

Description of the routine CALCULATION

1 Introduction

The routine CALCULATION is the routine of encapsulation of all elementary calculations : elementary matrices, vectors or fields.

In this description being used for maintenance of the routine CALCULATION, we will present successively:

- 1) simplified flow charts of the principal routines of CALCULATION ("dynamic" aspect of the program) [§2] and [§3],
- 2) then them **data** handled ("static" aspect). It is:
 - nodes, meshes [D3.01.01], type_element, finite elements, sizes [§4], GREL, one LIGREL, an option [D3.02.01] and [D4.06.02],
 - catalogues of finite elements [D4.04.01],
 - FIELDS : CHAM_NO, CHAM_ELEM, MAP and RESUELEM [D4.06.05],
 - "CARDS extended" [§5],
 - objects JEVEUX of work specific to the "package" CALCULATION [§6],
 - as well as COMMONS clean with the "package" CALCULATION [§7].
- 3) finally, us listels uses of COMMONS by the various routines [§8]

The routine CALCULATION primarily takes in argument the name of an option which one wishes calculation, the name of one LIGREL on which one wishes to carry out this calculation and of the parameter lists IN/OUT and fields IN/OUT corresponding to these parameters which are necessary to the calculation of the option. These lists are noted in the continuation LPAIN, LPAOU, LCHIN and LCHOU,

2 Flow charts

2.1 Simplified flow chart of the routine CALCULATION

The names of the routines are written in capital letters *Italic*. Only the great stages of the routine are presented.

CALCULATION (...)

DEBCA1 ! Initialization of COMMON CAII02 and CAII11

! Setting with clean of LPAIN, LCHIN, LPAOU and LCHOU
! It is checked that the parameters of LPAIN exist in the description of
! the option and that there are no doubled blooms in LPAIN and LPAOU

DEBCAL ! "Prologue" of the routine CALCULATION:
! setting in COMMON of addresses JEVEUX
! "extension" of the CARDS IN [\$2]
! various checks

ALRSLT ! Allowances of the total fields results (OUT)
ALCHLO ! Allowance of the local fields

EXTRA1 ! In the case of finished volumes: extraction local field
! associated with bread for the GREL grel by taking of
account them
! neighbors

buckle on the GREL of the LIGREL: grel

! When one uses parallelism 'GROUP_ELEM', one can sometimes
! "to jump" a whole GREL
if this GREL is not calculated by this processor: goto fine
buckles grel

INIGRL ! Allowances of objects '&INEL.XXX' for the GREL grel

EXTRA1 ! Extraction of the local fields associated with the LPAIN
! for the GREL grel

ZECHLO ! Zero setting of the OUT local fields
CAUNDF ! Addition of UNDEF to the ends of fields IN/OUT for
! later checks

TE0000 ! Call to the effective TE000I (elementary calculations)

CAUNDF ! Later checks of the UNDEF
RISE ! Recopy of the OUT local fields in the fields
! total results

fine grel buckles

RISE ! For finished volumes

if distributed parallel calculation: *SDMPIC* ! Communication MPI

2.2 Simplified flow charts of the routine EXTRA1

EXTRA1 (LCHIN, ...)

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Copyright 2021 EDF R&D - Licensed under the terms of the GNU FDL (<http://www.gnu.org/copyleft/fdl.html>)

```
! In the case finished volumes, one must allocate all the fields
before
    ! buckle on the grels to give access to the "neighbors"
if finished volumes and if one is before the loop on the grels
    buckle on the GREL of the LIGREL
        EXTRA1
    fine grel buckles
if not
    EXTRA1
EXTRA1 (LCHIN,...)
    buckle on LCHIN: champ_in
    so STANDARD (champ_in) = ' CARTE'
        EXCART(champ_in)
    so STANDARD (champ_in) = ' CHAM_ELEM
        EXCHML(champ_in)
    so STANDARD (champ_in) = ' CHAM_NO'
        EXCHNO(champ_in)
    so STANDARD (champ_in) = ' RESUELEM
        EXRESL(champ_in)
    end if
    fine buckles on LCHIN

EXCART (champ_in,...)
    EXCAR1

EXCAR1(champ_in,...)
    buckle on the elements of the GREL: iel
        TRIGD          ! recovery of the size carried
                        ! by the mesh ima associated with the element iel
    fine iel buckles

EXCHNO (champ_in,...)
    buckle on the elements of the GREL: iel
        buckle on the nodes of the mesh associated with iel: ino
            TRIGD          ! recovery of the size carried
                            ! by the node ino of the element iel
        fine ino buckles
    fine iel buckles

EXCHML (champ_in,...)
    buckle on the elements of the GREL: iel
        JACOPO          ! recopy of the "end" of the total field in the local
field
    fine iel buckles

EXRESL (idem EXCHML)
```

2.3 Simplified flow chart of the routine ALRSLT

```
ALRSLT (l_champ_out,...)
    buckle on the fields of l_champ_out: chout
        DETRSD(chout) ! destruction of the field
    if TYPE(chout) = ' CHAM_ELEM'
        ALCHML(chout)
    if TYPE(chout) = ' RESUELEM'
        ALRESL(chout)
    end if
```

