Data structures sd_carte , sd_cham_no , sd_cham_elem and sd_resuelem

Summarized:

Contents

1 General information

the data structures representing the fields are:

- `sd_cham_no` : field on the nodes of a mesh
- `sd_carte` : field on meshes of a mesh
- `sd_cham_elem` : field on the elements of a ligrel
- `sd_resuelem` : field of matrixes (or vectors) elementary on the elements of a ligrel

We call "quantity" a "vector" of components (CMP) of the field.
For example, for a field of displacement: ("DX", "DY", "DZ").
A discretized field is a set of quantities located on nodes, Gauss points or meshes.
All the quantities of a field do not have the same components inevitably: for example, on certain parts of the mesh, the nodes can have 6 CMPS of displacement (beam elements) whereas on other parts, the nodes have only 3 CMPS (voluminal elements).

The components of a quantity are a subset of the CMPS declared in the catalog of quantities [D4.04.01]. To describe a quantity, in addition to its numerical values, it is necessary to know of which CMPS it act; for that, one uses the notion of "descripteur_grandor which describes" the presence (or not) of all the CMPS of the catalog. This notion is described below.

- `The sd_carte` are fields discretized on meshes of a mesh (or the meshes late ones of a ligrel). There exists 1 quantity by mesh,
- `the sd_cham_no` are fields discretized on the nodes of a mesh (or late nodes of a ligrel). There exists 1 quantity by node,
- `the sd_cham_elem` are fields discretized on the elements of a ligrel. It can exist several quantities by element (for example a quantity by Gauss point or node). The points of dicretisation (nodes or Gauss point) can have subpoints; if it is the case, all the points have the same one number of subpoints,
- `the sd_resuelem` are fields discretized on the elements of a ligrel. The quantities associated with such fields are the quantities known as "elementary": elementary matrixes or elementary vectors. All the values of `a resuelem` can be bulky, this is why the object containing these values (.RESL) has a structure of dispersed collection.

Notice important:

The data structures described here are not easy use. They are SD normally used in operations of low levels: elementary computations, assemblies, resolutions… When one wants to read or write in such SD, it is often preferable to transform them beforehand into more convenient SD to use ("simple" fields). Routines of ad hoc transformation: CNOCNS, CNSCNO, CELCES, CARCES… are described in [D4.06.06].

2 Tree structures

```
sd_carte  (K19)  :: =record
  ♦ ".NOMA"     :   OJB   S.E.    K8
  ◊ ".NOLI"     :   OJB   S   V  K24
  ♦ ".DESC"     :   OJB   S   V  I
  ◊ ".LIMA"     :   OJB   XC  V  I
  ♦ ".VALE"     :   OJB   S   V  R/C/K8/K24/…

sd_cham_no  (K19)  :: =record
  ♦ ".DESC"     :   OJB   S   V  I
  ♦ ".REFE"     :   OJB   S   V  K24
  ♦ ".VALE"     :   OJB   S   V  R/C/K8
```

◊ % so FETI solver (REFE (3) = ' FETI') and CHAM_NO representing a second member or a vector solution

```
    ".FETC"     :   OJB   S   V  K24  long=nbsd (many subdomains)
```

```
sd_cham_elem  (K19)  :: =record
  ♦ ".CELK"     :   OJB   S   V  K24
  ♦ ".CELD"     :   OJB   S   V  I
  ♦ ".CELV"     :   OJB   S   V  R/C/K8/…

sd_resuelem  (K19)  :: =record
  ♦ ".NOLI"     :   OJB   S   V  K24
  ♦ ".DESC"     :   OJB   S   V  I
  ♦ ".RESL"     :   OJB   XD  V  R/C
  ◊ ".RSVI"     :   OJB   XC  V  I
```

3 Contained JEVEUX objects

3.1 DESCRIPTEUR_GRANDEUR

It is a vector of integers. It describes the CMPS present indeed in a quantity.

All the possible CMPS of a quantity are described in the catalog of quantities. They are ordered there. To describe the CMPS indeed present in a quantity one decides to keep a vector of Boolean which answers following question: is the ième CMP (in the order of the catalog of quantities) it presents in the quantity which one wants to describe? To save core memory (and disc), one decides "to code" this vector of Boolean on a vector of integers: on each integer (called entier_codé), one codes 30 Boolean.

Example:

If quantity "DEPL_R" were described in the catalog by:

DX	DY	DZ	DRX	DRY	DRZ	LAGR

On element of type a beam the descripteur_grandor is worth 126. Indeed:

	DX	DY	DZ	DRX	DRY	DRZ	LAGR
	1	1	1	1	1	1	0
126 =	21	+ 22	+ 23	+ 24	+ 25	+ 26	

On element of type `voluminal` descripteur_grandor is worth 14. Indeed:

	DX	DY	DZ	DRX	DRY	DRZ	LAGR
	1	1	1	0	0	0	0
14 =	21	+ 22	+ 23				

On an additional node creates for the introduction of kinematical condition by dualisation, the descripteur_grandor is worth 128. Indeed:

	DX	DY	DZ	DERX	DRY	DRZ	LAGR
	0	0	0	0	0	0	1
128 =							27

a descripteur_grandor is a vector of entier_codés: `V` of dimension `n_ec` where `n_ec` is the number of entier_codés necessary to the description of the quantity described in the catalog.

