Data format sd_cata_elem

Summarized:

Contents

1 General information

the data structure sd_cata_elem gathers all the information provided in the files of catalogs of finite elements [D3.02.01].

This SD is created by the procedure of update of code MAJNEW and is saved in the base elements. This base is recopied in the base of the user at the time of the command debut. 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 sd_cata_elem type ; its name is "&CATASTROPHES".

The sd_cata_grandor contains information of the catalog compelem/grandeur_simple__.catastrophes

the sd_cata_type_maille contains information of the catalog compelem/type_maille__.catastrophes

the sd_cata_option contains information of the catalogs option/*.catastrophes

the sd_cata_type_elem contains information of the catalogs typelem/*.catastrophes

the sd_cata_phen_mode contains information of the catalog compelem/phenomene_modelisation__.catastrophes

Note::

All the objects described in this document (except 4 objects &CATA.TE.DIM_GEOM, &CATA.TE.OPTTE, &CATA.TE.TAILLMAX and &CATA.TE.NBLIGCOL) are created by scripts python of Lecture_Cata_ Ele/.py. These scripts generate a file ASCII containing these objects which are then read again by routine FORTRAN lccata.f. This routine calculates the 4 missing objects then . Tree structures sd_cata_elem (K5):: =record ♦ ".CL":*

2 sd_catastrophes_com_libr

```
♦ ".GD"  : sd_  catastrophes_grandor
    ♦ ".TM     "     : sd_cata_type_
    mesh ♦     ".OP     ": sd_cata_option
    ♦ ".     TE"     : sd_cata_type_elem
    ♦ "$VIDE     "     : sd_cata_phen
    _mode sd_     catastrophes     _com_libr (K8):
    : =record ♦   ".COMLIBR     ": OJB XC

V K80 NU LONG=1 sd_cata  _grandor
    ( 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 _catastrophes_type    _maille  (  K8)  ::
    =record ♦ ".NBNO    ": OJB   XC

V I NO () LONG=1 NBOBJ= nb_tm ♦
    ".NOMTM ":     OJB  S N K8  LENGTH    = nb_tm ♦ ".TMDIM":
    OJB S V I    LONG  = nb_  MT  ♦        ".NOELRF":
    OJB S N K8    LENGTH  = 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 ♦ ".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    sd_
    catastrophes _phen_mode (K    5)  :: = record   ♦  ".PHENOMENE": OJB

S N K16 ♦ ".ACOUSTIQUE  .MODL
    ": OJB S N K16          ♦ "  .ACOUSTIQUE
    " : OJB XC V   I NO ♦  ".MECANIQUE    .MODL
    " : …… Notations          ,  dimensions    Nb
    _te number of    type_element  of
     the catalog
```

3 nb_tm number

of the _maille	type of the catalog nb_op number D
"option	of the catalog nb_gd number of quantity
	of the catalog sd_cata _com_libr:
"&CATAS TROPHES	.CL" sd_cata _com_libr (K8):: =record

4 ♦" .COMLIBR ": OJB XC

```
V K80 NU LONG  =1 Objet   .COMLIBR
      This object contains      the "  free     comments
```

4.1 " that L" one

can write in certain catalogs between "blah. ". Currently, one can write some in the catalog grandeur_simple and the catalogs of options. A free comment is a contiguous continuation of K80 stored

in object .COMLIBR. It is then necessary to store (elsewhere!) the number of lines and the number of 1st line of the free comment. SD sd_cata_grandor: "&CATASTROPHES.GD" sd_cata_grandor (K8):

