

Structure de donnée `sd_cham_mater`

Résumé :

On décrit ici la structure de données `sd_cham_mater` (produite par la commande `AFPE_MATERIAU`).

On décrit également la structure de données `sd_cham_mater_code` qui est une structure de données temporaire utilisée dans les commandes de calcul.

Table des Matières

1 Généralités.....	3
2 Arborescences.....	3
3 Contenu des objets Jeveux.....	3
3.1 sd_cham_mater.....	3
3.2 sd_cham_mater_code.....	4
3.3 sd_cham_mater_varc.....	4

1 Généralités

La structure de donnée `sd_cham_mater` est produite par l'opérateur `AFFE_MATERIAU`. Elle contient une carte "essentielle" et obligatoire contenant le nom des matériaux (`sd_mater`) affectés sur les mailles du maillage.

Pour un accès rapide aux caractéristiques des matériaux dans les routines `te00xx`, on a introduit la notion de "matériau codé" (`sd_mater_code`).

En conséquence, il a fallu créer la `sd_cham_mater_code` qui est une carte dans laquelle les matériaux ont été remplacés par des "matériaux codés".

Les structures de données `sd_mater` et `sd_mater_code` sont décrites dans [D4.06.18]

2 Arborescences

```
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'       :      OBJ   S   V   K8   long=nbcvrc
  (o)   '.CVRCGD'       :      OBJ   S   V   K8   long=nbcvrc
  (o)   '.CVRCVARC'     :      OBJ   S   V   K8   long=nbcvrc
  (o)   '.CVRCCMP'     :      OBJ   S   V   K8   long=nbcvrc
```

+ 2*nbvarc `sd_carte` dont le nom est déduit du contenu de l'objet `.CVRCVARC`

3 Contenu des objets Jveux

3.1 `sd_cham_mater`

`.CHAMP_MAT`

Cette `sd_carte` contient le nom du (ou des) `sd_mater` affecté(s) sur chaque maille du maillage.

Dans le cas général, chaque maille n'est affectée que par un seul `sd_mater`. Mais parfois, il faut indiquer une liste de `sd_mater` (lorsque le comportement mécanique non linéaire est obtenu par la commande `DEFI_COMPOR` [U4.43.06]).

Sur chaque, maille, on peut affecter jusqu'à 28 `sd_mater` différents.

Remarque: gestion particulière de la température de référence :

La transformation `sd_cham_mater` → `sd_cham_mater_code` oblige à garder une correspondance (`sd_mater` → température de référence). La raison en est la transformation de la fonction $\alpha(T)$ à partir des 2 températures `TREF` et `TDEF` (voir routine `alfint.f`). Cette correspondance n'est plus assurée automatiquement depuis que le mot clé `AFFE/TEMP_REF` a été remplacé par `AFFE_VARC/VALE_REF`.
On rétablit cette correspondance à la fin de l'opérateur `AFFE_MATERIAU` (routine `cmtref.f`).
Pour cela, on écrit la température de référence affectée dans la `sd_carte.CHAMP_MAT`.
On stocke la température de référence en écrivant 2 `sd_mater` supplémentaires (et fictifs) :
(`'ACIER'`, `'TREF=>'`, `'20.50'`).
La température est écrite au format `F8.2`.

.COMPOR

Cette `sd_carte` contient les informations d'affectation du mot clé facteur `AFFE_COMPOR` (routine `rccomp.f`).
La grandeur associée à cette carte est `COMPOR`.
Les 7 composantes utilisées sont : `RELCOM`, `NBVARI`, `DEFORM`, `INCELA`, `C_PLAN`, `XXXX1` et `XXXX2`

3.2 `sd_cham_mater_code`

Cette `sd_carte` est une copie de la `sd_carte.CHAMP_MAT`. La différence entre ces 2 `sd_carte` est que les valeurs de la carte des matériaux codés sont des adresses de `sd_mater_code` à la place des noms des `sd_mater`.

Remarque :

Cette `sd_carte` est créée sur la base Volatile en début des opérateurs de calcul (routine `rcmfmc.f`). Comme elle contient des adresses `JEVEUX`, elle ne peut pas avoir une durée de vie illimitée.

3.3 `sd_cham_mater_varc`

Vocabulaire, définitions

On appelle "CVRC" (variable de commande scalaire) une variable réelle scalaire qui influe sur les lois de comportement mécaniques. Exemples : température, hydratation, ...

On appelle "VARC" (variable de commande vectorielle) un ensemble de "CVRC" reliées entre elles logiquement.

