

Structures de données `champ_no_s` et `cham_elem_s`

Résumé :

Ce document décrit les structures de données `cham_no_s` et `cham_elem_s`.

On donne également la liste des principaux utilitaires travaillant sur ces structures de données.

On définit deux nouvelles SD : `cham_no_s` et `cham_elem_s` qui contiennent les mêmes informations que les SD `cham_no` et `cham_elem` mais qui sont plus "simples" à manipuler dans le fortran.

Il existe des utilitaires permettant de transformer un `cham_no` en `cham_no_s` (et réciproquement) (de même pour les `cham_elem`).

Ces SD seront donc en général des SD temporaires permettant de travailler plus simplement.

Remarque importante :

Les SD `cham_no_s` et `cham_elem_s` ne sont pas aussi générales que les SD `cham_no` et `cham_elem`. Pour les `cham_no_s`, on ne décrit que les champs portés par les nœuds du maillage (et pas les éventuels nœuds tardifs), Pour les `cham_elem_s`, on ne décrit que les champs portés par les éléments finis dont la maille support est une maille du maillage (et non une maille tardive)

Table des Matières

1	SD <code>cham_no_s</code>	3
1.1	Contenu des OJB.....	3
1.2	Objet '.CNSK'.....	3
1.3	Objet '.CNSD'.....	3
1.4	Objet '.CNCS'.....	3
1.5	Objet '.CNSV'.....	3
1.6	Objet '.CNLS'.....	4
2	SD <code>cham_elem_s</code>	4
2.1	Description de la SD.....	4
2.2	Objet .CESK.....	4
2.3	Objet .CESD.....	5
2.4	Objet .CESC.....	5
2.5	Objets .CESL et .CESV.....	5
2.6	Exemple de boucle sur les valeurs d'un <code>cham_elem_s</code>	6
3	Routines utilitaires.....	6

1 SD `cham_no_s`

```
cham_no_s (K19) ::= record

    ♦ '.CNSK' : OJB S V K8 long = 2
    ♦ '.CNSD' : OJB S V I long = 2
    ♦ '.CNSC' : OJB S V K8 long = nb_CMP
    ♦ '.CNSV' : OJB S V R/C/I/... long = nb_NOEUD*nb_CMP
    ♦ '.CNSL' : OJB S V L long = nb_NOEUD*nb_CMP
```

1.1 Contenu des OJB

Cette SD sert à décrire un champ de grandeurs portées par les nœuds d'un maillage.

1.2 Objet '.CNSK'

'.CNSK' (1)	mailla : nom du maillage sous-jacent au <code>cham_no_s</code> .
'.CNSK' (2)	nomgd : nom de la grandeur associée au <code>cham_no_s</code> ('DEPL_R', 'SIEF_R', ...)

1.3 Objet '.CNSD'

'.CNSD' (1)	nb_NOEUD : nombre de nœuds du maillage sous-jacent.
'.CNSD' (2)	nb_CMP : nombre maximum des CMPS portées par les nœuds.

1.4 Objet '.CNSC'

'.CNSC' (icmp)	cmp_i : nom de la ième CMP du <code>cham_no_s</code>
----------------	--

Remarque :

L'ordre des CMPS dans `.CNSC` peut être quelconque. On n'est pas obligé de respecter l'ordre du catalogue des grandeurs. En revanche, les CMPS doivent faire partie des CMPS de la grandeur `nomgd`.

1.5 Objet '.CNSV'

Cet objet contient les valeurs du `cham_no_s`. Le type de ce vecteur `JEVEUX (R/C/I/K8, ...)` est celui de la grandeur `nomgd`. Sa dimension est `nb_NOEUD*nb_CMP`; c'est-à-dire que tous les nœuds de maillage peuvent porter toutes les CMPS décrites dans `.CNSC`.

On accède à la `ICMP`-ème CMP du `INO`-ème NOEUD par la formule :

$$\text{VALEUR}(\text{INO}, \text{ICMP}) = \text{.CNSV}((\text{INO}-1) * \text{nb_CMP} + \text{ICMP})$$

Remarque :

La présence (ou l'absence) d'une CMP sur un NOEUD est indiquée via l'objet `.CNSL` (voir ci-dessous). Lors de la création d'un `cham_no_s`, ses valeurs non affectées sont en mises à "undef" pour mieux détecter leur usage illicite.

