

Structure de données sd_matr_elem et sd_vect_elem

Résumé :

Table des matières

1 Les Structures de Données en quelques mots.....	3
2 Arborescence.....	3
3 Contenu des objets JEVEUX.....	3
3.1 Objet '.RERR'.....	3
3.2 Objet '.RELR'.....	4
3.3 Objet '.RELC'.....	4
3.4 Objet '.RECC'.....	4
4 Exemples.....	5
4.1 sd_matr_elem.....	5
4.2 sd_vect_elem.....	5

1 Les Structures de Données en quelques mots

Un `sd_matr_elem` représente un ensemble de matrices élémentaires.
Un `sd_vect_elem` représente un ensemble de vecteurs élémentaires.

On peut "assembler" des `sd_matr_elem` pour obtenir une `sd_matr_asse`.
On peut "assembler" des `sd_vect_elem` pour obtenir une `sd_cham_no`.

Pratiquement, les `sd_matr_elem` et les `sd_vect_elem` sont constitués d'un ensemble de `sd_resuelem` [D4.06.05]. Un `sd_resuelem` étant l'ensemble des matrices (ou vecteurs) élémentaires correspondants aux éléments d'un `sd_ligrel`.

Un `sd_matr_elem` (ou un `sd_vect_elem`) peut ne contenir aucun `sd_resuelem`. Cela peut arriver si le `sd_modele` ne contient que des sous-structures statiques.

2 Arborescence

```
sd_matr_elem (K19) ::=record  
  
  (o)   '.RERR' : OJB S V K24 long=5  
  (f)   '.RELR' : OJB S V K24 long=*  
  (f)   '.RECC' : OJB S V K8  long=*
```

```
sd_vect_elem (K19) ::=record  
  
  (o)   '.RERR' : OJB S V K24 long=5  
  (f)   '.RELR' : OJB S V K24 long=*  
  (f)   '.RELC' : OJB XC V I NO
```

3 Contenu des objets JEVEUX

3.1 Objet '.RERR'

`'.RERR'` : S V K24 long=5

Soit $V = \backslash.RERR'$

V(1)	nom du <code>sd_modele</code> sous-jacent
V(2)	nom de la sur-option attachée : <code>'RIGI_MECA'</code> , <code>'MASS_THER'</code> , <code>'CHAR_MECA'</code> , ...
V(3)	/ <code>'OUI_SOUS_STRUC'</code> / <code>'NON_SOUS_STRUC'</code>
V(4)	Nom du <code>sd_cham_mater</code> sous-jacent au <code>sd_matr_elem</code> (ou <code>sd_vect_elem</code>).
V(5)	Nom du <code>sd_cara_elem</code> sous-jacent au <code>sd_matr_elem</code> (ou <code>sd_vect_elem</code>).

$V(3) = /'OUI_SOUS_STRUC' / 'NON_SOUS_STRUC'$: Indique si les termes élémentaires (matrices ou vecteurs) des sous-structures statiques sont à prendre en compte (ou non).

Par exemple, pour un `sd_matr_elem` de type `'RIGI_MECA'` qui ne concernerait que des blocages de ddl par dualisation, il faut ignorer les sous-structures. Sinon, lors de l'assemblage, on risquerait de doubler la rigidité des sous-structures.

Remarque :

'`OUI_SOUS_STRUC`' ne veut pas dire que le modèle possède forcément des sous-structures actives. Mais s'il en possède, elles seront prises en compte.

L'objet `.RERR` est obtenu en appelant la routine `memare.f`.

3.2 Objet '`.RELR`'

'`.RELR`' : S V K24

Cet objet contient la liste des `sd_resuelem` composant le `sd_matr_elem` (ou le `sd_vect_elem`).

Cet objet n'existe pas si le modèle ne contient que des sous-structures (et pas d'éléments finis ordinaires).

Soit $V = \text{'}.RELR'$,

$V(i)$ (1:19) : nom du $i^{\text{ème}}$ `sd_resuelem` du `sd_matr_elem` (ou du `sd_vect_elem`).

Attention :

Pour les `sd_vect_elem`, $V(i)$ est le nom d'1 `sd_cham_no` (et non pas 1 `resuelem`)

Lorsque ce second membre provient d'une charge de type `AFFE_CHAR_MECA/VECT_ASSE`

Remarques :

Le nombre de `sd_resuelem` est obtenu par '`LONUTI`' de l'objet `.RELR`.

L'utilitaire `reajre.f` permet de stocker les `resuelem` dans la `sd_matr_elem` (ou `sd_vect_elem`).

3.3 Objet '`.RELC`'

'`.RELC`' : XC V I NO()

Cet objet n'existe que si le maillage contient des super-maillages. Cette collection est nommée par les cas de charges indiqués par l'utilisateur dans la commande `CALC_VECT_ELEM`. Tous les objets de cette collection ont la même longueur.

Soit `nomcas` un tel cas de charge,

$V = \text{'}.RELC'$ (`nomcas`).

`LONG (V) = nbmas =` nombre de super-maillages du maillage sous-jacent

pour $i = 1, nbmas$

$V(i)$:

/ 1 si la super-maille i est active pour le chargement `nomcas`

/ 0 si la super-maille i n'est pas active pour le chargement `nomcas`

3.4 Objet '`.RECC`'

'`.RECC`' : S V K8

Cet objet n'existe que si le `sd_matr_elem` a été obtenu par la commande `CALC_MATR_ELEM`. On stocke dans ce vecteur le nom des charges ayant servi au calcul du `sd_matr_elem`. Cet objet est utilisé

lors de l'assemblage du sd_matr_elem afin de retrouver les éventuelles relations linéaires éliminées "cachées" au sein des sd_charge.

```
V = '.RECC'.  
LONG (V) = nbmas = nombre de charges  
V(i) : nom de la ième charge
```

4 Exemples

4.1 sd_matr_elem

```
MATELE_1=CALC_MATR_ELEM( MODELE=MODELE_1, CHARGE=CHARGE_1,  
CARA_ELEM=CARAC_1,  
CHAM_MATER=CH_MAT_1, OPTION='RIGI_MECA' )
```

donne :

```
-----  
IMPRESSION SEGMENT DE VALEURS >MATELE_1 .RELR <  
>>>>>  
1 - >MATELE_1.ME001 <>MATELE_1.ME002 <  
3 - > <  
-----  
IMPRESSION SEGMENT DE VALEURS >MATELE_1 .RERR <  
>>>>>  
1 - >MODELE_1 <>RIGI_MECA <  
3 - >OUI_SOUS_STRUC <>CH_MAT_1.MATE_CODE <  
5 - >CARAC_1 <
```

4.2 sd_vect_elem

```
VECELE_1=CALC_VECT_ELEM( CHARGE=CHARGE_1, OPTION='CHAR_MECA' )
```

donne :

```
-----  
IMPRESSION SEGMENT DE VALEURS >VECELE_1 .RELR <  
>>>>>  
1 - >VECELE_1.VE001 <>VECELE_1.VE002 <  
3 - > <> <  
5 - > <> <  
7 - > <> <  
9 - > <> <  
-----  
IMPRESSION SEGMENT DE VALEURS >VECELE_1 .RERR <  
>>>>>  
1 - >MODELE_1 <>CHAR_MECA <  
3 - >NON_SOUS_STRUC <> <  
5 - > <> <
```