`n_ec`	many CMPS in the entier_codé
catalog	1 1 to
30	2 31 to
60	3

… ième informs about the presence (or not) numbered CMPS of `30* (i-1) +1 → 30*i`.

4 SD card

4.1 General information

a card is a field discretized by mesh. Each mesh can be "affected" of a quantity (with more). The cards are in general SD create starting from the data of the user. Its structure is made to store (with less possible volume) information concerning the assignment of the quantities on "pieces" of the mesh.

Note:

The selected structure is economic spaces some but she does not answer question quickly: which quantity is affected on the M1 mesh? To answer this question, it is necessary "to extend" the card (that to create bulkier temporary objects); it is the object of routine `ETENCA` called by `CALCUL`. A card is thus an ordered list of couples (quantity, zone_affectée). The order of the couples is important because it is used to take into account the principle of overload of the assignments: the last assignment takes precedence over the preceding ones.

One zone_affectée can be:

- the group of meshes of the mesh (`TOUT: "YES"`),
- all meshes late a ligrel,
- `a GROUP_MA` of the mesh,
- a list of meshes of the mesh,
- a list of meshes late of a ligrel.

4.2 Object .NOMA

Name of the mesh associated with the card.

4.3 Object .DESC

“.DESC” S V I DIM = 3 + (2+n_ec) *n_gd_max

field “ DOCU ” of the object . DESC contains: “ CART “

DESC (1)	Gd (number of the quantity associated with the card)
DESC (2)	n_gd_max (raising amongst zone_affectée)
DESC (3)	n_gd_edit (real number of zone_affectée) n_gd_edit can be = 0
DESC (3+1)	code_1er_zone (“code” of the zone_affectée first)
DESC (3+2)	number of the 1st zone_affectée
.....	
DESC (3+2*n_gd_max-1)	code_der_ent (code of the zone_affectée last)
DESC (3+2*n_gd_max)	number of the zone_affectée last

Attention:

one needs n_gd_max=1 for a constant card on all meshes late (for example CHTIME in me2mme.f)

the “code” of one zone_affectée can be worth: 1

	the group of meshes of the mesh (TOUT : “YES “), -1
L”	together of meshes late D” U N ligrel , 2
	GROUP_MA of the mesh , 3
	a list of meshes of the mesh , a -3
	list of meshes late of a ligrel . If

code =1 (or -1) :
the number of zone_affectée corresponding is not used for nothing. If

code =2:
the number of zone_affectée corresponding is the number of the group _ma in the collection netted .GROUPEMA If

code =3 (or -3) :
the number of zone_affectée corresponding is the number of the object of the collection .LIMA which contains the numbers of meshes composing zone_affectée. In

object .DESC a continuation comes then from descripteur_grandor describing the various affected quantities. That is to say n_ EC the number of entier_codé necessary to describe the CMPS of the quantity Gd : DESC

(3+2*n_gd_max+1) beginning	of the first descripteur_grandor….
….	DESC
(3+2*n_gd_max + (n_gd_max-1) *n_ec +1) beginning	of the last descripteur_grandor Note

: For

a constant field (1 only quantity assigned to all meshes of the mesh). One has then: DESC
(2) = 1 DESC
(3) = 1 DESC
(4) = 1 DESC
(5) = it does not matter DESC
(6) = beginning of the descripteur_grandor of zone_affectée (TOUT : "OUI' In this case .LIMA and .NOLI are not allocated (saving of space). Object

4.4 .NOLI This

object is present only if the card relates to the meshes late ones. It

is a vector of K24 of dimension Nb _gd_max. Opposite izone one finds, if this zone_affectée is a list of meshes late, the name of the ligrel or are defined these meshes. izone

```
---> nom_ligrel Object
```

4.5 .LIMA It

is a numbered contiguous family of vectors of integers. LIMA

```
(izone): V (I) V
```

contains the numbers of meshes constituting zone_affectée.

The numbers of meshes of the list are numbers relating to the ligrel referred in .NOLI (izone). if

a number of mesh is > 0, it is a mesh of the mesh associated with the card. if
a number of mesh is < 0, it is a mesh of additional ligrel . Object

4.6 .VALE It

is a vector of scalars dimensioned with Nb _gd_max * nb_cmp_max, if Nb _cmp_max is the number of CMPS in the catalog for the quantity associated with the card.