5 : =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 Pointer Object           .NOMGD  of  name
    allowing D          "  to associate  with  all
```

5.1 the quantities

(simple or elementary) a number. C" is this number which we will identify thereafter with the quantity. Note: Collections .DESCRIGD and .NOMCMP are numbered

in the same way

> *that .NOMGD. Object. NOMCMP Collection of V (K8). One reaches it by the number*

5.2 of the quantity

: Gd, or by its name. All the simple quantities have all their named CMP. One thus finds opposite Gd, the list of all the names of the Gd CMP. If the quantity is elementary, there is nothing opposite Gd. Object TYPEGD: V (K8) Gd — > K8: type_scalaire (quantity)

5.3 (R, I, C , K8, K16, K24) Object

.DESCRIGD : contiguous collection of V (I) length

5.4 7 Gd — > V (I) : descriptor of the quantity Gd . V (1): code

_gd 1: simple quantity 3: elementary quantity

(vector) 4

: elementary quantity
(matrice_sym) 5: elementary quantity
(matrice_rectangle) V (3): n_ec
: many entier_codés necessary to describe

the CMP of the quantity. V (4): gd_ligne: quantity “line” for the elementary quantities
“vector ” and “matrix”. V (5): gd_colonne: quantity “column” for the elementary quantities “stamps ”. V (6): nblcom: many lines of the free comment associated
with the quantity Gd V (7): indcom: index in “&CATA.CL.COMLIBR” of 1st line of the free
comment associated with the quantity Gd SD sd _catastrophes_type_maille: “&CATASTROPHES.TM” This catalog contains the contained information

6 in the catalog type_

maille__.catastrophes Is: nb_tm: number of type_maille nb_elrefe: number of ELREFE nb_fam

_pg:

many families of Gauss points
sd _catastrophes _type_maille (
K8):: =record ♦ “.NBNO”: OJB XC V I NO NBOBJ

= nb_tm LONG =1 ♦ “.NOMTM”:
OJB S N K8 LENGTH = nb_tm ♦ “. TMDIM”: OJB S V I
LONG = nb_tm ♦ “.NOELRF ” : OJB S N K8
LENGTH = 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 : This pointer of name
contains the names of the type _maille (K8) .NOELRF

: This pointer of name contains the names of the ELREFE (K8) .NOFPG : This pointer of name contains the names of the families of Gauss points
. The name of a family of Gauss points (K16) is obtained by concaténant

the name of the ELREFE (K8) and the surname in this ELREFE (K8). For example: “HE8 FPG1” .NBNO: NBNO (i_tm): many nodes for

the type_maille i_tm. TMDIM

: TMDIM (i_tm): topological dimension of the type_ mesh (0 2/1/3)
.TMELRF : TMELRF (i_elrf): number of the type_maille associated with the ELREFE I
_elrf. .TMFPG: TMFPG (i_fpg): number of Gauss points for the i_fpg family
. sd_cata _option: “&CATASTROPHES .OP” sd_cata_option (K8):: =record ♦ “. NOMOPT”

7 : OJB S N K16 ♦ ".DESCOPT

```
": OJB XC  V I  NO ♦ ".
    OPTPARA": OJB    XC  V K8  NO  Pointer
    Object .NOMOPT    of name  (K16     )
    making it possible   to associate  has all
```

7.1 the options a number

which one will confuse with the option: opt. Object .DESCOPT: Contiguous collection of V (I) opt ---> DESCOPT (opt) = V

7.2 the length of V is 6+3* (nbin+nbou) with: nbin

: many parameters

" in "option nbou : many parameters
" out "of the 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) quantity	associated with the parameter "
in" 1 V	(4+2): Gd
(in, 2) quantity associated	with the parameter "in" 2… V (4+nbin
+1): Gd (out, 1)	quantity associated with the parameter "out" 1
…	
V (4+nbin+nbou): quantity	associated with the last parameter "out"
Gd (	
out, nbou) V (4+nbin	+nbou+1): nblcom Many lines of the general free comment
associated with the option.	V (4+nbin+nbou+2): indcom Index in "&CATA.CL.COMLIBR" of 1st line of the general
free comment associated	with the option Comes then the free comments associated with the various parameters ("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" of 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 parameter
"	
out "V (6+3* (nbin+nbou)): indcom	Index in "&CATA.CL.COMLIBR" of 1st line of the free comment associated with the last
" out "parameter	Object .OPTPARA: Contiguous collection of V (K8) opt — > OPTPARA (opt) = V V (1): (in, 1) name of the parameter " in

7.2.1 ” number 1 will nom_para V (2): will nom_para (in, 2) name of

the parameter “in” number

2… V (nbin+nbou):	(out, nbout) name of the last “out
” parameter will nom_para SD	sd_cata_type_elem: “&CATASTROPHES.TE” sd_cata
_type	
_elem (K8):: =record ♦ ' .DIM_GEOM	“: OJB S V I ♦” .MODELOC':

8 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_             you  ♦" .NOELREFE
  ": OJB S V          K8  ♦" .PNLOCFPG
  ": OJB S            V  LONG K32   =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
  Dimensions      Object  .NBLIGCOL
  : vector of integers         length  6        :
  V V (1) nb_op: number          D"  options        V (2) nb_te: number
```

