

Structure de Données sd_cata_elem

Résumé :

Table des matières

1 Généralités.....	3
2 Arborescences.....	4
3 Notations, dimensions.....	5
4 sd_cata_com_libr : '&CATA.CL'.....	5
4.1 Objet .COMLIBR.....	5
5 SD sd_cata_grandeur : '&CATA.GD'.....	5
5.1 Objet .NOMGD.....	5
5.2 Objet .NOMCMP.....	5
5.3 Objet TYPEGD : V(K8).....	5
5.4 Objet .DESCRIGD : collection contiguë de V(I) de longueur 7.....	6
6 SD sd_cata_type_maille : '&CATA.TM'.....	6
7 sd_cata_option : '&CATA.OP'.....	7
7.1 Objet .NOMOPT.....	7
7.2 Objet .DESCOPT : Collection contiguë de V(I).....	7
7.2.1 Objet .OPTPARA : Collection contiguë de V(K8).....	8
8 SD sd_cata_type_elem : '&CATA.TE'.....	8
8.1 Dimensions.....	8
8.1.1 Objet .NBLIGCOL : vecteur d'entiers de longueur 6 : V.....	8
8.2 Nom, TYPE_MAILLE, dimension géométrique, familles d'intégration des TYPE_ELEMENT.....	9
8.2.1 Objet .PNLOCFPG.....	9
8.2.2 Objet .NOLOCFPG.....	9
8.2.3 Objet .DIM_GEOM : vecteur (I) de longueur nb_te : V.....	10
8.3 Familles de Points de Gauss "listes".....	10
8.3.1 Objet .NOFPG_LISTE : OJB S N K24.....	10
8.3.2 Objet .FPG_LISTE : OJB XC V K8 NU().....	10
8.4 Modes locaux.....	10
8.4.1 Objet .NOMMOLOC.....	11
8.4.2 Objet .MODELOC.....	11
8.4.3 Objet .TAILLMAX : vecteur (I) de longueur nb_te : V.....	12
8.5 Options calculées par les type_element.....	12
8.5.1 Objet .OPTTE : Objet simple V(I).....	12
8.5.2 Objet .OPTMOD : Collection contiguë de V(I).....	12
8.5.3 Objet .OPTNOM : Collection contiguë de V(K8).....	13
8.6 Objet '.CTE_ATTR' : Collection de V(K16) de longueur nb_te.....	13
9 sd_cata_phen_mode : '&CATA'.....	13
9.1 Objet .PHENOMENE : S N K16.....	13
9.2 Objets .MODL.....	13
9.3 Autres objets.....	14

1 Généralités

La structure de données `sd_cata_elem` regroupe toutes les informations fournies dans les fichiers de catalogues d'éléments finis [D3.02.01].

Cette SD est créée par la procédure de mise à jour du code `MAJNEW` et sauvegardée dans la base éléments. Cette base est recopiée dans la base de l'utilisateur lors de la commande `DEBUT`. Les objets qui composent cette SD sont alors accessibles en lecture par tous les opérateurs du code.

Il n'existe qu'une seule SD de type `sd_cata_elem`; son nom est '&CATA'.

La `sd_cata_grandeur` contient les informations du catalogue `compelem/grandeur_simple__.cata`

La `sd_cata_type_maille` contient les informations du catalogue `compelem/type_maille__.cata`

La `sd_cata_option` contient les informations des catalogues `option/*.cata`

La `sd_cata_type_elem` contient les informations des catalogues `typelem/*.cata`

La `sd_cata_phen_mode` contient les informations du catalogue `compelem/phenomene_modelisation__.cata`

Remarque :

Tous les objets décrits dans ce document (sauf les 4 objets `&CATA.TE.DIM_GEOM`, `&CATA.TE.OPTTE`, `&CATA.TE.TAILLMAX` et `&CATA.TE.NBLIGCOL`) sont créés par les scripts python de `Lecture_Cata_Ele/*.py`. Ces scripts génèrent un fichier ASCII contenant ces objets qui sont ensuite relus par la routine FORTRAN `lccata.f`. Cette routine calcule alors les 4 objets manquants.

