

## 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
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=*
```

```
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  $V = \text{'}.RERR'$

V(1)	nom du <code>sd_modele</code> sous-jacent
V(2)	nom de la sur-option attachée : 'RIGI_MECA', 'MASS_THER', 'CHAR_MECA', ...
V(3)	/ 'OUI_SOUS_STRUC' / 'NON_SOUS_STRUC'
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 ddlis 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 = '.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)` peut être le nom d'un `sd_cham_no` (et non pas d'un `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 = '.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`

## 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 - >                          <                          <
```