1.6 Objet '.CNLSL'

Cet objet contient des booléens indiquant la présence (ou l'absence) des valeurs du *cham_no_s*. Sa dimension est *nb_NOEUD*nb_CMP*; on s'y déplace de la même façon que dans l'objet *.CNLSV*

On examine la présence de la *ICMP*-ème *CMP* du *INO*-ème *NOEUD* par la formule :

```
EXISTE (INO, ICMP) = .CNLSL ( (INO-1) *nb_CMP + ICMP)
```

2 SD *cham_elem_s*

2.1 Description de la SD

Cette structure de données permet de représenter les valeurs des champs discrétisés sur les mailles d'un maillage.

Plus précisément, l'accès à une valeur réelle (ou complexe, ...) du champ se fait en précisant :

- le numéro de la maille supportant l'élément fini (*IMA*),
- le numéro du point dans la maille (*IPT*),
- le numéro du sous-point dans le point (*ISP*) (*ISP* =1 en général),
- le numéro de la composante de la grandeur associée au champ (*ICMP*),

```
cham_elem_s (K19) ::= record

♦ '.CESK' : OJB S V K8 long = 3
♦ '.CESD' : OJB S V I long = 5 + 4*nb_MAILLE
♦ '.CESC' : OJB S V K8 long = nb_CMP
♦ '.CESV' : OJB S V /R/C/I/... long = nbval
♦ '.CESL' : OJB S V L long = nbval
```

2.2 Objet *.CESK*

<i>.CESK</i> (1)	<i>mailla</i> : nom du maillage sous-jacent au <i>cham_elem_s</i> .
<i>.CESK</i> (2)	<i>nomgd</i> : nom de la grandeur associée au <i>cham_elem_s</i> ('DEPL_R', 'SIEF_R', ...)
<i>.CESK</i> (3)	'ELNO' : champ connu aux nœuds des éléments, 'ELGA' : champ connu aux points de Gauss des éléments, 'ELEM' : champ constant par élément (on dira alors qu'il est connu au centre de gravité)

2.3 Objet .CESD

.CESD (1)	<code>nb_MAILLE</code> : nombre de mailles du maillage sous-jacent.
.CESD (2)	<code>nb_CMP</code> : nombre de CMPS portées par les points. C'est la dimension de l'objet .CESC
.CESD (3)	<code>nbptmx</code> : maximum du nombre de points portés par les mailles
.CESD (4)	<code>nbspm</code> : maximum du nombre de sous-points portés par les points des mailles
.CESD (5)	<code>nucmpmx</code> : numéro d'ordre le plus élevé des CMP possibles du <code>cham_elem_s</code> (dans l'ordre de l'objet .CESC)
.CESD (5+4*(<code>ima</code> -1)+1)	<code>nbpt(ima)</code> : nombre de points de la maille <code>ima</code> .
.CESD (5+4*(<code>ima</code> -1)+2)	<code>nbsp(ima)</code> : nombre de sous-points de la maille <code>ima</code> .
.CESD (5+4*(<code>ima</code> -1)+3)	<code>nbcmp(ima)</code> : numéro maximum des CMPS portées par les sous-points des points de la maille <code>ima</code> .
.CESD (5+4*(<code>ima</code> -1)+4)	<code>IAD(ima)</code> : <code>IAD+1</code> est l'adresse dans les objets .CESL et .CESV de la 1ere CMP du 1er sous-point du 1er point de la maille <code>ima</code> (s'ils existent)

2.4 Objet .CESC

.CESC (<code>icmp</code>)	<code>cmp_i</code> : nom de la <code>i</code> ème CMP du <code>cham_elem_s</code>
-----------------------------	---

Remarque :

L'ordre des CMPS dans .CESC peut être quelconque. On n'est pas obligé de respecter l'ordre du catalogue des grandeurs. En revanche, les CMPS doivent faire partie des CMPS de la grandeur `nomgd` (exception faite de la grandeur `VARI_R`).

2.5 Objets .CESL et .CESV

Ces objets contiennent les valeurs du `cham_elem_s` (.CESV) et des booléens (.CESL) indiquant si ces valeurs ont été affectées (ou si elles sont indéterminées).