Exemple : phases métallurgiques de l'acier : pourcentages de ferrite, de perlite, de bainite, ...

Les `VARC` et `CVRC` sont nommées (K8). Pour simplifier, chaque `CVRC` isolée est rattachée à une `VARC` de même nom. L'accès à une variable de commande scalaire (`CVRC`) se fait donc logiquement en donnant le nom de la `VARC` et le nom de la `CVRC`.

Exemples :

`VARC='TEMP'` `CVRC='TEMP'` => température (`CVRC` isolée)
`VARC='M_ACIER'` `CVRC='PBAINITE'` => proportion de bainite pour la métallurgie de l'acier

Remarque :

même si certaines `CVRC` sont reliées logiquement par des `VARC`, il est nécessaire que les noms des `CVRC` soient tous distincts. La raison en est qu'elles sont souvent les paramètres de certaines

fonctions des matériaux (sd_mater). Lorsque, par exemple, un module d'Young est défini comme une fonction de 'PBAINITE ', ce nom doit avoir un sens "absolu".

On appellera *nbcvrc* le nombre de CVRC affectées (même partiellement) dans le *sd_cham_mater*. Si par exemple, l'utilisateur a écrit :

```
chmat= AFFE_MATERIAU( ... AFFE_VARC = (  
    _F(NOM_VARC='TEMP', GROUP_MA='GM1', ...)  
    _F(NOM_VARC='M_ACIER', GROUP_MA='GM2', ...)
```

Le nombre de CVRC (*nbcvrc*) vaut 8 (1 pour la VARC 'TEMP' + 7 pour la VARC 'M_ACIER') même si toutes les CVRC ne sont pas affectées sur tout le modèle.

Objet .CVRCNOM

Ce vecteur donne le nom de toutes les CVRC affectées (mêmes partiellement) sur le modèle. L'ordre des CVRC dans ce vecteur est l'ordre qui est aussi utilisé dans les 4 autres objets ci-dessous.

Objet .CVRCVARC

Ce vecteur donne le nom des VARC correspondant aux CVRC .

Pour chaque VARC (de nom *novarc*), il existe 2 cartes nommées :

```
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)) contient les valeurs de référence (VALE_REF) affectées pour la variable de commande VARC .

CART2 (*sd_carte* (NEUT_K16)) contient l'information nécessaire à évaluer la variable de commande VARC.

Cette information est un "tuple" de 7 valeurs (*varc*, *tysd*, *nomsd*, *nomsym*, *proldr*, *prolga*, *finst*)

varc : nom de la VARC

tysd : type de la SD affectée : 'EVOL' / 'CHAMP'

si *tysd*='CHAMP':

- *nomsd* : nom du champ (supposé stationnaire) affecté
- *nomsym* = *proldr* = *prolga* = *finst* = ' '

si *tysd*='EVOL':

- *nomsd* : nom de la *sd_evol_xxx* affectée
- *nomsym* : nom symbolique du champ à utiliser dans la *sd nomsd*
- *proldr* : prolongement "à droite" c'est à dire au delà de l'instant final de l'*evol_xxx* ('EXCLU'/'CONSTANT'/'LINEAIRE'/' ')
- *prolga* : prolongement "à gauche" c'est à dire en deçà de l'instant initial de l'*evol_xxx* ('EXCLU'/'CONSTANT'/'LINEAIRE'/' ')
- *finst* : nom de la *sd_fonction* (ou *sd_formule*) permettant de transformer le "temps du calcul mécanique" en "temps de l'*evol_xxx*". Si *finst*=' ', la fonction "identité" est utilisée.

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

Objet .CVRCGD

$V(k)$: nom de la grandeur associée au champ (ou à l'evol_xxxxx) affecté pour la $k^{\text{ème}}$ CVRC.

Objet .CVRCCMP

$V(k)$: nom de la composante de la grandeur associée au champ (ou à l'evol_xxxxx) affecté pour la $k^{\text{ème}}$ CVRC.

Exemple

On pourrait, par exemple, trouver dans ces 4 objets :

```
.CVRCNOM = 'TEMP'      'SECH'      'EPSXX'      'EPSYY'      ...  
.CVRCVARC= 'TEMP'      'SECH'      'EPSA'      'EPSA'      ...  
.CVRCGD   = 'TEMP_R'   'TEMP_R'   'EPSI_R'   'EPSI_R'   ...  
.CVRCCMP  = 'TEMP'     'TEMP'     'EPSXX'     'EPSYY'     ...
```