2 Arborescences

```
sd_cata_elem (K5) ::=record
  ◆ '.CL'      : sd_cata_com_libr
  ◆ '.GD'      : sd_cata_grandeur
  ◆ '.TM'      : sd_cata_type_maille
  ◆ '.OP'      : sd_cata_option
  ◆ '.TE'      : sd_cata_type_elem
  ◆ '$VIDE'    : sd_cata_phen_mode

sd_cata_com_libr (K8) ::=record
  ◆ '.COMLIBR' : OJB XC V K80 NU LONG=1

sd_cata_grandeur (K8) ::=record
  ◆ '.DESCRIGD' : OJB XC V I NO LONG=7
  ◆ '.NOMCMP'   : OJB XC V K8 NO
  ◆ '.NOMGD'    : OJB S N K8
  ◆ '.TYPEGD'   : OJB S V K8

sd_cata_type_maille (K8) ::=record
  ◆ '.NBNO'     : OJB XC V I NO() LONG=1 NBOBJ= nb_tm
  ◆ '.NOMTM'    : OJB S N K8 LONG= nb_tm
  ◆ '.TMDIM'    : OJB S V I LONG= nb_tm
  ◆ '.NOELRF'   : OJB S N K8 LONG= nb_elrefe
  ◆ '.NOFPG'    : OJB S N K16 LONG= nb_fam_pg
  ◆ '.TMELRF'   : OJB S V I LONG= nb_elrefe
  ◆ '.TMFPG'    : OJB S V I LONG= nb_fam_pg

sd_cata_option (K8) ::= record
  ◆ '.DESCOPT'  : OJB XC V I NO
  ◆ '.NOMOPT'   : OJB S N K16
  ◆ '.OPTPARA'  : OJB XC V K8 NO

sd_cata_type_elem (K8) ::= record
  ◆ '.DIM_GEOM' : OJB S V I
  ◆ '.MODELOC'  : OJB XC V I NO
  ◆ '.NBLIGCOL' : OJB S V I
  ◆ '.NOMMOLOC' : OJB S N K24
  ◆ '.NOMTE'    : OJB S N K16
  ◆ '.OPTMOD'   : OJB XC V I NU
  ◆ '.OPTNOM'   : OJB XC V K8 NU
  ◆ '.OPTTE'    : OJB S V I
  ◆ '.TAILLMAX' : OJB S V I
  ◆ '.TYPEPMA'  : OJB S V K8
  ◆ '.NBELREFE' : OJB S V I LONG=2*nb_te
  ◆ '.NOELREFE' : OJB S V K8
  ◆ '.PNLOCFPG' : OJB S V K32 LONG=nb_loc_fpg
  ◆ '.NOLOCFPG' : OJB S V I LONG=nb_loc_fpg
  ◆ '.NOFPG_LISTE' : OJB S N K24
  ◆ '.FPG_LISTE' : OJB XC V K8 NU
  ◆ '.CTE_ATTR'  : OJB S V K16 LONG=2*nb_attributs

sd_cata_phen_mode (K5) ::= record
  ◆ '.PHENOMENE' : OJB S N K16
  ◆ '.ACOUSTIQUE .MODL' : OJB S N K16
  ◆ '.ACOUSTIQUE' : OJB XC V I NO
  ◆ '.MECANIQUE .MODL' : ...
  ...
```

3 Notations, dimensions

nb_te	nombre de type_element du catalogue
nb_tm	nombre de type_maille du catalogue
nb_op	nombre d'option du catalogue
nb_gd	nombre de grandeur du catalogue

4 sd_cata_com_libr : '&CATA.CL'

```
sd_cata_com_libr (K8) ::=record
  ♦ '.COMLIBR' : OJB XC V K80 NU LONG=1
```

4.1 Objet .COMLIBR

Cet objet contient les “commentaires libres” que l’on peut écrire dans certains catalogues entre « blabla .. ». Actuellement, on peut en écrire dans le catalogue `grandeur_simple__` et dans les catalogues d’options.

Un commentaire libre est une suite contigüe de K80 stockée dans l’objet .COMLIBR. Il faut alors stocker (ailleurs !) le nombre de lignes et le numéro de la 1ère ligne du commentaire libre.