The quantity associated with zone_affectée the izone starts in .VALE with the index: izone

```
--> .VALE ((izone-1) *nb_cmp_max + 1) Attention
```

: Only

the affected CMPS are stored (consecutively and in the order of the catalog) in object .VALE For example, for a card of DEPL_R , if the 1st zone is affected by: (DX=2. and DZ =4.) VALE
(1) = 2. VALE
(2) = 4. SD

5 cham_no Object

5.1 .DESC

field "DOCU" of object .DESC contains : "CHNO" DESC

(1) Gd	(quantity associated with the cham _no) DESC
(2) num	DESC
(3),…, DESC (3 + n_ec - 1) descriptor	_grandor of the quantity if num is < 0 If

num is negative num = "-" nb_cmp If

num is < 0, its absolute value is the number of CMP of the quantity for TOUS the nodes of the mesh (e.g. the field of geometry). In this case the field relates to only the nodes of the mesh (not of late nodes) and it is supposed that all the nodes have the same representation of the quantity.

The descripteur_grandor is then stored DESC (3) with DESC (3 + n_ec - 1). If num is positive, there exists then a structure of the type professor _chno referred in object .REFE. Object

5.2 .REFE REFE

(1) name	of MAILLAGE. REFE
(2) name	of a professor _chno [D 4.06.07] (if DESC (2) >0) the SD professor _chno describes the CMPS carried by the nodes of the cham _no. It is used to point in the object .VALE which contains the values. If FETI , it acts of the professor _chno of the total field, then for each subdomain, it is of course that local with the subdomain. REFE
(3) So	FETI solver : "FETI" REFE
(4) So	FETI solver : name of data structure of SD_FETI type (information coming from NUMERICAL _DDL.NUME.REFN (4)). Object .VALE This

5.3 object contains

the "values" of the field at nodes on the nodes of the mesh or the late nodes of the ligrel used in the prof_chno. In the general case

6 the description

of object .VALE if the cham_no is not with "constant representation" is made in [D4.06.07 §3]. If

7 the cham_no is with "constant representation" Is: nb_no:

the number
of nodes of the mesh. ncmp: the number
of CMPS carried by all the nodes of the mesh. LENGTH (.VALE) = Nb

_no * ncmp VALE (1) value

of the 1st	CMP carried by the 1st node VALE (2) value
of the 2nd	CMP carried by the 1st node…… VALE (
ncmp)	value
of the last	CMP carried by the 1st node VALE (ncmp+1) value
of the 1st	CMP carried by the 2nd node…… the order
	of the CMPS

is that of the catalog of quantities (object "&CATA.GD.NOMGD" [D4.04.01]). Complements for FETI Object

7.1 .FETC S V K24 indirect

7.1.1 (*) DIM = nbsd

(many subdomains) (*): CHAM_NO not FETI (i.e. FETC
(K) .REFE (3) ("FETI " and
for the time imposed on MULT _FRONT') optional JEVEUX object
(present

only for total field if FETI, then absent for each subdomain) listing specific SD CHAM_NO to each subdomain . In the case of method FETI

, data structure CHAM_NO is recursive on two levels. A SD CHAM_NO "Master ", concerning the total field (.REFE (3) = ' FETI'), comprises the usual JEVEUX objects supplemented by a specific object of the decomposition of fields: the .FETC. It is in fact a pointer indicating
SD CHAM_NO" slaves "associated with each local subdomains. These SD CHAM_NO local are made up by the same JEVEUX objects as a CHAM_NO usual mono-field . For time , the implementation
of FETI in Code_Aster presupposes that these subdomains use all the same linear solver mono-field (.REFE (3) = ' MULT_FRONT' imposed by default). This homogeneity facilitates handling of the vectors solution and second local members. recursive data structure cham_no

7.1.2 for FETI: cham_no "main" (total field

) Object .FETC --
> subdomain
1 (cham_no slave) --> subdomain 2 (cham _no slave
) -->… In the case of one FETI solver

, one arbitrarily chose the following rule of naming for SD CHAM_NO slave related to a subdomain J: nom_de_la_SD_CHAM_NO_maître (1:

11)/"F" //chaîne_de_caractères_libre (2:8) \ \ the character string is generated

by a call to routine GCNCON. Example: The series of following

7.1.3 commands (resulting from benchmark FETI002A) debut (CODE=_F (NOM = ' FETI 002A', NIV