Titre : Descriptif de la routine CALCUL
Responsable : PELLET Jacques

Date : 09/10/2013 Page : 4/18
Clé : D9.02.03 Révision :
1ef5e8ee9324

fine chout buckles

! Update of COMMON CAII07

3 Description summary of the utility routines

ALCHLO	allocate the local fields
ALCHML	allocate a field result of the type CHAM_ELEM
ALRESL	allocate a field result of the type RESUELEM
ALRSLT	allocate the total fields result
DCHLMX	the dimension max. of a local field makes
DEBCA1	“prologue” with the routine CALCULATION
DEBCAL	“prologue” with the routine CALCULATION
DIGDE2	the dimension of a local field returns
ETENCA	one “extends” MAP :
	creation of the objects MAP ‘.PTMA’ and MAP ‘.PTMS’
EXCART	extract a local field from one MAP
EXCHML	extract a local field from one CHAM_ELEM
EXCHNO	extract a local field from one CHAM_NO
EXISDG	test the existence of one CMP in one DESCRIPTEUR_GRADEUR
EXRESL	extract a local field from one RESUELEM
EXTRAI	extract a local field starting from a total field
GRDEUR	give the name of the size associated with a parameter
INIGRL	initialize the objects “&INEL.XXX” for one GREL
INPARA	the number of a parameter for a couple returns (type_element, option) given
NETTED	the name returns of GRID associated with one LIGREL
MODATT	returns the mode local expected by one type_element for a given parameter
RISE	recopy a local field result in a total field.
NBEC	the number of entreties coded for a given size returns
NBELEM	the number of elements of one returns GREL
NBGREL	the number returns of GREL of one LIGREL
NBPARA	the number of parameters for a couple (type_element, option returns) given
NOPARA	the name of a parameter for a couple (type_element, option returns) given
NUCALC	the number of the routine returns TE000I for a couple (type_element, option) given
SCALAI	the type makes scalar: R, I, C,... of a size
TE0000	routine “hat” which calls all them TE000I
TRIGD	sort them CMPS of a size according to one DESCRIPTEUR_GRADEUR
TYPELE	it returns type_element associated with one GREL
ZECHLO	met a local field with “zero” between 2 GRELS

4 Recall on the sizes

We will call “instanciée size” (or “size” very runs to reduce):

- a reference to a “size catalogued” defined in the catalogue of the sizes,
- a vector of realities (or complexes, entières,...) whose components are associated with CMPS catalogued size,
- one DESCRIPTEUR_Grandeur : it is a vector of coded entières which informs about the presence (or the absence) of CMPS size catalogued in the instanciée size

For example, a field of displacements contains a set of instanciées sizes. Each size of this field is defined by:

- a reference to the size DEPL_R catalogue: real type, CMPS named: DX, DY,...
- a vector of 2 realities (for example): (2.3, 3.4)
- a coded entirety: ICODE=2 ** 2 + 2 ** 3 = 14 (for example), which makes it possible to say that for this size, DX is absent, DY is worth 2.3 and DZ is worth 3.4

A field (total or local) is primarily a list of instanciées sizes assigned to geometrical entities:

- nodes for one CHAM_NO
- meshes for one MAP
- be finite elements for one RESUELEM or one CHAM_ELEM

5 “Wide” cards

One MAP is an affected field by meshes or groups of meshes. The structure of data MAP “is condensed”: one stores in with respect to each instanciée size the list of the meshes affected by this size. In CALCULATION, the basic problem for the use of CARDS is the following: “how to find the instanciée size associated with the mesh ima to be able to recopy it in the local field provided to the routines TE00IJ ? “

This problem cannot be solved effectively with the structure MAP (especially if it is thought that a mesh can be affected several times: principle of “overload”), this is why at the beginning of CALCULATION (routine DEBCAL), they “are extended” CARDS “IN” (routine ETENCA). This extension consists in creating for each MAP, 2 temporary additional objects which make it possible to make association (mesh, size) once and for all.

These 2 objects correspond to the SD CARTE_ETENDUE following:

```
CARTE_ETENDUE (K19) :: = record
    \ $VIDE'          : MAP
    \ .PTMA'         : OBJ S V I LENGTH = nb_ma
    \ .PTMS'         : OBJ S V I LENGTH = nb_ms
```

Object \ .PTMA' :

that is to say nb_ma the number of meshes of GRID associated with the map,
for ima=1, nb_ma :
.PTMA (ima) : number of the size associated with the mesh ima

Object \ .PTMS' :

that is to say nb_ms the number of additional meshes of LIGREL associated with the map,
for ims=1, nb_ms :
.PTMA (ims) : number of the size associated with the additional mesh ims

6 Objects of work

The routine CALCULATION allocate objects JEVEUX of work (bases BIRD) that we will describe below. All these objects have a name starting with '&&CALCUL' .

6.1 &&CALCUL.OBJETS_TRAV OJB S V K24

This object contains the name of all the objects of work created by CALCULATION. It is used to do the "housework" (destruction) at the end of the routine CALCULATION. This object is created by the routine DEBCAL. The address of this object JEVEUX is stored in COMMON CAI105.