5 SD sd_cata_grandeur : '&CATA.GD'

```
sd_cata_grandeur (K8) ::=record
  ♦ '.DESCRIGD' : OJB XC V I NO LONG=7
  ♦ '.NOMCMP' : OJB XC V K8 NO
  ♦ '.NOMGD' : OJB S N K8
  ♦ '.TYPEGD' : OJB S V K8
```

5.1 Objet .NOMGD

Pointeur de nom permettant d’associer à toutes les grandeurs (simple ou élémentaire) un numéro. C’est ce numéro que nous identifierons par la suite avec la grandeur.

Remarque :

| Les collections `.DESCRIGD` et `.NOMCMP` sont numérotées de la même façon que `.NOMGD`.

5.2 Objet .NOMCMP

Collection de $V(K8)$. On y accède par le numéro de la grandeur : `gd`, ou par son nom. Toutes les grandeurs simples ont toutes leurs CMP nommées. On trouve donc en face de `gd`, la liste de tous les noms des CMP de `gd`. Si la grandeur est élémentaire, il n’y a rien en face de `gd`.

5.3 Objet TYPEGD : $V(K8)$

`gd` → $K8$: `type_scalaire(grandeur)` (R, I, C, K8, K16, K24)

5.4 Objet .DESCRIGD : collection contiguë de $V(I)$ de longueur 7

gd → $V(I)$: descripteur de la grandeur gd.

$V(1)$: code_gd

- 1 : grandeur simple
- 3 : grandeur élémentaire (vecteur)
- 4 : grandeur élémentaire (matrice_sym)
- 5 : grandeur élémentaire (matrice_rectangle)

$V(3)$: n_ec : nombre d'entier_codés nécessaires pour décrire les CMP de la grandeur.
 $V(4)$: gd_ligne : grandeur "ligne" pour les grandeurs élémentaires "vecteur" et "matrice".
 $V(5)$: gd_colonne : grandeur "colonne" pour les grandeurs élémentaires "matrice".
 $V(6)$: nblcom : nombre de lignes du commentaire libre associé à la grandeur gd
 $V(7)$: indcom : indice dans '&CATA.CL.COMLIBR' de la 1ère ligne du commentaire libre associé à la grandeur gd

6 SD *sd_cata_type_maille* : '&CATA.TM'

Ce catalogue contient les informations contenues dans le catalogue *type_maille__cata*

Soit :

nb_tm : nombre de type_maille
nb_elrefe : nombre d'ELREFE
nb_fam_pg : nombre de familles de points de Gauss

```
sd_cata_type_maille (K8) ::=record
  ◆ '.NBNO'      : OJB XC V I NO      NBOBJ = nb_tm    LONG =1
  ◆ '.NOMTM'     : OJB S N K8        LONG = nb_tm
  ◆ '.TMDIM'     : OJB S V I         LONG = nb_tm
  ◆ '.NOELRF'    : OJB S N K8        LONG = nb_elrefe
  ◆ '.NOFPG'     : OJB S N K16       LONG = nb_fam_pg
  ◆ '.TMELRF'    : OJB S V I         LONG = nb_elrefe
  ◆ '.TMFPG'     : OJB S V I         LONG = nb_fam_pg
```

.NOMTM : Ce pointeur de nom contient les noms des *type_maille* (K8)
.NOELRF : Ce pointeur de nom contient les noms des ELREFE (K8)
.NOFPG : Ce pointeur de nom contient les noms des familles de points de Gauss.

Le nom d'une famille de points de Gauss (K16) est obtenu en concaténant le nom de l'ELREFE (K8) et le nom de la famille dans cet ELREFE (K8).

Par exemple : 'HE8 FPG1'

.NBNO : NBNO(i_tm) : nombre de nœuds pour le *type_maille* i_tm
.TMDIM : TMDIM(i_tm) : dimension topologique du *type_maille* (0/1/2/3)
.TMELRF : TMELRF(i_elrf) : numéro du *type_maille* associé à l'ELREFE i_elrf.
.TMFPG : TMFPG(i_fpg) : nombre de points de Gauss pour la famille i_fpg.