_PUB_WEB=' INTRANET')) MATER=DEFI_MATERIAU (ELAS=_F (E

```
= 180000. , NU = 0.30,
ALPHA = 15.E-6, RHO = 7700. ,)) MAIL=LIRE_MAILLAGE () MODM=AFFE
_MODELE (MAILLAGE=MAIL,
AFFE= (_F (GROUP_MA = "STRU", PHENOMENE
                = "MECHANICAL", MODELISATION = "D_PLAN",), _F
                        (GROUP_MA = "POISCR", MODELISATION
                      = "2D_DIS_T", PHENOMENE
                         = ' MECANIQUE'), _F (GROUP_MA = "POIACR",
MODELISATION
                      = "2D_DIS_T", PHENOMENE
                         = ' MECANIQUE'),)) CHCAR=AFFE_CARA_ELEM
(MODELE=MODM
, DISCRET= (_F (GROUP_MA=' POIACR',
                     CARA = "K_T_N
                     ", VALE = (0. , 0. , 0. ,),), _F (GROUP_MA=' POISCR',
CARA = "K_T_N
                     ", VALE = (180000. , 0. , 180000. ,),),))
CHMAT=AFFE_MATERIAU (MAILLAGE=MAIL
, AFFE= (_F (TOUT=' OUI', MATER=MATER
                   , TEMP_REF=20.,),)) CH1=AFFE_CHAR_MECA (MODELE=MODM,
PRES_REP= (_F (GROUP_MA=' DDLI', NEAR
                   = 1000. ,), _F (GROUP_MA=' DDLI1', NEAR = 2000
                             . ,),),) SDFETI=DEFI_PART_OPS (NOM=' SD',
MODELE=MODM, INFO=1, DEFI= (_F (GROUP_MA
                    = "FETI
                    1", GROUP_MA_BORD
                    = "B1",), _F (GROUP_MA = "FETI2", GROUP_MA_BORD
                          = "B2",), _F (GROUP_MA = "FETI3", GROUP_MA_BORD
                          = "B3",), _F (GROUP_MA = "FETI4", GROUP_MA_BORD
                          = "B4",),),); RESU=MECA_STATIQUE (MODELE=MODM,
CARA_ELEM=CHCAR, CHAM_MATER=CHMAT
                  , SOLVEUR=_F (METHODE
                  = ' FETI', PARTITION
                  =SDFETI), EXCIT= (_F (CHARGE
                            =CH1,),),) Built
                  a SD CHAM_NO" Master
```

""&&MESTAT .2NDMBR _ASS "… ====> IMPR_CO OF DATA STRUCTURE

```
: &&MESTAT.2NDMBR_ASS????? ATTRIBUT: F CONTENU: T BASE
 : > < COLLECTION   (OR)  OBJECT  MANY
 FIND: 4 ============================                     ==
================================================= PRINTING OF THE CONTENU OF THE
OBJECTS
  FIND:
--------------------------------------------------------------------------------
PRINTING SEGMENT OF VALUES >
&&MESTAT.2NDMBR_ASS.DESC < 1 - 36 1
--------------------------------------------------------------------------------

PRINTING SEGMENT OF VALUES >
&&MESTAT.2NDMBR_ASS.FETC < 1 - >&&MESTAT.2.F0000022 <>&       &MESTAT
      .2.F0000026 < 3 - >&     &MESTAT.2.F0000028 <>&       &MESTAT
      .2.F0000032 <
--------------------------------------------------------------------------------
PRINTING SEGMENT OF VALUES >
&&MESTAT.2NDMBR_ASS.REFE < 1 - >MAIL <>RESU .00000.NUMÉRIQUE
      < 3 - >FETI                    <>SDFETI    <
--------------------------------------------------------------------------------
```

```
PRINTING SEGMENT OF VALUES >
&&MESTAT.2NDMBR_ASS.VALE < 1 - 0.00000D+00 0.00000D+00
      0.00000  D+00 0.00000  D+00 0.00000  D+00 6 - 0.00000  D+00 0.00000  D+00
      0.00000  D+00 1.50000  D+03 1.12500  D+03 11 -  1.00000D+03  7.50000D+02
     5.00000  D+02 3.75000  D+02 0.00000  D+00 16 -  0.00000D+00  0.00000D+00
     0.00000  D+00 0.00000  D+00 0.00000  D+00 21 -  0.00000D+00  0.00000D+00
     0.00000  D+00 0.00000  D+00 0.00000  D+00 26 -  0.00000D+00  0.00000D+00
     0.00000  D+00 0.00000  D+00 0.00000  D+00 31 -  0.00000D+00  0.00000D+00
     0.00000  D+00 0.00000  D+00 1.00000  D+03 36 -  7.50000D+02  2.00000D+03
     1.50000  D+03 and of  SD CHAM_NO "  slaves"

"&&MESTAT.2.F00000…" of type… ====> IMPR_CO OF DATA STRUCTURE


: &&MESTAT.2.F0000022????? ATTRIBUT: F CONTENU:  T BASE
 : > < COLLECTION   (OR)  OBJECT  MANY
 FIND: 3 =============================                     ==
================================================= PRINTING OF THE CONTENU OF THE
OBJECTS
  FIND:
--------------------------------------------------------------------------------
PRINTING SEGMENT OF VALUES >
&&MESTAT.2.F0000022.DESC < 1 - 36 1
--------------------------------------------------------------------------------

PRINTING SEGMENT OF VALUES >
&&MESTAT.2.F0000022.REFE < 1 - >MAIL <>RESU. F0000007.NUMÉRIQUE
      < 3 - <
--------------------------------------------------------------------------------

PRINTING SEGMENT OF VALUES >
&&MESTAT.2.F0000022.VALE < 1 - 0.00000D+00 0.00000D+00
      0.00000  D+00 0.00000  D+00 1.00000  D+03 6 - 7.50000  D+02 1.00000  D+03
      7.50000  D+02 0.00000  D+00 0.00000  D+00 11 -  0.00000D+00  0.00000D+00
     2.00000  D+03 1.50000  D+03 During  an execution  in parallel
```

mode MPI, a processor is seen allotting a certain number of subdomains (additional cf objects "&FETI.LISTE…" data structure SD_FETI [D4.06.21]). SD CHAM_NO "Master " is always built , but its pointer .FETC will indicate only the subdomains concerned with the processor running: .FETC (jk) will be one K24 valid that if the jk subdomain is in the perimeter of the processor J. recursive data structure cham_no

