Structure of Data sd_cata_elem

Summary:

Description of the structure of data sd_cata_elem obtained starting from the catalogues of elements and stored in Jeveux objects.

Contents

1 General information

The structure of data sd_cata_elem gather all the furnished information in the files of catalogues of finite elements [D3.02.01].

Ittte SD is Créée by the procedure of update of the code MAJNEW and saved in the base elements. This base is recopied in the base of the user during the ordering BEGINNING. The objects which compose this SD are then accessible in reading by all the operators from the code.

There exists only one SD of the type sd_cata_elem ; its name is '&CATASTROPHES'.

sd_cata_grandor contains information of the cataloford Commons/Physical_quantities.py

sd_cata_type_maille contains information DU catalogues Commons/mesh_types.py

sd_cata_option contains information of the catalogues Options/*.py

sd_cata_type_elem contains information of the catalogues Elements/*.py

sd_cata_phen_mode contains information of the catalogue Commons/phenomenons_modelisations.py

Note:

All objects described in this document (except the 4 objects &CATA.TE.DIM_GEOM, &CATA.TE.OPTTE, &CATA.TE.TAILLMAX and &CATA.TE.NBLIGCOL) during the "compilation" of the catalogues of elements . These scripts generate a file ASCII containing these objects which are then read again by routine FORTRAN lccata. F90 . This routine calculates the 4 missing objects then.

2 Tree structures

```
sd_cata_elem  (K5)  :: =record
    ♦ '.CL'     :     sd_cata_com_libr
    ♦ '.GD'     :     sd_cata_grandor
    ♦ '.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  NAKED K80  LONG=1

sd_cata_grandor (K8)  :: =record
    ♦ '.DESCRIGD':  OJB XC V  I  NO   LONG=7
    ♦ '.NOMCMP'   :  OJB XC V  K8 NO
    ♦ '.NOMGD'    :  OJB S  NR  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  NR  K8       LONG= nb_tm
    ♦ '.TMDIM'    :  OJB S  V  I        LONG= nb_tm
    ♦ '.NOELRF'   :  OJB S  NR  K8       LONG= nb_elrefe
    ♦ '.NOFPG'    :  OJB S  NR  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  NR  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  NR  K24
    ♦ '.NOMTE'       :  OJB S  NR  K16
    ♦ '.OPTMOD'      :  OJB XC V  I    NAKED
    ♦ '.OPTNOM'      :  OJB XC V  K8   NAKED
    ♦ '.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  NR  K24
    ♦ '.FPG_LISTE'   :  OJB XC V  K8   NAKED
    ♦ '.CTE_ATTR'    :  OJB S  V  K16  LONG=2*nb_attributs

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

3 Notations, dimensions

nb_te	number of type_elemenT of the catalogue
nb_tm	number of type_maille catalogue
nb_op	number ofoption catalogue
nb_gd	number of size catalogue

4 SD sd_cata_com_libr : '&CATASTROPHES.CL'

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

4.1 Object .COMLIBR

This object contains the "free comments" which one can write in certain catalogues (comment= "blah…").
Currently, one can write some in the catalogues physical_quantities.py, parameters.py and in the catalogues of options.

A free comment is a contiguous continuation of K80 stored in the object .COMLIBR. It is then necessary to store (elsewhere!) the number of lines and the number of the 1st line of the free comment.

5 SD sd_cata_grandor : '&CATASTROPHES.GD'

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

5.1 Object .NOMGD

Pointer of name allowing to associate with all the sizes (simple or elementary) a number. It is this number which we will identify thereafter with the size.

Notice :
Collections .DESCRIGD and .NOMCMP are numbered in the same way that .NOMGD.

5.2 Object .NOMCMP

Collection of V (K8). One reaches it by the number of the size: Gd, or by its name. All the simple sizes have all theirs CMP named. One thus finds opposite Gd, the list of all the names of CMP of Gd. If the size is elementary, there is nothing opposite Gd.

5.3 Object TYPEGD: V (K8)

Gd − > K8 : type_scalaire (size) (R, I, C, K8, K16, K24)

5.4 Object .DESCRIGD : contiguous collection of V (I) of length 7

Gd — > V (I) : descriptor of the size Gd.

V (1): code_gd

1: simple size
3: elementary size (vector)
4: elementary size (matrice_sym)
5: elementary size (matrice_rectangle)

V (3) : n_ec : many entier_codés necessary to describe them CMP size.
V (4) : gd_ligne : size “line” for the elementary sizes “vector” and “matrix”.
V (5) : gd_colonne : size “column” for the elementary sizes “stamps”.
V (6) : nblcom : many lines of the free comment associated with the size Gd
V (7) : indcom : index in '&CATA.CL.COMLIBR' 1st line of the free comment associated with the size Gd

6 SD sd_cata_type_maille : '&CATASTROPHES.TM'

This catalogue contains the contained information in the catalogue Commons.mesh_types.py

Are:

nb_tm : number of type_maille
nb_elrefe : number ofELREFE
nb_fam_pg : many families of points of Gauss

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

.NOMTM : This pointer of name contains the names of type_maille (K8)
.NOELRF : This pointer of name contains the names of ELREFE (K8)
.NOFPG : This pointer of name contains the names of the families of points of Gauss.

The name of a family of points of Gauss (K16) is obtained by concaténant the name ofELREFE (K8) and the surname in this ELREFE (K8).

For example: 'HE8 FPG1'

.NBNO : NBNO (i_tm) : many nodes for type_maille i_tm
.TMDIM : TMDIM (i_tm) : topological dimension of type_maille (0 2/1/3)
.TMELRF: TMELRF (i_elrf) : number of the type_maille associated withELREFE i_elrf.
.TMFPG : TMFPG (i_fpg) : many points of Gauss for the family i_fpg.

7 sd_cata_option: '&CATASTROPHES.OP'

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

7.1 Object .NOMOPT

Pointer of name (K16) making it possible to associate all the options a number has which one will confuse with the option: opt.

7.2 Object .DESCOPT : Contiguous collection of V (I)

