

## Descriptif de la routine CALCUL

---

### 1 Introduction

---

La routine `CALCUL` est la routine d'encapsulation de tous les calculs élémentaires : matrices, vecteurs ou champs élémentaires.

Dans ce descriptif servant à la maintenance de la routine `CALCUL`, nous présenterons successivement :

- 1) les organigrammes simplifiés des routines principales de `CALCUL` (aspect "dynamique" du programme) [§2] et [§3],
- 2) puis les **données** manipulées (aspect "statique"). Ce sont :
  - des nœuds, des mailles [D3.01.01], des `type_element`, des éléments finis, des grandeurs [§4], des `GREL`, un `LIGREL`, une option [D3.02.01] et [D4.06.02],
  - des catalogues d'éléments finis [D4.04.01],
  - des `CHAMPS` : `CHAM_NO`, `CHAM_ELEM`, `CARTE` et `RESUELEM` [D4.06.05],
  - des "CARTES étendues" [§5],
  - des objets `JEVEUX` de travail propres au "paquet" `CALCUL` [§6],
  - ainsi que des `COMMONS` propres au "paquet" `CALCUL` [§7].
- 3) enfin, nous listons les usages des `COMMONS` par les différentes routines [§8]

La routine `CALCUL` prend essentiellement en argument le nom d'une option dont on souhaite le calcul, le nom d'un `LIGREL` sur lequel on souhaite réaliser ce calcul et des listes de paramètres IN/OUT et des champs IN/OUT correspondant à ces paramètres qui sont nécessaires au calcul de l'option. Ces listes sont notées dans la suite `LPAIN`, `LPAOU`, `LCHIN` et `LCHOU`,

## 2 Organigrammes

### 2.1 Organigramme simplifié de la routine CALCUL

Les noms des routines sont écrits en majuscules italiques. Seules les grandes étapes de la routine sont présentées.

*CALCUL* (...)

*DEBCA1* ! Initialisation des COMMON CAII02 et CAII11

! Mise au propre de LPAIN, LCHIN, LPAOU et LCHOU  
! On vérifie que les paramètres de LPAIN existent dans la description de  
! l'option et qu'il n'y a pas de doublons dans LPAIN et LPAOU

*DEBCAL* ! "Prologue" de la routine CALCUL :  
! mise dans des COMMON d'adresses JEVEUX  
! "extension" des CARTES IN [\$2]  
! vérifications diverses

*ALRSLT* ! Allocations des champs globaux résultats (OUT)  
*ALCHLO* ! Allocation des champs locaux

*EXTRAI* ! Dans le cas des volumes finis : extraction du champ local  
! associé à pain pour le GREL grel en prenant en compte les  
! voisins

boucle sur les GREL du LIGREL : grel

! Lorsqu'on utilise le parallélisme 'GROUP\_ELEM', on peut parfois  
! « sauter » tout un GREL  
si ce GREL n'est pas calculé par ce processeur : goto fin boucle  
grel

*INIGRL* ! Allocations des objets '&INEL.XXX' pour le GREL grel

*EXTRAI* ! Extraction des champs locaux associé aux LPAIN  
! pour le GREL grel

*ZECHLO* ! Mise à zéro des champs locaux OUT  
*CAUNDF* ! Ajout de UNDEF aux bouts des champs IN/OUT pour  
! vérifications ultérieures

*TE0000* ! Appel aux TE000I effectifs (calculs élémentaires)

*CAUNDF* ! Vérifications ultérieures des UNDEF  
*MONTEE* ! Recopie des champs locaux OUT dans les champs  
! globaux résultats