7.1.4 for FETI if parallelism MPI: cham_no "main" (total field

) Object .FETC --
> "vacuum" --> j1
subdomain (cham _no
slave concerned with the processor J) --> "vacuum" --> j2
subdomain (cham _no
slave concerned with the processor J) -->… SD champ_elem Case
of
the cham_ elem

8 having

8.1 subpoints (structural elements) the number of points of discretization

(nodes, Gauss points,…) of a cham_elem on a mesh is a priori given by the number of points defined in the catalog of the type_elem associated with the mesh . For “the structure” elements type, one wants to be able to store more quantities than of points defined in the catalog. During a nonlinear computation

on a shell (for example), the integration chosen for the nonlinear behavior requires to store the stress state in several points in the thickness: it is necessary to discretize the thickness of the shell. For that, it will be said that each Gauss point positioned on the surface of the element (their number is fixed in the catalog of the type_elem), is composed of N subpoints representing the discretization of the norm to the element in this point. In the same way, a nonlinear

element of pipe, will be able to discretize its section (circular ring) by cutting out it in sectors and layers. For a given element, all

the points of discretization number of subpoints have obligatorily the same one. Caution: For creating

a cham_elem with

subpoints, it is necessary to say for all the elements of the ligrel the number of subpoints desired one. For that, one uses a cham_elem_s of quantity DCEL _I (argument DCELZ of the routine alchml.f). Lorsqu " one calls the routine of elementary computations (calcul.f), the transition of this argument is underground: the cham_elem_s must have the same name as the cham_ elem (OUT) qu" it is used to dimension . Case of the cham_elem not having

8.2 subpoints By means of computer, all the cham_elem

has subpoints. A cham _elem which does not need for this notion is in fact a cham_elem for which each point of discretization has one subpoint; one then confuses the point and his single subpoint. Case of the cham_elem of quantity VARI_R

8.3 quantity VARI_R is the "reserved

" quantity which is used to represent a quantity of which the number of components (CMP) is undetermined on the level of the catalogs of type_elem. One makes use for example of this quantity

to represent the local variables of the constitutive laws, because each model can have a number different from such variables. In the catalog of quantities, this

quantity has only one CMP: VARI. At the time of the creation of a cham

_elem_VARI_R, one must say for each element, how much components will have quantity VARI_R. These components will be called then: "V1", "V2",..., "Vn". For that, one uses the same mechanism as to declare the number of the subpoints (see above). Object .CELK CELK (1) name of the ligrel associated

8.4 with

the cham_elem	. CELK (2) name of the computation option
associated	with the cham_elem. CELK (3)/"ELNO": CHAM_ELEM with
the nodes	/"ELGA": CHAM_ELEM with Gauss points /"ELEM ": CHAM_ELEM constant by element CELK (4) numerical _couche: number of
the layer (tallied	on the left) for a CHAM_ELEM calculated on a layer of shell element . CELK (5) nive_couche: position in
the layer	for a CHAM_ELEM calculated on a layer of shell element : /"INF"/"MOY"/"SUP" CELK (6) Name of the parameter of the option
associated	with cham_elem CELK (7)/"MPI_COMPLET" /"MPI_INCOMPLET
"	CELK (7): "MPI_INCOMPLET": computation

- was distributed (MPI) and object .CELV is not complete on all the processors . Each processor calculated only one subset of the elements. "MPI_COMPLET": if not. I.e.
- that computation was not distributed, or, cham_elem (MPI_INCOMPLET with its creation) "was supplemented " by calling the utility sdmpic.f Object .CELD .CELD: vector of integers.

8.5 Field “

DOCU″ of object .CELD contains: “CHML” This object is the descriptor of the object

containing the values of the cham_elem (.CELV). CELD (1) Gd quantity associated with the cham

_elem.	CELD	(2) nb_gr number of grel of the associated
ligrel	. CELD	(3) mxsp maximum amongst subpoints
for		the elements of ligrel CELD (4) mxcmp maximum amongst CMP
(quantity	VARI_R	) for the elements of the ligrel. 0 if quantity different of VARI_R CELD (4+1) debu_grel_1 addresses (− 1) in
.CELD of	the beginning of information	concerning the 1st GREL… CELD (4+nb_gr) debu_grel_n addresses
(− 1)		
in .CELD of	the beginning of information	concerning the last GREL then one stores end to end the description