8.1 of type_element

8.1.1 V (3) nb_te : number of type_element V (4) nb_gd

: many	quantities V (5) nb_
you:	number of type_element V (6) Nb
_gd:	many quantities Name, TYPE_MAILLE
	, geometrical dimension , families
	of integration of TYPE_ELEMENT
.NOMTE	: Pointer of name allowing

8.2 to associate with type_element a number (of 1 to N) which makes it possible to identify it

: you. .TYPEMA: vector (K8) length nb_te: V V (you): name of the type_maille associated with type_element. .NBELREFE: vector

(I) length 2*nb_te: V |V (2* (you 1) +1

) | number of ELREFE for type_element you| |V (2* (you

- 1) +2) | addresses in .NOELREFE of the 1st ELREFE

for type_element you| .NOELREFE : vector (K8): V V (.NBELREFE (2* (you-1) +2+ k-1)) : name of the kth ELREFE of type_element you . Pointer object

.PNLOCFPG of name allowing

D" of Gauss points to associate to a " local family " a number which

8.2.1 will be used as index

in object "&CATA.TE.NOLOCFPG". A "local family of Gauss points" is identified by a name (K32) obtained while concaténant: the name of type_element (K16),

the name of the ELREFE (K8) and the surname (K8). For example: ENTETE __ELEMENT THER_PENTA6_D NETS of Gauss points __PENTA6 ELREFE _PE6 GAUSS RIGI=FPG1 the "family local

" will be called

: "THER_PENTA6_D PE6 RIGI " Attention:
the pointers of names JEVEUX being

restricted to K24, object .PNLOCFPG is not a true pointer of names. It

is simply

about a vector from K 32. To make the equivalent of the 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. Certain families are "simple here" (: RIGI) of others are "lists" (see paragraph below). Object .NOLOCFPG Vector
of integers making it possible "to point" towards objects .TM.NOFPG and .TM.TMFPG For a "simple"

8.2.2 family .NOLOCFPG

> 0 For a family "lists" .NOLOCFPG = 0 In short, the use of 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) Object .DIM_GEOM: vector (
I) length nb_te: V V (you):
geometrical dimension associated with type_element/the 0 type_ element
does not know quantity GEOM_R/1 type_element

8.2.3 knows the CMP DX from quantity GEOM_R/2

type_element knows the CMP DY quantity GEOM_R/3

	the type _element of Gauss points knows the CMP DZ of quantity
GEOM_R	Familles "lists "One can define in
the catalogs	of type_element families which are lists of
existing	families (" simple") . For example: ENTETE __ELEMENT

8.3 __ MAILLE HEXA20 ELREFE

__H20 GAUSS RIGI=FPG27 MASS=FPG8 FPG _LISTE MATER = (RIGI, FARMHOUSE) For type_element the, family called MATER

is a family

```
of 35 items (27+8). The 3rd      point of RIGI
   is  the 3rd point of  MATER  the 3 point of  FARMHOUSE is the 30ème not
```

MATER One stores this information in the 2 following objects : Object .NOFPG_LISTE: OJB S N K24 It is a pointer of names making it possible to point in 2nd object (.FPG_LISTE) the name of a family
"lists" (NOFPGL2) is one K24: NOFPGL2=NOMTE (

8.3.1 1:16)/ NOFPGL (1:8) if NOFPGL is

the name given to the family "lists" (MATER in our example). .NOFPG_LISTE (NOFPGL