fin boucle grel

*MONTEE* ! Pour les volumes finis

si calcul parallèle distribué : *SDMPIC* ! Communication MPI

### 2.2 Organigrammes simplifiés de la routine EXTRAI

*EXTRAI* (LCHIN, ...)

```
! Dans le cas volumes finis, on doit allouer tous les champs avant la
! boucle sur les grels pour permettre l'accès aux « voisins »
si volumes finis et si on est avant la boucle sur les grels
  boucle sur les GREL du LIGREL
    EXTRA1
  fin boucle grel
sinon
  EXTRA1
EXTRA1 (LCHIN,...)
  boucle sur LCHIN : champ_in
  si TYPE(champ_in)='CARTE'
    EXCART(champ_in)
  si TYPE(champ_in)='CHAM_ELEM'
    EXCHML(champ_in)
  si TYPE(champ_in)='CHAM_NO'
    EXCHNO(champ_in)
  si TYPE(champ_in)='RESUELEM'
    EXRESL(champ_in)
  fin si
fin boucle sur LCHIN

EXCART (champ_in,...)
  EXCAR1

EXCAR1(champ_in,...)
  boucle sur les éléments du GREL : iel
    TRIGD ! récupération de la grandeur portée
          ! par la maille ima associée à l'élément iel
  fin boucle iel

EXCHNO (champ_in,...)
  boucle sur les éléments du GREL : iel
    boucle sur les noeuds de la maille associée à iel : ino
      TRIGD ! récupération de la grandeur portée
            ! par le noeud ino de l'élément iel
    fin boucle ino
  fin boucle iel

EXCHML (champ_in,...)
  boucle sur les éléments du GREL : iel
    JACOPO ! recopie du "bout" du champ global dans le champ local

  fin boucle iel

EXRESL (idem EXCHML)
```

## 2.3 Organigramme simplifié de la routine ALRSLT

```
ALRSLT (l_champ_out,...)
  boucle sur les champs de l_champ_out : chout
    DETRSD(chout) ! destruction du champ
    si TYPE(chout)='CHAM_ELEM'
      ALCHML(chout)
    si TYPE(chout)='RESUELEM'
      ALRESL(chout)
    fin si
  fin boucle chout
```

! Mise à jour du COMMON CAII07

## 3 Description sommaire des routines utilitaires

ALCHLO	alloue les champs locaux
ALCHML	alloue un champ résultat de type CHAM_ELEM
ALRESL	alloue un champ résultat de type RESUELEM
ALRSLT	alloue les champs globaux résultat
DCHLMX	rend la dimension max. d'un champ local
DEBCA1	"prologue" à la routine CALCUL
DEBCAL	"prologue" à la routine CALCUL
DIGDE2	rend la dimension d'un champ local
ETENCA	"étend" une CARTE : création des objets CARTE//'.PTMA' et CARTE//'.PTMS'
EXCART	extraie un champ local à partir d'une CARTE
EXCHML	extraie un champ local à partir d'un CHAM_ELEM
EXCHNO	extraie un champ local à partir d'un CHAM_NO
EXISDG	teste l'existence d'une CMP dans un DESCRIPTEUR_GRADEUR
EXRESL	extraie un champ local à partir d'un RESUELEM
EXTRAI	extraie un champ local à partir d'un champ global
GRDEUR	donne le nom de la grandeur associée à un paramètre
INIGRL	initialise les objets "&INEL.XXX" pour un GREL
INPARA	rend le numéro d'un paramètre pour un couple (type_element,option) donné
MAILLA	rend le nom du MAILLAGE associé à un LIGREL
MODATT	rend le mode local attendu par un type_element pour un paramètre donné
MONTEE	recopie un champ local résultat dans un champ global.
NBEC	rend le nombre d'entiers codés pour une grandeur donnée
NBELEM	rend le nombre d'éléments d'un GREL
NBGREL	rend le nombre de GREL d'un LIGREL
NBPARA	rend le nombre de paramètres pour un couple (type_element,option) donné
NOPARA	rend le nom d'un paramètre pour un couple (type_element,option) donné
NUCALC	rend le numéro de la routine TE000I pour un couple (type_element,option) donné
SCALAI	rend le type scalaire : R, I, C, ... d'une grandeur
TE0000	routine "chapeau" qui appelle tous les TE000I
TRIGD	trie les CMPS d'une grandeur suivant un DESCRIPTEUR_GRADEUR
TYPELE	rend le type_element associé à un GREL
ZECHLO	met un champ local à "zéro" entre 2 GRELS

## 4 Rappel sur les grandeurs

Nous appellerons "grandeur instanciée" (ou "grandeur" tout court pour alléger) :