6.2 Local fields

One calls "local field" object JEVEUX (and zones it associated memory) where the information extracted from a total field is stored if "IN" (or calculated by a routine TE000I if "OUT"). This information "is arranged" in accordance with the description given in the catalogue of type_element (mode_local of the parameter associated with the field).

The local fields are vectors JEVEUX whose names are form '&&CALCUL' //nom_parameter, for example:
'&&CALCUL.PGEOMER', '&&CALCUL.PCACOQU', ...

Note:

- One continues to use the "local" vocabulary although since 1993, the "local" field contains (put end to end) information concerning all the finite elements of GREL running.
- When a routine TE00IJ fact for example:

CAL JEVECH ('PGEOMER', 'IT', 'IAD)

the address IAD returned is the address of the object '&&CALCUL.PGEOMER' shifted of what it is necessary so that one reaches information concerning the current element.

The organization of these objects is the following one:

1st CMP	1st size	1st element
2nd CMP	1st size	1st element
3rd CMP	1st size	1st element
1st CMP	2nd size	1st element
2nd CMP	2nd size	1st element
...		
1st CMP	1st size	2nd element
2nd CMP	1st size	2nd element
...		

One finds the various elements end to end there of GREL (which has the same one type_element and thus same the mode_local).

example for a local field of geometry to the nodes of one GREL of TRIA3 :

DX	Node: 1	1st element
DY	Node: 1	1st element
DX	Node: 2	1st element
DY	Node: 2	1st element
DX	Node: 3	1st element
DY	Node: 3	1st element
DX	Node: 1	2nd element
DY	Node: 1	2nd element
...		

- Local fields of work for conversions:

For conversions, CALCULATION zones memory "plug needs". These zones memory have the same organization as the local fields above. It is necessary to allocate 2 for each scalar type of them associated with the sizes of the fields of calculation: I, R, C, K8,...

Their names are:

&&CALCUL.ORIGINAL.C	&&CALCUL.CONVERTI.C
&&CALCUL.ORIGINAL.I	&&CALCUL.CONVERTI.I
&&CALCUL.ORIGINAL.K8	&&CALCUL.CONVERTI.K8
&&CALCUL.ORIGINAL.K16	&&CALCUL.CONVERTI.K16
&&CALCUL.ORIGINAL.K24	&&CALCUL.CONVERTI.K24
&&CALCUL.ORIGINAL.R	&&CALCUL.CONVERTI.R

For the conversion of a local field of real type (for example), conversion will use as starter the local field: &&CALCUL.ORIGINAL.R and at exit: &&CALCUL.CONVERTI.R

These local fields "plug" are allocated with the max. length of the local fields "ordinary".

6.3 Objects &&CALCUL.NOM_&INEL and &&CALCUL.IAD_&INEL

- &&CALCUL.NOM_&INEL

This object contains them **names** objects related to the initialization of type_element of the courant GREL: objects '&INEL.XXX'.

- &&CALCUL.IAD_&INEL

This object contains them **addresses** objects related to the initialization of type_element of the courant GREL: objects '&INEL.XXX'.

6.4 Objects `&&CALCUL.TYPE_SCA` `&&CALCUL.IA_CHLOC` and `&&CALCUL.MODELO`

- `&&CALCUL.TYPE_SCA` S V K8 will dim=nb_para
- `&&CALCUL.IA_CHLOC` S V I will dim=7*nb_para
- `&&CALCUL.MODELO` S V I will dim=nb_para

That is to say will nb_para the number of parameters (“in” and “out”) of the option of calculation and ipar the number of such a parameter.

<code>&&CALCUL.TYPE_SCA</code> (ipar)	scalar type (I, R, C,...) size associated with the parameter ipar
<code>&&CALCUL.IA_CHLOC</code> (7* (ipar-1)+1)	address of the local field associated with ipar
<code>&&CALCUL.IA_CHLOC</code> (7* (ipar-1)+2)	length of the local field associated with ipar
<code>&&CALCUL.IA_CHLOC</code> (7* (ipar-1)+3)	local mode expected for the parameter
<code>&&CALCUL.IA_CHLOC</code> (7* (ipar-1)+4)	working length of the local field for 1 element. (account takes of ICOEF)
<code>&&CALCUL.IA_CHLOC</code> (7* (ipar-1)+5)	type of the local field: 1: MAP 2: CHAM_NO 3: CHAM_ELEM 4: VECTOR 2ND MEMBER 5: MATRIX
<code>&&CALCUL.IA_CHLOC</code> (7* (ipar-1)+6)	many points of discretization of the local field: 1 for one MAP nb_no for one CHAM_NO nb_pg for one CHAM_ELEM nb_no for one VECTOR 0 for one MATRIX
<code>&&CALCUL.IA_CHLOC</code> (7* (ipar-1)+7)	value of the “multiplying” coefficient (ICOEF) for CHAM_ELEM with dynamic size (local modes Zxxxxxx)
<code>&&CALCUL.MODELO</code> (ipar)	mode_local associated with the parameter ipar

6.5 Objects `&&CALCUL.LCHIN_EXI`, `&&CALCUL.LCHIN_K8` and `&&CALCUL.LCHIN_I`

- `&&CALCUL.LCHIN_EXI` S V L dim= nb_in
- `&&CALCUL.LCHIN_K8` S V K8 dim= 11*nb_in
- `&&CALCUL.LCHIN_I` S V I dim= 2*nb_in