```
opt ---> DESCOPT (opt) = V
```

The length of V is 6+3* (nbin+nbou) with:
nbin : many parameters " in "option
nbou : many parameters " out "option

V (1): 1	useless
V (2): nbin	many parameters 'in'
V (3): nbou	many parameters 'out'
V (4): 1	useless
V (4+1): Gd (in, 1)	size associated with the parameter 'in' 1
V (4+2): Gd (in, 2)	size associated with the parameter 'in' 2
…	
V (4+nbin+1): Gd (out, 1)	size associated with the parameter 'out' 1
…	
V (4+nbin+nbou):	size associated with the last parameter 'out' Gd (out, nbou)
V (4+nbin+nbou+1): nblcom	Many lines of the free comment general associated with the option.
V (4+nbin+nbou+2): indcom	Index in '&CATA.CL.COMLIBR'1st line of the free comment general associated with L' option
	Then the free comments associated with the various parameters come ("in" or "out") of the option:
V (6+nbin+nbou+1): nblcom	Many lines of the free comment associated with the 1st parameter " in "
V (6+nbin+nbou+2): indcom	Index in '&CATA.CL.COMLIBR' 1st line of the free comment associated with the 1st parameter "in"
…	
V (6+3* (nbin+nbou) - 1): nblcom	Many lines of the free comment associated with the last "out" parameter
V (6+3* (nbin+nbou)): indcom	Index in '&CATA.CL.COMLIBR' 1st line of the free comment associated with the last parameter " out "

7.2.1 Object .OPTPARA : Contiguous collection of V (K8)

opt — > OPTPARA (opt) = V

V (1): will nom_para (in, 1)	name of the parameter 'in' number 1
V (2): will nom_para (in, 2)	name of the parameter 'in' number 2
…	
V (nbin+nbou): will nom_para (out, nbout)	name of the last parameter 'out'

8 SD sd_cata_type_elem : '&CATASTROPHES.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  NR  K24
  ♦ ' .NOMTE'             :  OJB S  NR  K16
  ♦ ' .OPTMOD'            :  OJB XC V  I          NAKED
  ♦ ' .OPTNOM'            :  OJB XC V  K8         NAKED
  ♦ ' .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  NR  K24
  ♦ ' .FPG_LISTE'         :  OJB XC V  K8         NAKED
  ♦ ' .CTE_ATTR'          :  OJB S  V  K16        LONG=2*nb_attributs
```

8.1 Dimensions

8.1.1 Object .NBLIGCOL : vector of entireties length 6: V

V (1)	nb_op : number ofoptions
V (2)	nb_te : number of type_element
V (3)	nb_te : number of type_element
V (4)	nb_gd : number of sizes
V (5)	nb_te : number of type_element
V (6)	nb_gd : number of sizes

8.2 Name, TYPE_MAILLE, geometrical dimension, families of integration of TYPE_ELEMENT

.NOMTE : Pointer of name allowing to associate with one type_element a number (of 1 to N) who allows to identify it: you.

.TYPEMA : vector (K8) of length nb_te : V

V (you) : name of type_maille associated with type_element.

.NBELREFE : vector (I) length 2*nb_te : V

|V (2* (you-1) +1) | number ofELREFE for type_element you|
|V (2* (you-1) +2) | address in .NOELREFE 1st ELREFE for type_element you|

.NOELREFE : vector (K8): V

V (.NBELREFE (2* (you-1) +2+k-1)) : name of kth ELREFE type_element you.