- une référence à une "grandeur cataloguée" définie dans le catalogue des grandeurs,
- un vecteur de réels (ou de complexes, d'entiers, ...) dont les composantes sont associées aux CMPS de la grandeur cataloguée,
- un DESCRIPTEUR\_Grandeur : c'est un vecteur d'entiers codés qui renseigne sur la présence (ou l'absence) des CMPS de la grandeur cataloguée dans la grandeur instanciée

Par exemple, une champ de déplacements contient un ensemble de grandeurs instanciées. Chaque grandeur de ce champ est définie par :

- une référence à la grandeur DEPL\_R du catalogue : type réel, CMPS nommées : DX, DY, ...
- un vecteur de 2 réels (par exemple) : (2.3 , 3.4)
- un entier codé : ICODE=2\*\*2 + 2\*\*3 = 14 (par exemple), qui permet de dire que pour cette grandeur, DX est absent, DY vaut 2.3 et DZ vaut 3.4

Un champ (global ou local) est essentiellement une liste de grandeurs instanciées affectées à des entités géométriques :

- des nœuds pour un CHAM\_NO
- des mailles pour une CARTE
- es éléments finis pour un RESUELEM ou un CHAM\_ELEM

## 5 Cartes "étendues"

Une CARTE est un champ affecté par mailles ou groupes de mailles. La structure de données CARTE est "condensée" : on stocke en vis-à-vis de chaque grandeur instanciée la liste des mailles affectées par cette grandeur. Dans CALCUL, le problème de base pour l'utilisation des CARTES est le suivant : "comment retrouver la grandeur instanciée associée à la maille ima pour pouvoir la recopier dans le champ local fourni aux routines TE00IJ ?"

Ce problème ne peut pas être résolu efficacement avec la structure CARTE (surtout si l'on pense qu'une maille peut être affectée plusieurs fois : principe de "surcharger"), c'est pourquoi au début de CALCUL (routine DEBCAL), on "étend" les CARTES "IN" (routine ETENCA). Cette extension consiste à créer pour chaque CARTE, 2 objets supplémentaires temporaires qui permettent de faire l'association (maille, grandeur) une fois pour toutes.

Ces 2 objets correspondent à la SD CARTE\_ETENDUE suivante :

```
CARTE_ETENDUE (K19) ::= record
    '$VIDE'      : CARTE
    '.PTMA'     : OJB S V I LONG = nb_ma
    '.PTMS'     : OJB S V I LONG = nb_ms
```

Objet '.PTMA' :

soit nb\_ma le nombre de mailles du MAILLAGE associé à la carte,

pour ima=1, nb\_ma :

.PTMA(im) : numéro de la grandeur associée à la maille im

Objet '.PTMS' :

soit nb\_ms le nombre de mailles supplémentaires du LIGREL associé à la carte,

pour ims=1, nb\_ms :

.PTMA(ims) : numéro de la grandeur associée à la maille supplémentaire ims

## 6 Objets de travail

La routine CALCUL alloue des objets JEVEUX de travail (base VOLATILE) que nous allons décrire ci-dessous. Tous ces objets ont un nom commençant par '&&CALCUL'.

### 6.1 &&CALCUL.OBJETS\_TRAV OJB S V K24

Cet objet contient le nom de tous les objets de travail créés par CALCUL. Il sert à faire le "ménage" (destruction) à la fin de la routine CALCUL. Cet objet est créé par la routine DEBCAL. L'adresse de cet objet JEVEUX est stocké dans le COMMON CAPI05.

### 6.2 Champs locaux

On appelle "champ local" l'objet JEVEUX (et la zone mémoire associée) où sont stockées les informations extraites d'un champ global si "IN" (ou calculées par une routine TE000I si "OUT"). Ces informations sont "rangées" conformément à la description donnée dans le catalogue du type\_element (mode\_local du paramètre associé au champ).

Les champs locaux sont des vecteurs JEVEUX dont les noms sont de la forme '&&CALCUL'//nom\_paramètre, par exemple : '&&CALCUL.PGEOMER', '&&CALCUL.PCACQU', ...

Remarques :

- On continue à utiliser le vocabulaire "local" bien que depuis 1993, le champ "local" contienne (mis bout à bout) les informations concernant tous les éléments finis du GREL courant.
- Lorsqu'une routine TE00IJ fait par exemple :

CALL JEVECH ('PGEOMER','L','IAD)

l'adresse IAD rendue est l'adresse de l'objet '&&CALCUL.PGEOMER' décalée de ce qu'il faut pour que l'on atteigne les informations concernant l'élément courant.