of the field for each GREL of ligrel CELD (debu_grel +1) nel number D "element

of GREL CELD (	debu_	grel +2) modelo mode_local associated
with local field	(or 0	so non-existent field on the GREL) CELD (debu_grel +3) lgcata length of
the local field within sight of	the catalog	. C" is to be said without taking account of the subpoints and of the components multiple of VARI_R. (= 0 if modelo = 0) CELD (debu_ grel +4) lggrel length total
of the segment containing	all	the values of the field on the GREL then: C iel = 1, nel (if modelo > 0)
CELD (debu_grel+4+4* (iel		- 1) + 1) nbsp number
of sous_points for the element		iel CELD (debu_grel+4+4* (iel-1) +2) ncdyn many
CMP (VARI_R) for the element		iel CELD (debu_grel+4+4* (iel-1) +3) lgchel many
values of the local field	for	the element iel. lgchel= lgcata * nbsp * ncdyn CELD (debu_grel+4+4* (iel-1) +4) adiel addresses
in object .CELV of	the 1st	value of the element iel Object .CELV It is a vector containing

8.6 end to end

the values of the local fields of the various elements. The description of the segment concerning an element is given by the definite mode_local for the type_elem. This description is possibly supplemented by the data amongst subpoints and CMPS (VARI_R). For a field of variables (different from

VARI_R) not having subpoints, all the elements of same a grel having same the type_elem, their local fields have all the same length and the same organization. One moves in object .CELV thanks to

object .CELD. One can describe the organization of

object .CELV by these definitions: CELV (ligrel) continuation of CELV (GREL) put end to end

CELV	(GREL) continuation of CELV (element) put
end to end	CELV (element) continuation of CELV (not) put
end to end	CELV (not) continuation of CELV (subpoint)
put end to end	CELV (subpoint) continuation of CMP (scalar
) put end to end	Some "formulas" frequently used

8.7 in programming LIGREL number of the quantity associated

8.7.1 with the CHAM_ELEM

: NUMGD=ZI (JCELD-1+1) many GREL of

the LIGREL associated with

the CHAM_ELEM: NGREL =ZI (JCELD - 1+2) maximum number. subpoints

of the elements

of a CHAM_ELEM: (perhaps = 0) MXSP=ZI (JCELD – 1+3) maximum number. CMPS

(VARI_R) of the elements

of a CHAM_ELEM : (/=0 <=> VARI_R) MXCDY=ZI (JCELD – 1+4) GREL

: IGR number D
"elements D"

8.7.2 a GREL (IGR)

: NEL=ZI (JCELD-1+ZI (JCELD-1+4 +IGR) +1) _local

mode of a GREL (IGR): IMOLO=ZI

(JCELD-1+ZI (JCELD-1 +4+IGR) +2) cumulated

length of the elements of

a GREL (IGR): LGGREL=ZI (JCELD-1+ZI (JCELD- 1+4+IGR) +4)

addresses (in .CELV) beginning of GREL

IGR: DEBUGR=ZI (JCELD-1+ZI (JCELD-1 +4+IGR) +8) then

: ZR (JCELV -1 +DEBUGR) =… length
(CATALOG) of an element of

a GREL (IGR): LGCATA=ZI (JCELD-1+ZI (JCELD- 1+4+IGR) +3)

Element IEL of GREL IGR addresses (in

8.7.3 .CELV) beginning of element

IEL of GREL IGR: ADIEL=ZI (JCELD-1+ZI (JCELD - 1+ 4+IGR) +4 +4

* (IEL-1) +4) then: ZR (JCELV -1 +ADIEL) =… length
of element IEL of GREL IGR

: LGIEL=ZI (JCELD-1+ZI (JCELD - 1+ 4+IGR) +4 +4

* (IEL-1) +3) number of subpoints of element IEL

of GREL IGR: NBSPT=ZI (JCELD-1+ZI (JCELD - 1+ 4+IGR) +4 +4

* (IEL-1) +1) it does not have there subpoints: NBSPT=0 (or
1; notice JD) many CMPS (VARI_R) of element IEL

of GREL IGR : NCDYN=ZI (JCELD-1+ZI (JCELD - 1+ 4+IGR) +4 +4

* (IEL-1) +2) 0 - > the quantity is not VARI_R SD resuelem
Object .NOLI NOLI (1) name of

9 the ligrel associated

9.1 with the resuelem

. NOLI	(2) name of the option associated with
the resuelem	. NOLI (3)/“MPI_COMPLET” /“MPI_INCOMPLET
”	NOLI (4)/““/” VOISIN_VF” NOLI
(3): “	MPI_INCOMPLET”: computation

- was distributed (MPI) and the object. RESL is not complete on all the processors . Each processor calculated only one subset of the elements. “MPI_COMPLET”: if not. I.e.
- that computation was not distributed, or, resuelem (MPI_INCOMPLET with its creation) “ was supplemented” by calling the utility sdmpic.f NOLI (4): this value has meaning only for

the resuelem “matrix” ““: the elementary matrixes are “ordinary
- ”. For an element, the elementary matrix is a “square” (or a demy-folio if the matrix is symmetric) concerning the ddls of L” element “VOISIN_VF”: the elementary matrixes
- are of type “VOISIN_VF” (larger). In this case, one stores end to end the matrixes concerning the coupling of the ddls of L” element with those of these neighbors. See the details in the object. RSVI. Object .DESC field “DOCU” of L “object