8.2.1 Object .PNLOCFPG

Pointer of name allowing to associate to a "local family of points of Gauss" a number which will be used as index in the object '&CATA.TE.NOLOCFPG'.
A "local family of points of Gauss" is identified by a name (K32) obtained while concaténant: the name of type_element (K16), the name ofELREFE (K8) and the surname (K8).

For example:
```
class THER_PENTA6_D (THER_HEXA20):
    …
    elrefe = (
            ElrefeLoc (MT.PE6, gauss = ('RIGI=FPG6', 'FPG1=FPG1',…
```

The "local family of points of Gauss" will be called: 'THER_PENTA6_D PE6 RIGI'

Caution :

pointers of names JEVEUX being limited to K24, the object .PNLOCFPG is not a true pointer of names. It is simply about a vector of K32. To make the equivalent of JENUNO, it is necessary to traverse the vector until finding the name sought. The index of the name in the vector is the sought number.

8.2.2 Object .NOLOCFPG

Vector of entireties allowing "to point" towards the objects .TM.NOFPG and .TM.TMFPG

For a "simple" family	.NOLOCFPG > 0
For a family "lists"	.NOLOCFPG = 0

In short, the use of the objects .PNLOCFPG and .NOLOCFPG will be done in FORTRAN (for a "simple" family) by:

```
NOFLPG=TYPELE//ELREFE//FAMILL  ("local" name of a family of PG (K32))
NUFLPG=INDK32 ('&CATA.TE.PNLOCFPG', NOFLPG)
NUFGPG=&CATA.TE.NOLOCFPG (NUFLPG)
NOFGPG=&CATA.TM.NOFPG (NUFGPG) ("total" name of the family (K16))
NBPOIN=&CATA.TM.TMFPG (NUFGPG)  (many points of the family)
```

8.2.3 Object .DIM_GEOM : vector (l) length nb_te: V

V (you): geometrical dimension associated with type_element

/ 0	type_element does not know the size GEOM_R
/ 1	type_element knows CMP DX size GEOM_R
/ 2	type_element knows CMP DY size GEOM_R
/ 3	type_element knows CMP DZ size GEOM_R

8.3 Family of Points of Gauss “MATER”

One can define in the catalogues of type_element a family of points of Gauss (always invited “to subdue”) who is a list of existing families (“simple”).

For example:

```
ElrefeLoc (MT.TR7, gauss = ('RIGI=FPG3', 'MASS=FPG7', 'FPG1=FPG1',),
           mater= ('RIGI', 'FARMHOUSE', 'FPG1',),),
```

For type_element, the family to subdue is a family of 11 items (3+7+1). The 3rd point of RIGI is the 3rd point of MATER. The 3rd point of FARMHOUSE is the 6th point of to subdue.
One stores this information in the 2 following objects:

8.3.1 Object .NOFPG_LISTE: OJB S NR K24

It is a pointer of names making it possible to point in the 2nd object (.FPG_LISTE)

The name of a family “lists” (NOFPGL2) is one K24 :

NOFPGL2=NOMTE (1:16) //NOFPGL (1:8) if NOFPGL is the name given to the family “lists” (MATER in our example).

```
.NOFPG_LISTE (NOFPGL2) → KFPGL
```

8.3.2 Object .FPG_LISTE: OJB XC V NAKED K8 ()

The access to this collection is done thanks to the preceding object (.NOFPG_LISTE).

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

This vector of K8 is dimensioned with nb_fam +1

V (ifam) : surname ifam list.

V (nb_fam +1) : name ofelrefe.

For our example: V= ('RIGI', 'FARMHOUSE', 'H20')

8.4 Local modes

Local modes of all them type_element are identified by an entirety: moloc. This entirety is single for each couple (type_element, definition of local mode)

8.4.1 Object .NOMMOLOC

Pointer of name. (K24)

With each made up name: nom_te *nom_mode* *one can associate a number:* *moloc .*
ex: ‘*DKT*’‘NGEOMER' <-> 67.

moloc vary from 1 with nb_mode_locaux (total on all them type_element). moloc is used as pointer of access to the collection .MODELOC

8.4.2 Object .MODELOC

Contiguous collection of V (I).

moloc — > V (I)

V (1): code

1	ELEM
2	ELNO
3	ELGA
4	VECTOR
5	MATRIX

V (3): Gd

size associated with mode_local

V (3): nb_scal

number of scalars (I, R.) representing the local mode (i.e length of the local field).