L'organisation de ces objets est la suivante :

1ère CMP	de la 1ère grandeur	du 1er élément
2ème CMP	de la 1ère grandeur	du 1er élément
3ème CMP	de la 1ère grandeur	du 1er élément
1ère CMP	de la 2ème grandeur	du 1er élément
2ème CMP	de la 2ème grandeur	du 1er élément
...		
1ère CMP	de la 1ère grandeur	du 2ème élément
2ème CMP	de la 1ère grandeur	du 2ème élément
...		

On y trouve bout à bout les différents éléments du GREL (qui ont le même type\_element et donc le même mode\_local).

exemple pour un champ local de géométrie aux nœuds d'un GREL de TRIA3 :

DX	Noeud :1	du 1er élément
DY	Noeud :1	du 1er élément
DX	Noeud :2	du 1er élément
DY	Noeud :2	du 1er élément
DX	Noeud :3	du 1er élément
DY	Noeud :3	du 1er élément
DX	Noeud :1	du 2ème élément
DY	Noeud :1	du 2ème élément
...		

- Champs locaux de travail pour les conversions :

Pour les conversions, CALCUL a besoin de zones mémoire "tampon". Ces zones mémoire ont la même organisation que les champs locaux ci-dessus. Il est nécessaire d'en allouer 2 pour chaque type scalaire associés aux grandeurs des champs du calcul : I, R, C, K8, ...

Leur noms sont :

&&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

Pour la conversion d'un champ local de type réel (par exemple), la conversion utilisera en entrée le champ local : &&CALCUL.ORIGINAL.R et en sortie : &&CALCUL.CONVERTI.R

Ces champs locaux "tampon" sont alloués à la longueur max. des champs locaux "ordinaires".

## 6.3 Objets &&CALCUL.NOM\_&INEL et &&CALCUL.IAD\_&INEL

- &&CALCUL.NOM\_&INEL

Cet objet contient les **noms** des objets liés à l'initialisation du type\_element du GREL courant : objets '&INEL.XXX'.

- &&CALCUL.IAD\_&INEL

Cet objet contient les **adresses** des objets liés à l'initialisation du type\_element du GREL courant : objets '&INEL.XXX'.



## 6.4 Objets `&&CALCUL.TYPE_SCA` `&&CALCUL.IA_CHLOC` et `&&CALCUL.MODELO`

- `&&CALCUL.TYPE_SCA` S V K8 dim=nb\_para
- `&&CALCUL.IA_CHLOC` S V I dim=7\*nb\_para
- `&&CALCUL.MODELO` S V I dim=nb\_para

Soit `nb_para` le nombre de paramètres ("in" et "out") de l'option de calcul et `ipar` le numéro d'un tel paramètre.

<code>&amp;&amp;CALCUL.TYPE_SCA(ipar)</code>	type scalaire (I,R,C,...) de la grandeur associée au paramètre <code>ipar</code>
<code>&amp;&amp;CALCUL.IA_CHLOC(7*(ipar-1)+1)</code>	adresse du champ local associé à <code>ipar</code>
<code>&amp;&amp;CALCUL.IA_CHLOC(7*(ipar-1)+2)</code>	longueur du champ local associé à <code>ipar</code>
<code>&amp;&amp;CALCUL.IA_CHLOC(7*(ipar-1)+3)</code>	mode local attendu pour le paramètre
<code>&amp;&amp;CALCUL.IA_CHLOC(7*(ipar-1)+4)</code>	longueur utile du champ local pour 1 élément. (tient compte du <code>ICOEF</code> )
<code>&amp;&amp;CALCUL.IA_CHLOC(7*(ipar-1)+5)</code>	type du champ local : 1 : CARTE 2 : CHAM_NO 3 : CHAM_ELEM 4 : VECTEUR 2ND MEMBRE 5 : MATRICE
<code>&amp;&amp;CALCUL.IA_CHLOC(7*(ipar-1)+6)</code>	nombre de points de discrétisation du champ local : 1 pour une CARTE <code>nb_no</code> pour un CHAM_NO <code>nb_pg</code> pour un CHAM_ELEM <code>nb_no</code> pour un VECTEUR 0 pour une MATRICE
<code>&amp;&amp;CALCUL.IA_CHLOC(7*(ipar-1)+7)</code>	valeur du coefficient "multiplicateur" ( <code>ICOEF</code> ) pour les CHAM_ELEM à taille dynamique (modes locaux Zxxxxxx)
<code>&amp;&amp;CALCUL.MODELO(ipar)</code>	mode_local associé au paramètre <code>ipar</code>