That is to say nb_in the number of fields "in" and iin =1, nb_in

<code>&&CALCUL.LCHIN_EXI</code> (iin)	.FALSE. : the field does not exist
<code>&&CALCUL.LCHIN_K8</code> (2* (iin-1)+1)	type of the field: 'CHNO', 'CART', 'CHML' OR 'RESL'
<code>&&CALCUL.LCHIN_K8</code> (2* (iin-1)+2)	scalar type of the field: 'IT', 'R', 'I', 'K8',...
<code>&&CALCUL.LCHIN_I</code> (11* (iin-1)+1)	IGD : size associated with the field
<code>&&CALCUL.LCHIN_I</code> (11* (iin-1)+2)	NEC : many coded entireties
<code>&&CALCUL.LCHIN_I</code> (11* (iin-1)+3)	NCMPMX : max. number of CMPS for IGD
<code>&&CALCUL.LCHIN_I</code> (11* (iin-1)+4)	IADESC : address of CHIN//'.DESC'
<code>&&CALCUL.LCHIN_I</code> (11* (iin-1)+5)	IAVALE : address of CHIN//'.VALE'
<code>&&CALCUL.LCHIN_I</code> (11* (iin-1)+6)	IAPTMA : address of CHIN//'.PTMA' (if MAP)
<code>&&CALCUL.LCHIN_I</code> (11* (iin-1)+7)	IAPTMS : address of CHIN//'.PTMS' (if MAP)
<code>&&CALCUL.LCHIN_I</code> (11* (iin-1)+8)	IAPRN1 : address of .PRNO (\$MAILLA) (if CHAM_NO)
<code>&&CALCUL.LCHIN_I</code> (11* (iin-1)+9)	IAPRN2 : address of .PRNO (LIGREL) (if CHAM_NO)
<code>&&CALCUL.LCHIN_I</code> (11* (iin-1)+10)	IANUEQ : address of .NUEQ (if CHAM_NO)
<code>&&CALCUL.LCHIN_I</code> (11* (iin-1)+11)	1: IANUEQ is valid 0: if not

6.6 Objects `&&CALCUL.LCHOU_K8` and `&&CALCUL.LCHOU_I`

- `&&CALCUL.LCHOU_K8` S V K8 dim= 2*nb_out
- `&&CALCUL.LCHOU_I` S V I dim= 2*nb_out

That is to say nb_out the number of "out" fields and iout =1, nb_out

<code>&&CALCUL.LCHOU_K8</code> (2* (iout-1)+1)	type of the field: 'CHML' (CHAM_ELEM) or 'RESL' (RESUELEM)
<code>&&CALCUL.LCHOU_K8</code> (2* (iout-1)+2)	scalar type of the field: 'IT', 'R'
<code>&&CALCUL.LCHOU_I</code> (2* (iout-1)+1)	IADESC : address of CHOUT//'.DESC'
<code>&&CALCUL.LCHOU_I</code> (2* (iout-1)+2)	IAVALE : address of CHOUT//'.VALE' (if CHAM_ELEM)

6.7 Objects `&&CALCUL.SCALAIRE` and `&&CALCUL.IA_CONVERS`

- `&&CALCUL.SCALAIRE` S V K8 dim= nb_scal
- `&&CALCUL.IA_CONVERS` S V I dim= 2*nb_scal + 1

That is to say nb_scal the number of the scalar types: I, R, C, ...

<code>&&CALCUL.SCALAIRE (I)</code>	I ^{ème} possible scalar type of the fields "in" field: <code>SCAL (I)</code>
--	--

<code>&&CALCUL.IA_CONVERS (1)</code>	nb_scal
<code>&&CALCUL.IA_CONVERS (2* (i-1) +2)</code>	address in ZR or ZC or... of the object: <code>\&&CALCUL.ORIGINAL. ' //SCAL (I)</code>
<code>&&CALCUL.IA_CONVERS (2* (i-1) +3)</code>	address in ZR or ZC or... of the object: <code>\&&CALCUL.CONVERTI. ' //SCAL (I)</code>

6.8 Objects `&&CALCUL.TECAEL_K24` and `&&CALCUL.TECAEL_I`

- `&&CALCUL.TECAEL_K24` S V K24
- `&&CALCUL.TECAEL_I` S V I

that is to say nb_no the number of nodes of the mesh associated with the current element.

<code>&&CALCUL.TECAEL_K 24 (1)</code>	name of GRID
<code>&&CALCUL.TECAEL_K 24 (2)</code>	name of LIGREL
<code>&&CALCUL.TECAEL_K 24 (3)</code>	name of the mesh
<code>&&CALCUL.TECAEL_K24 (3+1)</code>	name of the 1st node of the mesh
<code>&&CALCUL.TECAEL_K24 (3+2)</code>	name of the 2nd node of the mesh
...	...
<code>&&CALCUL.TECAEL_I (1)</code>	number of the mesh
<code>&&CALCUL.TECAEL_I (2)</code>	many nodes of the mesh (nb_no)
<code>&&CALCUL.TECAEL_I (2+1)</code>	number of the 1st node of the mesh
<code>&&CALCUL.TECAEL_I (2+2)</code>	number of the 2nd node of the mesh
...	...