7 sd_cata_option : '&CATA.OP'

```
sd_cata_option (K8) ::=record
  ♦ '.NOMOPT'      : OJB S N K16
  ♦ '.DESCOPT'    : OJB XC V I NO
  ♦ '.OPTPARA'    : OJB XC V K8 NO
```

7.1 Objet .NOMOPT

Pointeur de nom (K16) permettant d'associer a toutes les options un numéro que l'on confondra avec l'option : opt.

7.2 Objet .DESCOPT : Collection contiguë de V(I)

opt ---> DESCOPT(opt) = V

La longueur de V est $6+3*(nbin+nbou)$ avec :

nbin : nombre de paramètres " in " de l'option

nbou : nombre de paramètres " out " de l'option

V(1) : 1	inutile
V(2) : nbin	nombre de paramètres 'in'
V(3) : nbou	nombre de paramètres 'out'
V(4) : 1	inutile
V(4+1) :gd(in,1)	grandeur associée au paramètre 'in' 1
V(4+2) :gd(in,2)	grandeur associée au paramètre 'in' 2
...	
V(4+nbin+1) :gd(out,1)	grandeur associée au paramètre 'out' 1
...	
V(4+nbin+nbou) :	grandeur associée au dernier paramètre 'out' gd(out, nbou)
V(4+nbin+nbou+1) : nblcom	Nombre de lignes du commentaire libre général associé à l'option.
V(4+nbin+nbou+2) : indcom	Indice dans '&CATA.CL.COMLIBR' de la 1ère ligne du commentaire libre général associé à l'option
	Viennent ensuite les commentaires libres associés aux différents paramètres ("in" ou "out") de l'option :
V(6+nbin+nbou+1) : nblcom	Nombre de lignes du commentaire libre associé au 1er paramètre " in "
V(6+nbin+nbou+2) : indcom	Indice dans '&CATA.CL.COMLIBR' de la 1ère ligne du commentaire libre associé au 1er paramètre " in "
...	
V(6+3*(nbin+nbou)-1) : nblcom	Nombre de lignes du commentaire libre associé au dernier paramètre " out "
V(6+3*(nbin+nbou)) : indcom	Indice dans '&CATA.CL.COMLIBR' de la 1ère ligne du commentaire libre associé au dernier paramètre " out "

7.2.1 Objet .OPTPARA : Collection contiguë de V(K8)

opt -> OPTPARA(opt) = V

V(1) : nom_para(in,1)	nom du paramètre 'in' numéro 1
V(2) : nom_para(in,2)	nom du paramètre 'in' numéro 2
...	
V(nbin+nbou) : nom_para(out,nbout)	nom du dernier paramètre 'out'

8 SD sd_cata_type_elem : '&CATA.TE'

```
sd_cata_type_elem      (K8)      ::=record
  ♦'.DIM_GEOM'          : OJB S V I
  ♦'.MODELOC'          : OJB XC V I          NO
  ♦'.NBLIGCOL'         : OJB S V I
  ♦'.NOMMOLOC'         : OJB S N K24
  ♦'.NOMTE'            : OJB S N K16
  ♦'.OPTMOD'           : OJB XC V I          NU
  ♦'.OPTNOM'           : OJB XC V K8        NU
  ♦'.OPTTE'            : OJB S V I
  ♦'.TAILLMAX'         : OJB S V I
  ♦'.TYPEMA'           : OJB S V K8
  ♦'.NBELREFE'         : OJB S V I          LONG=2*nb_te
  ♦'.NOELREFE'         : OJB S V K8
  ♦'.PNLOCFPG'         : OJB S V K32        LONG=nb_loc_fpg
  ♦'.NOLOCFPG'         : OJB S V I          LONG=nb_loc_fpg
  ♦'.NOFPG_LISTE'      : OJB S N K24
  ♦'.FPG_LISTE'        : OJB XC V K8        NU
  ♦'.CTE_ATTR'         : OJB S V K16        LONG=2*nb_attributs
```

8.1 Dimensions

8.1.1 Objet .NBLIGCOL : vecteur d'entiers de longueur 6 : V

V(1)	nb_op : nombre d'options
V(2)	nb_te : nombre de type_element
V(3)	nb_te : nombre de type_element
V(4)	nb_gd : nombre de grandeurs
V(5)	nb_te : nombre de type_element
V(6)	nb_gd : nombre de grandeurs

8.2 Nom, TYPE_MAILLE, dimension géométrique, familles d'intégration des TYPE_ELEMENT

.NOMTE : Pointeur de nom permettant d'associer à un type_element un numéro (de 1 à n) qui permet de l'identifier : te.