Le type `JEVEUX (R/C/I/K8, ...)` de l'objet .CESV est celui de la grandeur `nomgd`.

La dimension de ces 2 vecteurs est `nbval` :

`nbval` est la somme sur toutes les mailles `ima` de `nbpt(ima) * nbsp(ima) * nbcmp(ima)`

pour accéder à la `ICMP` -ème CMP du `ISP` -ème sous-POINT du `IPT` -ème POINT de la `IMA` -ème MAILLE d'un `cham_elem_s`, on utilise la routine utilitaire `CESEXI` :

```
CALL CESEXI (STOP, JCESD, JCESL, IMA, IPT, ISP, ICMP, IAD)
```

où : `JCESD` et `JCESL` sont les adresses des objets .CESD et .CESL du `cham_elem_s`.
`IAD` est la "sortie" de cette routine.

si $IAD > 0$, cela veut dire que la composante recherchée existe dans le *cham_elem_s*. On peut alors la récupérer par : $VALEUR = ZR(JCESV-1+IAD)$ (si le champ est réel).

si $IAD < 0$, cela veut dire que la composante recherchée a une place possible dans le *cham_elem_s* mais qu'elle pas affectée actuellement. On peut alors affecter une valeur dans le *cham_elem_s* en faisant :

```
ZR(JCESV-1+IAD) = VALEUR  
ZR(JCESL-1+IAD) = .TRUE.
```

si $IAD = 0$, cela veut dire que la composante recherchée n'a pas de place possible dans le *cham_elem_s*. C'est à dire que l'une au moins des conditions suivantes est vérifiée :

```
IMA > nb_MAILLE  
IPT > nbpt(IMA)  
ISP > nbSP(IMA)  
ICMP > nbcmp(IMA)
```

2.6 Exemple de boucle sur les valeurs d'un *cham_elem_s*

```
CALL JEVEUO(CES//'.CESD','L',JCESD)  
CALL JEVEUO(CES//'.CESL','L',JCESL)  
CALL JEVEUO(CES//'.CESV','L',JCESV)  
NBMA = ZI(JCESD-1+1)  
DO 40, IMA = 1, NBMA  
  NBPT = ZI(JCESD-1+5+4* (IMA-1)+1)  
  NBSP = ZI(JCESD-1+5+4* (IMA-1)+2)  
  NBCMP = ZI(JCESD-1+5+4* (IMA-1)+3)  
  DO 30, IPT = 1, NBPT  
    DO 20, ISP = 1, NBSP  
      DO 10, ICMP = 1, NBCMP  
        CALL CESEXI(STOP, JCESD, JCESL, IMA, IPT, ISP, ICMP, IAD)  
        IF (IAD.GT.0) VALEUR = ZR(JCESV-1+IAD)
```

3 Routines utilitaires

CARCES	transformer une carte en un <i>cham_elem_s</i>
CELCES	transformer un <i>cham_elem</i> en <i>cham_elem</i>
CESCES	changer la discrétisation d'un <i>cham_elem</i> (ELNO/CART/ELGA)
CESCNS	transformer un <i>cham_elem_s</i> en un <i>cham_no_s</i>
CESCRE	créer un <i>cham_elem_s</i>
CESEXI	tester l'existence d'une CMP d'un point d'une maille d'un <i>cham_elem_s</i>
CESRED	"réduire" un <i>cham_elem_s</i> sur une liste de mailles et/ou une liste de CMPS.
CESTAS	"retasser" le contenu d'un <i>cham_elem_s</i>
CNOCNS	transformer un <i>cham_no</i> en <i>cham_no_s</i>
CNSCES	transformer un <i>cham_no_s</i> en <i>cham_elem</i>
CNSCNO	transformer un <i>cham_no_s</i> en <i>cham_no</i>
CNSCRE	créer un <i>cham_no_s</i>
CNSPRJ	projeter un <i>cham_no_s</i> sur un autre maillage
CNSRED	"réduire" un <i>cham_no_s</i> sur une liste de nœuds et/ou une liste de CMPS.

COISD	copier un cham_no_s ou un cham_elem_s
DETRSD	détruire un cham_no_s ou un cham_elem_s
IMPRSD	imprimer sur listing un cham_no_s ou un cham_elem_s