6.9 Objects `&&CALCUL.NOMOP` and `&&CALCUL.NOMTE`

- `&&CALCUL.NOMOP` S V K16
- `&&CALCUL.NOMTE` S V K16

<code>&&CALCUL.NOMOP (iopt)</code>	name of the option of number iopt
<code>&&CALCUL.NOMTE (ite)</code>	name of type_element of number ite

7 Description of COMMONS clean with the routine CALCULATION: ICOELXX

7.1 COMMONS ICOEL1 and ICOEL2

Description summary:

Information concerning the total field and the local field associated with the parameter running in the loop on the extraction.

```
COMMON /ICOEL1/IGD, NEC, NCMPMX, IACHIN, IACHLO, IICHIN, IANUEQ, LPRNO
COMMON /ICOEL2/TYPEGD
CHARACTER*8 TYPEDG
C THESE COMMONS ARE UPDATED BY EXTRAI.
C IGD: NUMBER OF THE SIZE ASSOCIATED WITH THE FIELD TO BE EXTRACTED
C NEC: MANY WHOLE CODES OF IGD
C NCMPMX: NUMBER MAX OF CMPS FOR IGD
C IACHIN: ADDRESS JEVEUX OF CHIN.VALE
C IACHLO: ADDRESS JEVEUX OF CHLOC.VALE (&&CALCUL.NOMPAR)
C IICHIN: NUMBER OF FIELD CHIN IN LIST LCHIN.
C IANUEQ: ADDRESS OF OBJECT .NUEQ OF THE PROF_CHNO ASSOCIATES POSSIBLE
C - LIES TO FIELD CHIN. (IF LPRNO=1).
C LPRNO: 1-> OBJECT .NUEQ IS TO TAKE INTO ACCOUNT
C (CHAM_NO WITH PROF_CHNO)
C 0-> OBJECT .NUEQ IS NOT TO TAKE INTO ACCOUNT
C (CHAM_NO WITH CONSTANT REPRESENTATION OR ANOTHER FIELD)
C TYPEDG: SCALAR TYPE OF SIZE IGD: 'R', 'I', 'K8',...
```

7.2 COMMON ICOEL3 , ICOELD , ICOELE and ICOELH

Description summary:

Information concerning the conversion of a local field.

```
COMMON /ICOEL3/IAMOD1, IAMOD2, IAORIG, IACONV
C THIS COMMON IS UPDATED BY CONVER.
C IAMOD1 : ADDRESS OF THE MODE-LOCAL BEFORE CONVERSION
C IAMOD2 : ADDRESS OF THE MODE-LOCAL AFTER CONVERSION
C IAORIG : ADDRESS OF CHAMP-LOCAL '&&CALCUL.ORIGINAL.SCAL'
C IACONV : ADDRESS OF CHAMP-LOCAL '&&CALCUL.CONVERTI.SCAL'

COMMON /ICOELD/IACVRS
COMMON /ICOELE/IASCAL
C THESE COMMONS ARE UPDATED BY ALCHLO.
C THESE COMMONS ARE USE BY CONVER.
C IASCAL : ADDRESS OF '&&CALCUL.SCALAIRE' V (K8)
C V (1). , V (NR): POSSIBLE TYPE_SCALAIRES OF THE CHIN.
C IACVRS : ADDRESS OF '&&CALCUL.IA_CONVERS' V (I)
C - DIM (V) = 2*DIM ('&&CALCUL.SCALAIRE') +1
C - V (1) = DIMENSION OF '&&CALCUL.SCALAIRE'
C - FOR I =1, NB_SCALAIRE
C - V (1+ 2* (I-1) +1) = ADDRESS IN ZR, OR ZC, OR ZI,...
C '&&CALCUL.ORIGINAL.SCAL (I)'
C - V (1+ 2* (I-1) +2) = ADDRESS IN ZR, OR ZC, OR ZI,...
C '&&CALCUL.CONVERTI.SCAL (I)'

COMMON /ICOELH/PARAM
CHARACTER*8 PARAM
C THIS COMMON IS UPDATED BY CONVER AND EAST USES IN
C TE00IJ OF CONVERSION.
C PARAM : NAME OF THE PARAMETER TO BE CONVERTED
```

7.3 COMMON ICOEL4 and ICOEL7

Description summary:

Information concerning the objects of the catalogue of finite elements "&CATASTROPHES.XXX".

```
COMMON /ICOEL4/IAOPTT, LGCO, IAOPMO, ILOPMO, IAOPNO, ILOPNO, IAOPDS,  
+ IAOPPA, LGOPPA, IAMLOC, ILMLOC, IACOVE, ILCOVE, IADSGD  
C THIS COMMON EAST INITIALIZES BY DEBCAL  
C THIS COMMON EAST USES A LITTLE EVERYWHERE  
C IAOPTT: ADDRESS OF THE OBJECT OF THE CATALOGUE: '&CATA.TE.OPTTE'  
C LGCO : LENGTH OF A COLUMN OF '&CATA.TE.OPTTE'  
C (FULL NUMBER OF POSSIBLE OPTIONS OF THE CATALOGUE)  
C IAOPMO: ADDRESS OF '&CATA.TE.OPTMOD'  
C ILOPMO: ADDRESS OF THE PT_LONG OF '&CATA.TE.OPTMOD'  
C IAOPNO: ADDRESS OF '&CATA.TE.OPTNOM'  
C ILOPNO: ADDRESS OF THE PT_LONG OF '&CATA.TE.OPTNOM'  
C IAOPDS: ADDRESS OF '&CATA.OP.DESCOPT (OPT)'  
C IAOPPA: ADDRESS OF '&CATA.OP.OPTPARA (OPT)'  
C LGOPPA: LENGTH OF '&CATA.OP.OPTPARA (OPT)'  
C IAMLOC: ADDRESS OF '&CATA.TE.MODELOC'  
C ILMLOC: ADDRESS OF THE PT_LONG OF '&CATA.TE.MODELOC'  
C IACOVE: ADDRESS OF '&CATA.TE.CONVERS'  
C ILCOVE: ADDRESS OF THE PT_LONG OF '&CATA.TE.CONVERS'  
C IADSGD: ADDRESS OF '&CATA.GD.DESCRIGD'  
  
COMMON /ICOEL7/IADFNO, IADFCA, LGCONO, LGCOCA  
C THIS COMMON IS UPDATED BY DEBCAL  
C THIS COMMON EAST USES IN EXTRAI AND ALCHLO  
C IADFNO: ADDRESS OF '&CATA.TE.MODEFNO'  
C IADFCA: ADDRESS OF '&CATA.TE.MODEFCA'  
C LGCONO: LENGTH OF 1 COLUMN OF MODEFNO.  
C LGCOCA: LENGTH OF 1 COLUMN OF MODEFCA.
```

7.4 COMMON ICOEL5

Description summary:

Information concerning connectivities of the meshes of the grid and the late meshes

```
COMMON /ICOEL5/IAMACO, ILMACO, IAMSCO, ILMSCO, IALIEL, ILLIEL  
C THIS COMMON IS UPDATED BY DEBCAL (OR TERLIG)  
C THIS COMMON EAST USES IN NUMAIL, EXCHNO,...  
C IAMACO : ADDRESS OF THE CONNECTIVITY OF THE GRID  
C ILMACO : ADDRESS OF THE POINTER LENGTH OF IAMACO  
C IAMSCO : ADDRESS OF THE CONNECTIVITY OF MESHES SUPPL. OF 1 LIGREL  
C ILMSCO : ADDRESS OF THE POINTER LENGTH OF IAMSCO  
C IALIEL : ADDRESS OF OBJECT '.LIEL' OF THE LIGREL.  
C ILLIEL : ADDRESS OF THE POINTER LENGTH OF '.LIEL'.
```

7.5 COMMON ICOEL8

Description summary:

Information concerning the whole of all the fields total "in".

```
COMMON /ICOEL8/IACHII, IACHIK, IACHIX
C THIS COMMON IS UPDATED BY DEBCAL
C THIS COMMON EAST USES IN EXTRAI, EXCHNO, EXCART, EXRESL, EXCHML
C , DCHLMX
C IACHII: ADDRESS OF '&&CALCUL.LCHIN_I'
C IACHIK: ADDRESS OF '&&CALCUL.LCHIN_K8'
C IACHIX: ADDRESS OF '&&CALCUL.LCHIN_EXI'
C
C '&&CALCUL.LCHIN_EXI':: = V (L) (DIM = NIN)
C V (1): .FALSE. : THE FIELD PARAMETER DOES NOT EXIST.
C
C '&&CALCUL.LCHIN_K8' :: = V (K8) (DIM = NIN*2)
C V (1): TYPE_CHAMP: 'CHNO', 'CART', 'CHML' OR 'RESL'.
C V (2): TYPE_GD : 'IT', 'R', 'I', 'K8',...
C
C '&&CALCUL.LCHIN_I' :: = V (I) (DIM = NIN*11)
C V (1): IGD SIZE ASSOCIATED WITH LCHIN (I)
C V (2): NEC MANY WHOLE CODES
C V (3): NCMPMX NUMBER MAX OF CMP FOR IGD
C V (4): IADESC ADDRESSES .DESC
C V (5): IAVALE ADDRESSES .VALE
C V (6): IAPTMA ADDRESSES .PTMA (FOR 1 MAP)
C V (7): IAPTMS ADDRESSES .PTMS (FOR 1 MAP)
C V (8): IAPRN1 ADDRESSES PRNO ($MAILLA) (FOR 1 CHAM_NO)
C V (9): IAPRN2 ADDRESSES PRNO (LIGREL) (FOR 1 CHAM_NO)
C V (10): IANUEQ ADDRESSES .NUEQ (FOR 1 CHAM_NO)
C V (11): LPRNO (KNOWN AS IF IANUEQ EAST USES FOR 1 CHAM_NO)
```

7.6 COMMON ICOEL9

Description summary:

Information concerning the names of the options and type_element as well as the list of the objects of routine work CALCULATION.

```
COMMON /ICOEL9/IANOOP, IANOTE, NBOBTR, IAOBTR, NBOBMX
C THIS COMMON IS UPDATED BY DEBCAL
C THIS COMMON EAST USES IN TE0000 FOR
C IANOOP: ADDRESS IN ZK16 OF '&&CALCUL.NOMOP' V (K16)
C V (IOP) --> NAME OF OPTION IOP
C IANOTE: ADDRESS IN ZK16 OF '&&CALCUL.NOMTE' V (K16)
C V (ITE) --> NAME OF TYPE_ELEMENT ITE
C THIS COMMON EAST USES IN ALCHLO, ALRSLT AND CALCULATION FOR:
C NBOBTR: MANY OBJECTS OF WORK '&&CALCUL....' WHICH
C WILL HAVE TO BE DESTROYED AT THE END OF CALCULATION.
C IAOBTR: ADDRESS IN ZK24 OF OBJECT '&&CALCUL.OBJETS_TRAV'
C NBOBMX: LENGTH OF OBJECT '&&CALCUL.OBJETS_TRAV'
```

7.7 COMMON ICOELA

Description summary:

number of the GREL running, number of the element running (in the GREL), characteristic of the whole of the local fields.

```
COMMON /ICOELA/IAWMOL, NPARIO, IAWLOC, IAWTYP, NBELGR, IGR, IEL
C THIS COMMON IS INITILISE BY ALCHLO
C THIS COMMON EAST MODIFIES BY MECOE1 (OBJECT .IA_CHLOC)
C THIS COMMON EAST MODIFIES BY EXTRAI, RISE, CALCULATION, CONVER, ZECHLO
C (OBJECT .MODELO)
C THIS COMMON EAST MODIFIES A LITTLE EVERYWHERE FOR NBELGR, IGR, IEL
C THIS COMMON EAST USES IN EXTRAI, RISE, CALCULATION, CONVER,
C JEVECH, ZECHLO, TECACH
C
C IGR : NUMBER OF THE GREL WHICH ONE TREATS
C NBELGR: MANY ELEMENTS IN GREL IGR
C (IGR AND NBELGR ARE UPDATED BY CALCULATION)
C IEL : NUMBER OF THE ELEMENT (IN GREL IGR)
C (IEL IS UPDATED BY EXTRAI, TE0000, CONVER, IS ASSEMBLED,...)
C IAWMOL: ADDRESS IN ZI OF '&&CALCUL.MODELO' V (I)
C V (IPAR) --> LOCAL MODE OF PARAMETER IPAR (OPT)
C NPARIO: LENGTH OF '&&CALCUL.MODELO' (NB OF PARAM IN/OUT (OPT))
C
C IAWLOC: ADDRESS IN ZI OF '&&CALCUL.IA_CHLOC' V (I)
C THIS OBJECT CONTAINS INFORMATION ON THE CHAMP_LOCAUX
C V (7* (IPAR-1) +1) --> ADDRESS OF CHAMP_LOCAL '&&CALCUL. //NOMPAR (IPAR)
C V (7* (IPAR-1) +2) --> LENGTH OF OBJECT '&&CALCUL. //NOMPAR (IPAR)
C L=MAX (PRESENT TYPE_ELEM) *NBELGR
C V (7* (IPAR-1) +3) --> LOCAL MODE EXPECTED FOR PARAMETER (IPAR)
C V (7* (IPAR-1) +4) --> WORKING LENGTH OF THE CHAMP_LOCAL FOR 1 ELEMENT
C TYPE_ELEM ASSOCIATES WITH GREL IGR: NCMPEL (MODE)
C THIS LENGTH TAKES ACCOUNT OF A POSSIBLE ICOEF
C = 1 (CF V (7* (IPAR-1) +7)).
C V (7* (IPAR-1) +5) --> STANDARD OF CHAMP_LOCAL:
C 1: MAP
C 2: CHAM_NO
C 3: CHAM_ELEM
C 4: VECTOR 2ND MEMBER
C 5: MATRIX
C V (7* (IPAR-1) +6) --> MANY POINTS OF DISCRETIZATION OF THE CHAMP_LOC
C (0 IF MATRIX)
C V (7* (IPAR-1) +7) --> VALUE OF THE MULTIPLYING COEFFICIENT FOR
C LOCAL MODES OF CHAM_ELEM "ZXXXX" (ICOEF).
C
C IAWTYP: ADDRESS IN ZK8 OF '&&CALCUL.TYPE_SCA' V (K8)
C V (IPAR) --> TYPE_SCALEIRE OF THE CHAMP_LOCAL
```

7.8 COMMON ICOELC

Description summary:

Information concerning the whole of all the fields total "out".

```
COMMON /ICOELC/IACHOI, IACHOK
C THIS COMMON IS UPDATED BY ALRSLT
C THIS COMMON EAST USES IN RISE, DCHLMX
C IACHOI: ADDRESS OF '&&CALCUL.LCHOU_I'
C IACHOK: ADDRESS OF '&&CALCUL.LCHOU_K8'
C
C '&&CALCUL.LCHOU_K8' :: = V (K8) (DIM = NIN*2)
C V (1): TYPE_CHAMP: 'CHML' OR 'RESL'.
C V (2): TYPE_GD : 'IT, 'R'
```

```
C      '&&CALCUL.LCHOU_I'  :: = V (I)      (DIM = NOUT*2)
C          V (1): ADDRESS OF L_CHOUT (I) .DESC
C          V (2): ADDRESS OF L_CHOUT (I) .VALE (IF CHAM_ELEM)
C
```