.TYPEMA : vecteur (K8) de longueur nb_te : V

V(te) : nom du type_maille associé au type_element.

.NBELREFE : vecteur (I) de longueur 2*nb_te : V

|V(2*(te-1)+1) | nombre d'ELREFE pour le type_element te|

|V(2*(te-1)+2) | adresse dans .NOELREFE du 1er ELREFE pour le type_element te|

.NOELREFE : vecteur (K8) : V

V(.NBELREFE(2*(te-1)+2+k-1)) : nom du kième ELREFE du type_element te.

8.2.1 Objet .PNLOCFPG

Pointeur de nom permettant d'associer à une "famille locale de points de Gauss" un numéro qui servira d'indice dans l'objet '&CATA.TE.NOLOCFPG'.

Une "famille locale de points de Gauss" est identifiée par un nom (K32) obtenu en concaténant : le nom du type_element (K16), le nom de l'ELREFE (K8) et le nom de la famille (K8).

Par exemple :

```
ENTETE__ELEMENT__THER_PENTA6_D MAILLE__PENTA6
ELREFE__PE6 GAUSS__RIGI=FPG1
```

La "famille locale de points de Gauss" s'appellera : 'THER_PENTA6_D PE6 RIGI'

Attention :

les pointeurs de noms JEVEUX étant limités à K24, l'objet .PNLOCFPG n'est pas un vrai pointeur de noms. Il s'agit simplement d'un vecteur de K32. Pour faire l'équivalent du JENUNO, il faut parcourir le vecteur jusqu'à trouver le nom cherché. L'indice du nom dans le vecteur est le numéro cherché.

Certaines familles sont "simples" (ici : RIGI) d'autres sont des "listes" (voir paragraphe ci-dessous).

8.2.2 Objet .NOLOCFPG

Vecteur d'entiers permettant de "pointer" vers les les objets .TM.NOFPFG et .TM.TMFPFG

Pour une famille "simple"	.NOLOCFPG > 0
Pour une famille "liste"	.NOLOCFPG = 0

En résumé, l'utilisation des objets .PNLOCFPG et .NOLOCFPG se fera dans le FORTRAN (pour une famille "simple") par :

```
NOFLPG=TYPELE//ELREFE//FAMILL (nom "local" d'une famille de PG (K32))
NUFLPG=INDK32 (' &CATA.TE.PNLOCFPG ', NOFLPG)
NUFGPG=&CATA.TE.NOLOCFPG (NUFLPG)
NOFGPG=&CATA.TM.NOFPFG (NUFGPG) (nom "global" de la famille (K16))
```

NBPOIN=&CATA.TM.TMFPG (NUFGPG) (nombre de points de la famille)

8.2.3 Objet `.DIM_GEOM` : vecteur (I) de longueur `nb_te` : V

V(te) : dimension géométrique associée au `type_element`

/ 0	le <code>type_element</code> ne connaît pas la grandeur <code>GEOM_R</code>
/ 1	le <code>type_element</code> connaît la CMP <code>DX</code> de la grandeur <code>GEOM_R</code>
/ 2	le <code>type_element</code> connaît la CMP <code>DY</code> de la grandeur <code>GEOM_R</code>
/ 3	le <code>type_element</code> connaît la CMP <code>DZ</code> de la grandeur <code>GEOM_R</code>

8.3 Familles de Points de Gauss "listes"

On peut définir dans les catalogues de `type_element` des familles qui sont des listes de familles existantes ("simples").