9.2 .DESC contains

: "RESL" DESC (1) Gd (quantity associated with the resuelem

) DESC (	2) nb_gr (many GREL of .NOLI (1)
) DESC (	2+1) mode_1er_gr (_local mode of the local
fields	of the first GREL)… DESC (2+nb_gr) mode_der_gr (_local mode
of	
the last GREL)	Object .RESL C" is a dispersed collection

9.3 of vectors

of R (or C). The access to this collection is done by the number of GREL: .RESL (IGREL) → V If ncmpel is the number of scalars representing

the local field for an element of the GREL, V (1,…, ncmpel) values of the field on the 1st

element of the GREL	V (ncmpel+1,…, 2*ncmpel) values of the field
on the 2nd element of	the GREL…… Attention: For the resuelem "
matrix	

" of type

"CLOSE _VF", the number of values per element is not ncmpel. To move in. RESL, it is necessary to use the object. RSVI . See below. Because of the " distribution " of elements (AFFE_MODELE/PARTITION/PARALLELISME = " GROUP_ELEM ") objects of the collection. RESL can exist within the meaning of JEEXIN without L" one being able to make a JEVEUO (… ," It,…) above because they do not have a disk image . To test if these objects really exist, routine JAEXIN should be used. Object .RSVI This object exists only when

9.4 the resuelem

is of type "stamps" and that it is of type "CLOSE _VF" . It is a contiguous collection of vectors of I. The access to this collection is done by the number of GREL: .RSVI (IGREL) → V If nel is the number of elements of the GREL

, V is length nel+1 V (1) (=1) : index in. RESL of the beginning

of	the elementary matrix of iel =1 V (2) index in. RESL of the beginning of the elementary
	matrix of iel =2 V (3) index in. RESL of the beginning of the elementary
	matrix of iel =3…… V (nel+1) (cumulated length of
the elementary	matrixes
	of the grel) +1 Is: iel1, iel2,…, ielp the p neighbors

of iel

0 nddl1, nddl2,…, nddlp the number of ddls
of the p neighbors of iel0 nddl 0: number of ddls of iel0 Then :

the elementary matrix MEL of iel

0 is

of dimension nddl0* (nddl0 +nddl 1+… +nddlp) MEL (1) - > coupling (iel0,1) X (iel0,1) MEL (
2) - > coupling (iel0,1) X (iel0,2)…
MEL (nddl 0+1) - > coupling (iel0,2) X (iel
0,1
)… MEL (nddl 0*nddl0) - > coupling (iel0, nddl
0)
X (iel0, nddl0) MEL (nddl0*nddl0 +1) - > coupling (iel0,1)
X

(iel1,1)… MEL (nddl 0* (nddl0+nddl 1) +1) - > coupling
(iel
0,1) X (iel2,1)… Examples SD card CARD = CREA_CHAMP (TYPE
_

10 FIELD: "

10.1 CART_META

```
 _R", OPERATION: "AFFE", MESH: AFFE NETTED: (TOUT: "OUI' NOM_CMP
                 : ("ZF" "ZP
                 " "ZB" "ZM" "P"    ) VALE
                          : (0.0 0.0 0.0 0.0 0.0)) AFFE:
                          (GROUP_MA    :   GM2 NOM_CMP   : (  "ZF" "
                 ZP") VALE: (0.2    0.3
                          )) AFFE: (MESH:
                          T2 NOM_CMP    : (  "ZP"  "ZM"
                 "P") VALE: (    0.4
                          0.5 0.6))); PRINTING
                          SEGMENT    OF VALUES  >  CARD.
                DESC


 < 1 - 64 3 3 3 1 6 - 3 2 3             3 254        11
      - 254                      254
--------------------------------------------------------------------------------


PRINTING OF THE COLLECTION: CARD .LIMA
 PRINTING OBJET OF CONTIGUOUS COLLECTION              >CARTE
 .LIMA< OC: 1 1 - 1 3 PRINTING OBJET             OF CONTIGUOUS       COLLECTION
                 >CARTE
 .LIMA< OC: 2 1 - 2 PRINTING OBJET OF            CONTIGUOUS  COLLECTION
      >CARTE
 .LIMA< OC: 3 1 - 4 5
--------------------------------------------------------------------------------

PRINTING SEGMENT OF VALUES >CARTE .NOLI
 < 1 - > <> < 3 - > <
--------------------------------------------------------------------------------


PRINTING SEGMENT OF VALUES >CARTE .NOMA
```

```
 < 1 - >MAILLA <
-------------------------------------------------------------------------------

PRINTING SEGMENT OF VALUES >CARTE .VALE
 < 1 - 2.00000E-01 3.00000E-01 0.00000              E+00
      0.00000  E+00 0.00000  E+00 6 -  0.00000E+00  0.00000E+00  2.00000E-01
      4.00000  E-01 0.00000  E+00 11 -  5.00000E-01  0.00000E+00  0.00000E+00
     6.00000  E-01 0.00000  E+00 16 -  0.00000E+00  0.00000E+00  0.00000E+00
     0.00000  E+00 0.00000  E+00 21 -  0.00000E+00  Note::  The contents of
     the objects  printed
```

above can

surprise: it does not correspond to what is known as higher. Indeed this card "was finished" by a call to the routine tecart.f. The purpose of this optional action is `allowing`
a "fine" overload of the values affected in command CREA_CHAMP. SD cham_no cham_no = CREA_CHAMP (`Mesh`: netted

