Structures of data champ_no_s and cham_elem_s

Summary:

This document describes the structures of data cham_no_s and cham_elem_s.

One also gives the list of the principal utilities working on these structures of data.

Two new SD are defined: cham_no_s and cham_elem_s who contain same information as the SD cham_no and cham_elem but which is more “simple” with manipuler in FORTRAN.

There exist utilities making it possible to transform one cham_no in cham_no_s (and reciprocally) (in the same way for cham_elem).

These SD will be thus in general temporary SD making it possible to work more simply.

Notice important:

SD cham_no_s and cham_elem_s are not as general as the SD cham_no and cham_elem. For cham_no_s, one describes only the fields carried by the nodes of the grid (and not the possible late nodes), For cham_elem_s, one does not describe that fields carried by the finite elements whose mesh support is a mesh of the grid (and not a late mesh)

Contents

1 SD cham_no_s

```
cham_no_s (K19)          :: = record
```

- '.CNSK' : OJB S V K8 length = 2
- '.CNSD' : OJB S V I length = 2
- '.CNSC' : OJB S V K8 length = nb_CMP
- '.CNSV' : OJB S V R/C/I/… length = nb_NOEUD*nb_CMP
- '.CNSL' : OJB S V L length = nb_NOEUD*nb_CMP

1.1 Contents of the OJB

This SD is used to describe a field of sizes carried by the nodes of a grid.

1.2 Object '.CNSK'

'.CNSK' (1)	netted : name of the grid subjacent with cham_no_s.
'.CNSK' (2)	nomgd : name of the size associated with cham_no_s ('DEPL_R', 'SIEF_R', …)

1.3 Object '.CNSD'

'.CNSD' (1)	nb_NOEUD : many nodes of the subjacent grid.
'.CNSD' (2)	nb_CMP : maximum number of the CMPS carried by the nodes.

1.4 Object '.CNSC'

'.CNSC' (ICMP)	cmp_i : name of the ième CMP of cham_no_s

Note:
The order of CMPS in.CNSC can be unspecified. One is not obliged to respect the order of the catalogue of the sizes. On the other hand, them CMPS must be part of CMPS size nomgd.

1.5 Object '.CNSV'

This object contains the values of cham_no_s. The type of this vector JEVEUX (R/C/I/K8,…) is that of the size nomgd. Its dimension is nb_NOEUD*nb_CMP; i.e. all the nodes of grid can carry all the CMPS described in.CNSC.

One reaches ICMP- ème CMP INO- ème NODE by the formula:

```
VALUE (INO, ICMP) = .CNSV ((INO-1) *NB_CMP + ICMP)
```

Note:
The presence (or the absence) of a CMP on one NODE is indicated via the object.CNSL (see below). During the creation of one cham_no_s, its nonaffected values are in settings with "undef" for better detecting their illicit use.

1.6 Object '.CNSL'

This object contains the Boolean ones indicating the presence (or the absence) of the values of cham_no_s.
Its dimension is nb_NOEUD*nb_CMP; one moves there in the same way that in the object .CNSV

One examines the presence of ICMP- ème CMP INO- ème NODE by the formula:

```
EXIST (INO, ICMP) = .CNSL ((INO-1) *NB_CMP + ICMP)
```

2 SD cham_elem_s

2.1 Description of the SD

This structure of data makes it possible to represent the values of the fields discretized on the meshs of a grid.

More precisely, the access to an actual value (or complex,…) field is done while specifying:

- the number of the mesh supporting the finite element (IMA),
- the number of the point in the mesh (IPT),
- the number of the under-point in the point (ISP) (ISP =1 in general),
- the number of the component of the size associated with the field (ICMP),

```
cham_elem_s (K19)                :: = record

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

2.2 Object .CESK

.CESK (1)	netted : name of the grid subjacent with cham_elem_s.
.CESK (2)	nomgd : name of the size associated with cham_elem_s ('DEPL_R', 'SIEF_R',...)
.CESK (3)	/'ELNO': field known with the nodes of the elements, /'ELGA': field known at the points of Gauss of the elements, /'ELEM': constant field by element (one will then say qu' it is known in the centre of gravity)

2.3 Object .CESD

.CESD (1)	nb_MAILLE : many meshs of the subjacent grid.
.CESD (2)	nb_CMP : many CMPS carried by the points. It is the dimension of the object.CESC
.CESD (3)	nbptmx : maximum amongst points carried by the meshs
.CESD (4)	nbspmx : maximum amongst under-points carried by the points of the meshs
.CESD (5)	nucmpmx : highest sequence number of the possible CMP of cham_elem_s (in the order of the object .CESC)
.CESD (5+4* (ima-1) +1)	nbpt (ima): many points of the mesh ima.
.CESD (5+4* (ima-1) +2)	nbsp (ima): many under-points of the mesh ima.
.CESD (5+4* (ima-1) +3)	nbcmp (ima): maximum number of the CMPS carried by the under-points of the points of the mesh ima.
.CESD (5+4* (ima-1) +4)	IAD (ima): IAD+1 is the address in the objects.CESL and.CESV 1st CMP of the 1st under-point of the 1st point of the mesh ima (if they exist)

2.4 Object .CESC

.CESC (ICMP)	cmp_i : name of the ième CMP of cham_elem_s

Note:

The order of CMPS in .CESC can be unspecified. One is not obliged to respect the order of the catalogue of the sizes. On the other hand, them CMPS must be part of CMPS size nomgd (except the size VARI_R).

2.5 Objects .CESL and .CESV

These objects contain the values of cham_elem_s (.CESV) and of Boolean (.CESL) indicating if these values were affected (or if they are unspecified).

The type JEVEUX (R/C/I/K8,…) object.CESV is that of the size nomgd.

The dimension of these 2 vectors is nbval :

nbval is the sum on all the meshs ima of nbpt (ima) * nbsp (ima) *nbcmp (ima)

to reach ICMP - ème CMP ISP - ème Under-POINT IPT - ème NOT IMA - ème MESH of one cham_elem_s , the utility routine is used CESEXI :

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

where: JCESD and JCESL are the addresses of the objects .CESD and .CESL cham_elem_s. IAD is the “exit” of this routine.

if IAD > 0 , that wants to say that the required component exists in cham_elem_s. One can then recover it by: VALUE = ZR (JCESV-1+IAD) (if the field is real).

if IAD < 0 , that wants to say that the required component has a possible place in cham_elem_s but that it not affected currently. One can then affect a value in cham_elem_s while making:

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

if IAD = 0 , that wants to say that the required component does not have possible room in cham_elem_s. I.e. that one at least of the following conditions is checked:

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

2.6 Example of loop on the values of one cham_elem_s

```
CAL JEVEUO (THESE '.CESD', 'IT, JCESD)
CAL JEVEUO (THESE '.CESL', 'IT, JCESL)
CAL JEVEUO (THESE '.CESV', 'IT, JCESV)
NBMA = ZI (JCESD-1+1)
C 40, IMA = 1, NBMA
   NBPT = ZI (JCESD-1+5+4* (IMA-1) +1)
   NBSP = ZI (JCESD-1+5+4* (IMA-1) +2)
   NBCMP = ZI (JCESD-1+5+4* (IMA-1) +3)
   C 30, IPT = 1, NBPT
      C 20, ISP = 1, NBSP
         C 10, ICMP = 1, NBCMP
            CAL CESEXI (STOP, JCESD, JCESL, IMA, IPT, ISP, ICMP, IAD)
            IF (IAD.GT.0)  VALUE = ZR (JCESV-1+IAD)
```

3 Utility routines

.

CARCES	to transform one map in one cham_elem_s
CELCES	to transform one cham_elem in cham_elem
CESCES	to change the discretization of one cham_elem (ELNO/CART/ELGA)
CESCNS	to transform one cham_elem_s in one cham_no_s
CESCRE	to create one cham_elem_s
CESEXI	to test the existence of one CMP of a point of a mesh of one cham_elem_s
CESRED	"to reduce" one cham_elem_s on a list of meshs and/or a list of CMPS.
CESTAS	"retasser" contents of one cham_elem_s
CNOCNS	to transform one cham_no in cham_no_s
CNSCES	to transform one cham_no_s in cham_elem
CNSCNO	to transform one cham_no_s in cham_no
CNSCRE	to create one cham_no_s
CNSPRJ	to project one cham_no_s on another grid
CNSRED	"to reduce" one cham_no_s on a list of nodes and/or a list of CMPS.

COPISD	to copy one cham_no_s or one cham_elem_s
DETRSD	to destroy one cham_no_s or one cham_elem_s
IMPRSD	to print on listing one cham_no_s or one cham_elem_s