Par exemple :

```
ENTETE__ ELEMENT__ XXXXXX          MAILLE__ HEXA20
      ELREFE__  H20 GAUSS__  RIGI=FPG27  MASS=FPG8  FPG_LISTE__ MATER=(RIGI,
MASS)
```

Pour le `type_element` `XXXXXX`, la famille appelée `MATER` est une famille de 35 points (27+8). Le 3ème point de `RIGI` est le 3ème point de `MATER`. Le 3 point de `MASS` est le 30ème point de `MATER`. On stocke ces informations dans les 2 objets suivants :

8.3.1 Objet `.NOFPG_LISTE` : OJB S N K24

C'est un pointeur de noms permettant de pointer dans le 2ème objet (`.FPG_LISTE`)

Le nom d'une famille "liste" (`NOFPGL2`) est un K24 :

`NOFPGL2=NOMTE(1:16)//NOFPGL(1:8)` si `NOFPGL` est le nom donné à la famille "liste" (`MATER` dans notre exemple).

```
.NOFPG_LISTE(NOFPGL2) → KFPGL
```

8.3.2 Objet `.FPG_LISTE` : OJB XC V K8 NU()

L'accès à cette collection se fait grâce à l'objet précédent (`.NOFPG_LISTE`).

```
.FPG_LISTE(KFPGL) = V(K8)
```

Ce vecteur de K8 est dimensionné à `nb_fam + 1`

V(ifam) : nom de la famille ifam de la liste.

V(nb_fam + 1) : nom de l'elrefe.

Pour notre exemple : `V = ('RIGI', 'MASS', 'H20')`

8.4 Modes locaux

Les modes locaux de tous les `type_element` sont identifiés par un entier : `moloc`. Cet entier est unique pour chaque couple (`type_element`, définition de mode local)

8.4.1 Objet `.NOMMOLOC`

Pointeur de nom. (K24)

A chaque nom composé : `nom_te nom_mode` on peut associer un numéro : `moloc`.
ex : 'DKT' 'NGEOMER' <-> 67.

`moloc` varie de 1 à `nb_mode_locaux` (total sur tous les `type_element`). `moloc` sert de pointeur d'accès à la collection `.MODELOC`

8.4.2 Objet `.MODELOC`

Collection contiguë de `V(I)`.

`moloc` -> `V(I)`

`V(1)` : code

1	ELEM__
2	ELNO__
3	ELGA__
4	VECTEUR__
5	MATRICE__

`V(3)` : `gd`

grandeur associée au `mode_local`

`V(3)` : `nb_scal`

nombre de scalaires (I,R,..) représentant le mode local (i.e. longueur du champ local).

Si code = EL.. :__

`V(4)` : `nb_pt`

`nb_pt` est le nombre de points de localisation du champ sur l'élément:

- pour 1 mode local de type `ELEM__`, `nb_pt` = 1,
- pour 1 mode local de type `ELNO__`, `nb_pt` est le nombre de nœuds de l'élément,
- pour 1 mode local de type `ELGA__`, `nb_pt` est le nombre de points de Gauss de l'élément.

On ajoute 10000 à la valeur absolue de `nb_pt` pour indiquer éventuellement que les différents points du champ n'ont pas la même représentation (`ELNO__` / `DIFF__`).

Si `ELNO__` / `DIFF__` : `V(4+1)` début du `descripteur_grandeur` du point 1

`V(4+n_ec*(i-1) +1)` début du `descripteur_grandeur` du point `i` Sinon : `V(4+1)` et la suite sont le `descripteur_grandeur` (`gd`).

si code = `ELGA__` :

- / `V(4+n_ec+1)` : +NUFGPG si cette famille est "simple".
- / `V(4+n_ec+1)` : -KFPGL si cette famille est "liste".

NUFGPG est le numéro de la famille "simple" associée au mode_local. Pointeur dans l'objet '&CATA.TM.NOFPG'.

KFPGL est le numéro de la famille "liste" associée au mode_local. Pointeur dans l'objet '&CATA.TE.FPG_LISTE'.

Si code = VECTEUR__ ou MATRICE__

V(4) : moloc (ligne)

Si code = MATRICE__

V(5) : moloc (colonne)

8.4.3 Objet .TAILLMAX : vecteur (I) de longueur nb_te : V

V(te) : Max (.MODELOC(3)) pour tous les modes locaux du type_element te

8.5 Options calculées par les type_element

8.5.1 Objet .OPTTE : Objet simple V(I)

V((te-1)*nb_op+op) → i_optte : numéro du optte (OPTion-Type_Element) associé au CALCUL(opt,te).

