ...
```