2) → KFPGL Object .FPG_LISTE: OJB XC V K8 NU ( ) 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 to Nb

8.3.2 _fam + 1 V (ifam): surname

ifam of the list. V (nb_fam +1): name of the elrefe. For our example: V= ("RIGI

", "FARMHOUSE", "H20")

local Modes the local modes of all the type

_element are identified by an integer : moloc

. This integer is single for each

couple (type_element, definition of local mode) Pointer Object

8.4 .NOMMOLOC

of name. (K24) A each made up name: name _te nom_mode one can associate a number: moloc. ex: “DKT” “NGEOMER” <-> 67. moloc vary from 1 with nb_mode_locaux

8.4.1 (total on all the

type_element).

moloc is used as pointer D "access to collection .MODELOC Object .MODELOC contiguous Collection of V (I). moloc —

> V (I) V (1): code 1 V ELEM 2 ELNO __3 ELGA 4 VECTEUR __5 MATRICE (3): Gd quantity associated with the mode_local V (3): Nb _scal number

8.4.2 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

L" 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 Gauss points

of the element. One to indicate adds 10000
to the absolute value of nb_pt possibly that the various points
of the field do not have same representation (ELNO/DIFF). 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 item 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 is the number of the family "simple" partner with the _local

mode . Pointer in

object "&CATA.TM.NOFPG". KFPGL is the number of the family
"lists" associated with the mode_local. Pointer in L

"object "&CATA.TE.FPG_LISTE". If code = VECTEUR or MATRICE V (4): moloc (line) If code = MATRICE V (5): moloc (column

) Object .TAILLMAX: vector (I) length nb_te: V V (you): Max (.MODELOC (3)) for all the local modes of the type

_element you Options calculated by

type_element the Object

.OPTTE : Simple object

V (I) V ((you-1) *nb_op+op

8.4.3) — > i_optte: number of the optte (OPTion-Type_Element

) associated with CALCUL (opt , you). This number i_optte is used to point in

8.5 collections .OPTMOD and .OPTNOM.

8.5.1 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 computation V (2) nbin number of parameter "

8.5.2 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" 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)

8.5.3 will nom_para (out, nbout) Object" .CTE_ATTR': Collection

of V (K16) length nb_te This collection contains the attributes of all

type_element

. .CTE	_ATTR (you): V (
K16)	
LONG=2*nb_att ribut	V (2* (iattr
- 1) +	
1) : name of	the attribute of number

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

Note:: To find the value D" an attribute of name nom_

attr, one must traverse this vector until

finding this name with an odd index. sd_cata_phen _mode
: "&CATASTROPHES" 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": OJB S N K16

9 ♦ ".MECANIQUE": OJB XC

```
V I NO ♦ ".THERMIQUE    .MODL"
    : OJB S N K16             ♦  ".THERMIQUE   "
    : OJB XC V I NO   Object    .PHENOMENE   :  S N
    K16 This pointer of           names  contains  all
    the names of phenomenon       read  in  the catalog
    : Today              : "MECHANICAL
    " "ACOUSTIC"    "THERMAL     "  Note:
    : It is not used              to point  in
```

9.1 a collection . Objects .MODL

".ACOUSTIQUE .MODL": Names of the modelizations of the ACOUSTIC phenomenon.

".MECANIQUE .MODL

- ": Names of
- the modelizations
- of the MECHANICAL

phenomenon.
".THERMIQUE .MODL": Names of the modelizations

9.2 of the THERMAL

phenomenon. Other objects the other objects of the data structure Catastrophes _PHEN_MODE are not " suffixes" "into tough" in documentation . One creates as many additional objects of phenomena read . These objects

9.3 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 the possible modelizations for this phenomenon. A a given

modelization, corresponds a vector

of integers V. For i_tm

of 1 to nb_tm : V (i_tm): number of type_élément

associated with the type nets i_tm, for the modelization. So V (i_tm) =0: the type_maille i_tm type_element did not associate for the modelization. V (Nb

_tm +1): toplogic dimension

of the " principal" elements of the modelization: 0 2/1/3 V (nb_tm +2): dimension of

physical space bathing the modelization : 2/3