10.2 , TYPE

```
_CHAMP: "NOEU_DEPL_R", OPERATION: "AFFE", AFFE: (GROUP_NO: gn1 nom_cmp
          : "DX" VALE_R: 1.0)
          AFFE: (THE NODE IS OUTSIDE THE FIELD OF DEFINITION WITH A RIGHT PROFILE
OF THE EXCLU TYPE NODE: (N2, n7)
                NOM_CMP: ("DX", "DZ") vale_
          R: (2. , 4.))); PRINTING
                SEGMENT OF VALUES >CHAM_NO .DESC



 < 1 - 32 6
-------------------------------------------------------------------------------

PRINTING SEGMENT OF VALUES >CHAM_NO .REFE
 < 1 - >MAILLA <>CHAM_NO <
-------------------------------------------------------------------------------

PRINTING SEGMENT OF VALUES >cham_no .VALE
 < 1 - 2.00000E+00 4.00000E+00 1.00000             E+00        1.00000
       E+00 2.00000  E+00 6 - 4.00000  E+00 SD cham  _elem FLUXN  =CALC_CHAM_
      ELEM (  MODELE=MOTH
```

10.3 , TEMP=T2,

```
CHAM_MATER=CHMAT, OPTION    = ' FLUX_ELNO'  ) PRINTING
            SEGMENT OF VALUES   >FLUXN .CELD <


>>>>> 1 - 47 2 1 0 6 6 - 18 2             6520        8
 16 11
      - 1           0            8            1            1            16
      - 0           8            9         3  6857                     21
     - 6            18            1            0            6            26
     - 17            1            0            6            23           31
     - 1            0            6 29            PRINTING
     SEGMENT                                OF            VALUES
     >FLUXN                        .CELV                        <

 >>>>> 1 - -8.78595D-12 -4.27645D-12 -8.78595                    D
 12 -4.08919
```

```
      D-12 6.96696D-12 6 - -4.07954D-12 6.96696D-12 -4.77838  D-12 4.96957
      D-12 -4.15161  D-12 11 - 4.96957D-12 -4.26679  D-12 -1.33159D-12 -4.54543
      D-12 -1.33159D-12 16 - -3.57760D-12 7.27596D-12 -8.41283D-12 7.27596
     D-12 -8.41283  D-12 21 - 7.27596D-12 -8.41283  D-12 0.00000D+00 -8.86757
      D-12 0.00000D+00 26 - -8.86757  D-12 0.00000D+00 -8.86757  D-12 0.00000
     D+00 -8.86757  D-12 31 - 0.00000D+00 -8.86757  D-12 0.00000D+00 -8.86757
      D-12 PRINTING SEGMENT OF  VALUES >FLUXN .CELK <

 >>>>> 1 - >MOTH .MODELE <>FLUX_ELNO              < 3 -         >
 ELNO
      <> < 5 - >    <>PFLUX         _R SD resuelem
      CHTH= AFFE_CHAR_THER
      (MODELS                          : MODEL
```

10.4 TEMP_IMPO: (

```
THE NODE IS OUTSIDE THE FIELD OF DEFINITION WITH A RIGHT PROFILE OF THE EXCLU TYPE
NODE: N8 TEMP: 3.4) SOURCE: (TOUT: "OUI'  SOUR: 7.)); VECTEL=CALC_
                     VECT_ELEM (CHARGE: CHTH  OPTION: "        CHAR_THER
"); The resuelem is extracted from VECT_ELEM VECTEL
```

: "VECTEL .VE 001" PRINTING SEGMENT OF VALUES >VECTEL .VE001

```
 .DESC < 1 - 105 3 5781 5648 0
--------------------------------------------------------------------------------

PRINTING SEGMENT OF VALUES >VECTEL .VE001
 .NOLI < 1 - >MODEL .MODELE <>CHAR_THER  _SOUR     _R <
--------------------------------------------------------------------------------

PRINTING OF THE COLLECTION: VECTEL .VE001
 .RESL PRINTING OBJET OF COLLECTION >  VECTEL     .VE001
 .RESL< OC: 1 1 - 3.50000E+00 3.50000  E+00     3.50000  E+00      4.66667
       E+00 4.66667  E+00 6 - 4.66667  E+00 PRINTING  OBJET  OF COLLECTION
      >VECTEL  .VE001
 .RESL< OC: 2 1 - 4.08333E+00 4.66667  E+00     4.66667  E+00      4.08333
       E+00
```