Ce numéro i_optte sert à pointer dans les collections .OPTMOD et .OPTNOM.

8.5.2 Objet .OPTMOD : Collection contiguë de V(I)

Cette collection décrit les modes locaux des options élémentaires.

i_optte → V(I)

V(1)	num_calc	numéro du calcul élémentaire
V(2)	nbin	nombre de paramètre "in"
V(3)	nbout	nombre de paramètre "out"
V(3+1)	moloc_in_1	mode local du premier paramètre "in"
V(3+2)	moloc_in_2	mode local du deuxième paramètre "in"
...		
V(3+nbin+1)	moloc_ou_1	mode local du premier paramètre "out"
...		
V(3+nbin+nbou)	moloc_ou_nbout	mode local du dernier paramètre "out"

8.5.3 Objet `.OPTNOM` : Collection contiguë de `V(K8)`

Cette collection décrit les noms de paramètres des options élémentaires.

`i_optte` -> `V(K8)`

<code>V(1)</code>	<code>nom_para(in,1)</code>
<code>...</code>	
<code>V(nbin+1)</code>	<code>nom_para (out,1)</code>
<code>...</code>	
<code>V(nbin+nbou)</code>	<code>nom_para(out,nbout)</code>

8.6 Objet `' .CTE_ATTR'` : Collection de `v(K16)` de longueur `nb_te`

Cette collection contient les attributs de tous les `type_element`.

`.CTE_ATTR(te)` : `V(K16)` `LONG=2*nb_attribut`

`V(2*(iattr-1)+1)` : nom de l'attribut de numéro `iattr`
`V(2*(iattr-1)+2)` : valeur de l'attribut de numéro `iattr`

Remarque :

Pour trouver la valeur d'un attribut de nom `nom_attr`, on doit parcourir ce vecteur jusqu'à trouver ce nom à un indice impair.

9 `sd_cata_phen_mode` : `' &CATA'`

```
sd_cata_phen_mode (K5) ::=record
♦ '.PHENOMENE' : OBJ S N K16
♦ '.ACOUSTIQUE .MODL' : OBJ S N K16
♦ '.ACOUSTIQUE' : OBJ XC V I NO
♦ '.MECANIQUE .MODL' : OBJ S N K16
♦ '.MECANIQUE' : OBJ XC V I NO
♦ '.THERMIQUE .MODL' : OBJ S N K16
♦ '.THERMIQUE' : OBJ XC V I NO
```

9.1 Objet `.PHENOMENE` : `S N K16`

Ce pointeur de noms contient tous les noms de phénomène lus dans le catalogue :

Aujourd'hui :

- `'MECANIQUE'`
- `'THERMIQUE'`
- `'ACOUSTIQUE'`

Remarque :

Il ne sert pas à pointer dans une collection.

9.2 Objets `.MODL`

`'ACOUSTIQUE .MODL'` : Noms des modélisations du phénomène `ACOUSTIQUE`.
`'MECANIQUE .MODL'` : Noms des modélisations du phénomène `MECANIQUE`.

'THERMIQUE .MODL' : Noms des modélisations du phénomène THERMIQUE.

9.3 Autres objets

Les autres objets de la structure de données CATA_PHEN_MODE ne sont pas "suffixés" "en dur" dans la documentation. On crée autant d'objets supplémentaires que de phénomènes lus. Ces objets ont pour noms complets :

```
'&CATA.'//nom_de_phenomene
```

Prenons l'exemple de :

```
'.MECANIQUE' : OJB XC V I NO LONG= nb_tm + 2
```

C'est une collection de $V(I)$, nommée par les modélisations possibles pour ce phénomène. A une modélisation donnée, correspond un vecteur d'entiers V .

Pour i_{tm} de 1 à nb_{tm} :

$V(i_{tm})$: numéro du type_élément associé au type maille i_{tm} , pour la modélisation.

Si $V(i_{tm})=0$: le type_maille i_{tm} n'a pas de type_element associé pour la modélisation.

$V(nb_{tm} + 1)$: dimension topologique des éléments "principaux" de la modélisation : 0/1/2/3

$V(nb_{tm} + 2)$: dimension de l'espace physique baignant la modélisation : 2/3