7.9 COMMON ICOELF

Description summary:

Information concerning the whole of the objects related to the intialisation of type_element: "&INEL.XXX"

```
COMMON /ICOELF/NBOBJ, IAINEL, ININEL
C      NBOBJ : MANY OBJECTS '&INEL.XXXX' CREATE BY INITIALIZATION
C          TYPE_ELEM
C      ININEL: ADDRESS IN ZK24 OF OBJECT '&&CALCUL.NOM_&INEL'
C          WHO CONTAINS THE NAMES OF OBJECTS '&INEL.XXXX'
C      IAINEL: ADDRESS IN ZI OF OBJECT '&&CALCUL.IAD_&INEL'
C          WHO CONTAINS THE ADDRESSES OF OBJECTS '&INEL.XXXX'
C      THIS COMMON EAST INITIALIZES BY DEBCAL
C      THIS COMMON EAST USES BY CALCULATION AND JEVETE
```

7.10 COMMON ICOELG

Description summary:

Information concerning the mesh subjacent with the current element

```
COMMON /ICOELG/ICAELI, ICAELK
C      THIS COMMON EAST CREATES BY DEBCAL.
C      IT IS USES BY TECAEL
C      ICAELK IS THE ADDRESS OF A VECTOR DE K24 CONTAINING:
C      V (1): NAME OF THE GRID (K8)
C      V (2): NAME OF THE LIGREL (K19)
C      V (3): NAME OF THE MESH (K8)
C      V (3+ 1) : NAME OF the 1st NODE OF the MESH
C      V (3+ 1) : NAME OF DER NODE OF THE MESH
C      ICAELI IS THE ADDRESS OF A VECTOR OF IS CONTAINING:
C      V (1): NUMBER OF THE MESH
C      V (2): MANY NODES OF MESH (NBNO)
C      V (2+ 1) : NUMBER OF the 1st NODE OF the MESH
C      V (2+NBNO): NUMBER OF DER NODE OF THE MESH
C      V (2+NBNO +1): NUMBER OF THE GREL
C      V (2+NBNO +2): NUMBER OF THE ELEMENT IN THE GREL
```


8 Use of the COMMONS: ICOELXX

8.1 List of the routines using the COMMONS ICOELXX

These routines should not be called apart from the routine CALCULATION

ALCHLO	EXRESL	NBPARA
ALRSLT	WILL EXTRAI	WILL NOPARA
CALCULATION	GRDOR	NUCALC
CONVER	WILL INPARA	OPCONV
DCHLMX	JEVECH	TE0000
DEBCAL	JEVETE	TECACH
DIGDE2	MECOE1	TECAEL
EXCART	MECOEL	TRIGD
EXCHML	MODATT	ZECHLO
EXCHNO	RISE	
EXPAND	MEANS	

8.2 Use of the COMMONS ICOELXX

Common block cross-country race-reference lists:

```

Common Block ICOEL1 used in:
EXCART      EXCHML      EXCHNO      EXRESL      EXTRAI      TRIGD
Common Block ICOEL2 used in:
CONVER      EXCHML      EXPAND      EXRESL      EXTRAI      MEANS
TRIGD
Common Block ICOEL3 used in:
CONVER      EXPAND      MEANS      (+ routines TEOOIJ of conversion)
Common Block ICOEL4 used in:
CALCULATION      CONVER      DEBCAL      DIGDE2      EXCART
EXCHNO
EXTRAI      GRDEUR      INPARA      JEVECH      MECOE1      MODATT
RISE      NBPARA      NOPARA      NUCALC      OPCONV      TECACH
ZECHLO
Common Block ICOEL5 used in:
DEBCAL      EXCART      EXCHNO      TECAEL
Common Block ICOEL7 used in:
DCHLMX      DEBCAL      EXTRAI
Common Block ICOEL8 used in:
CALCULATION      DCHLMX      DEBCAL      EXCART      EXCHML
EXCHNO
EXRESL      EXTRAI
Common Block ICOEL9 used in:
ALCHLO      ALRSLT      CALCULATION      DEBCAL      TE0000
Common Block ICOELA used in:
ALCHLO      CALCULATION      CONVER      EXCART      EXCHML
EXCHNO
EXPAND      EXRESL      EXTRAI      JEVECH      MECOE1      RISE
MEANS      TE0000      TECACH      TECAEL      ZECHLO
Common Block ICOELC used in:
ALRSLT      DCHLMX      RISE
Common Block ICOELD used in:
ALCHLO      CONVER
Common Block ICOELE used in:
ALCHLO      CONVER
Common Block ICOELF used in:
CALCULATION      DEBCAL      JEVETE
Common Block ICOELG used in:
DEBCAL      TECAEL
Common Block ICOELH used in:
CONVER      (+ routines TEOOIJ of conversion)

```

Code_Aster

Version
default

Titre : Descriptif de la routine CALCUL
Responsable : PELLET Jacques

Date : 09/10/2013 Page : 18/18
Clé : D9.02.03 Révision :
1ef5e8ee9324