If code = EL. :

V (4): nb_pt

nb_pt is the number of points of localization of the field on the element:

- for 1 local mode of type ELEM, nb_pt = 1,
- for 1 local mode of type ELNO, nb_pt is the number of nodes of the element,
- for 1 local mode of type ELGA, nb_pt is the number of points of Gauss of the element.

One adds 10000 to the absolute value of nb_pt to state if required that the various points of the field do not have the same representation (ELNO/DIFF). It is recognized that a local mode is “ DIFF ” by syntax used to define it.
For example:

```
DDL_MECA = LocatedComponents (phys=PHY.DEPL_R, type=' ELNO', diff=True,
    components= (
    ('EN1', ('DX', 'DY', 'GONF',)),
    ('EN2', ('DX', 'DY',)),
    ('EN3', ('NEAR',)),))
```

If ELNO/DIFF : V (4+1) beginning of the descripteur_grandor of item 1….

V (4+n_ec* (i-1) +1) beginning of the descripteur_grandor of the point I If not: V (4+1) and the continuation are the descripteur_grandor (Gd).

if code = ELGA :

/ V (4+n_ec+1): +NUFGPG if this family is “simple”.
/ V (4+n_ec+1): - KFPGL if this family is “list”.

NUFGPG of the family “simple” partner with the mode_local is the number. Pointer in the object ‘&CATA.TM.NOFPG’.

KFPGL is the number of the family “lists” associated with the mode_local. Pointer in the object ‘&CATA.TE.FPG_LISTE’.

If code = VECTOR or MATRIX

V (4): moloc (line)

If code = MATRIX

V (5): moloc (column)

8.4.3 Object .TAILLMAX : vector (I) of longuor nb_te: V

V (you): Max (.MODELOC (3)) for all the local modes of type_element you

8.5 Options calculated by type_element

8.5.1 Object .OPTTE : Simple object V (I)

V ((you-1) *nb_op+op) - > i_optte : number of optte (OPTion-Type_Element) associated with CALCULATION (opt, you).

This number i_optte is used to point in the collections .OPTMOD and .OPTNOM.

8.5.2 Object .OPTMOD : Contiguous collection of V (I)

This collection describes the local modes of the elementary options.

i_optte - > V (I)

V (1)	num_calc	number of elementary calculation
V (2)	nbin	number of parameter “in”
V (3)	nbout	number of parameter “out”
V (3+1)	moloc_in_1	local mode of the first parameter “in”
V (3+2)	moloc_in_2	local mode of the second parameter “in”
…		
V (3+nbin+1)	moloc_ou_1	local mode of the first parameter “out”

…		
V (3+nbin+nbou)	moloc_ou_nbout	local mode of the last parameter "out"

8.5.3 Object .OPTNOM : Contiguous collection of V (K8)

This collection describes the names of parameters of the elementary options.

i_optte — > V (K8)

V (1)	will nom_para (in, 1)
…	
V (nbin+1)	will nom_para (out, 1)
…	
V (nbin+nbou)	will nom_para (out, nbout)

8.6 Object '.CTE_ATTR': Collection of V (K16) of length nb_te

This collection contains the attributes of all them type_element.

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

V (2* (iattr-1) +1) : name of the attribute of number iattr
V (2* (iattr-1) +2) : value of the attribute of number iattr

Note:

To find the value of an attribute of name nom_attr, one must traverse this vector until finding this name with an odd index.

9 sd_cata_phen_mode: '&CATASTROPHES'

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

9.1 Object .PHENOMENE: S NR K16

This pointer of names contains all the names of phenomenon read in the catalogue:

Today:

- 'MECHANICS'
- 'THERMICS'
- 'ACOUSTICS'
- …

Note:
It is not used to point in a collection.

9.2 Objects .MODL

'.ACOUSTICS .MODL': Names of modelings of the phenomenon ACOUSTICS.
'.MECHANICS .MODL': Names of modelings of the phenomenon MECHANICS.
'.THERMICS .MODL': Names of modelings of the phenomenon THERMICS.

9.3 Other objects

Other objects of the structure of data CATA_PHEN_MODE are not "suffixes" "into hard" in documentation. One creates as many additional objects of phenomena read. These objects have as complete names:

'&CATASTROPHES.'/nom_de_phenomene

Let us take the example of:

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

It is a collection of V (I), named by possible modelings for this phenomenon. With a given modeling, a vector of entireties corresponds V.

For i_tm of 1 with nb_tm:

V (i_tm): number of type_élément associated with the type nets i_tm, for modeling.

If V (i_tm) =0: type_maille i_tm does not have type_element associated for modeling.

V (nb_tm +1): toplogic dimension of the "principal" elements of modeling: 0 2/1/3
V (nb_tm +2): dimension of physical space bathing modeling: 2/3