## 6.5 Objets `&&CALCUL.LCHIN_EXI`, `&&CALCUL.LCHIN_K8` et `&&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

Soit nb\_in le nombre de champs "in" et iin =1, nb\_in

<code>&amp;&amp;CALCUL.LCHIN_EXI(iin)</code>	.FALSE. : le champ n'existe pas
<code>&amp;&amp;CALCUL.LCHIN_K8(2*(iin-1)+1)</code>	type du champ : 'CHNO', 'CART', 'CHML' OU 'RESL'
<code>&amp;&amp;CALCUL.LCHIN_K8(2*(iin-1)+2)</code>	type scalaire du champ : 'C', 'R', 'I', 'K8', ...
<code>&amp;&amp;CALCUL.LCHIN_I(11*(iin-1)+1)</code>	IGD : grandeur associée au champ
<code>&amp;&amp;CALCUL.LCHIN_I(11*(iin-1)+2)</code>	NEC : nombre d'entiers codés
<code>&amp;&amp;CALCUL.LCHIN_I(11*(iin-1)+3)</code>	NCMPMX : nombre max. de CMPS pour IGD
<code>&amp;&amp;CALCUL.LCHIN_I(11*(iin-1)+4)</code>	IADESC : adresse de CHIN//'.DESC'
<code>&amp;&amp;CALCUL.LCHIN_I(11*(iin-1)+5)</code>	IAVALE : adresse de CHIN//'.VALE'
<code>&amp;&amp;CALCUL.LCHIN_I(11*(iin-1)+6)</code>	IAPTMA : adresse de CHIN//'.PTMA' (si CARTE)
<code>&amp;&amp;CALCUL.LCHIN_I(11*(iin-1)+7)</code>	IAPTMS : adresse de CHIN//'.PTMS' (si CARTE)
<code>&amp;&amp;CALCUL.LCHIN_I(11*(iin-1)+8)</code>	IAPRN1 : adresse de .PRNO (\$MAILLA) (si CHAM_NO)
<code>&amp;&amp;CALCUL.LCHIN_I(11*(iin-1)+9)</code>	IAPRN2 : adresse de .PRNO (LIGREL) (si CHAM_NO)
<code>&amp;&amp;CALCUL.LCHIN_I(11*(iin-1)+10)</code>	IANUEQ : adresse de .NUEQ (si CHAM_NO)
<code>&amp;&amp;CALCUL.LCHIN_I(11*(iin-1)+11)</code>	1 : IANUEQ est valide 0 : sinon

## 6.6 Objets `&&CALCUL.LCHOU_K8` et `&&CALCUL.LCHOU_I`

- `&&CALCUL.LCHOU_K8` S V K8 dim= 2\*nb\_out
- `&&CALCUL.LCHOU_I` S V I dim= 2\*nb\_out

Soit nb\_out le nombre de champs "out" et iout =1, nb\_out

<code>&amp;&amp;CALCUL.LCHOU_K8(2*(iout-1)+1)</code>	type du champ : 'CHML' (CHAM_ELEM) ou 'RESL' (RESUELEM)
<code>&amp;&amp;CALCUL.LCHOU_K8(2*(iout-1)+2)</code>	type scalaire du champ : 'C', 'R'
<code>&amp;&amp;CALCUL.LCHOU_I(2*(iout-1)+1)</code>	IADESC : adresse de CHOUT//'.DESC'
<code>&amp;&amp;CALCUL.LCHOU_I(2*(iout-1)+2)</code>	IAVALE : adresse de CHOUT//'.VALE' (si CHAM_ELEM)

## 6.7 Objets &&CALCUL.SCALAIRE et &&CALCUL.IA\_CONVERS

- &&CALCUL.SCALAIRE S V K8 dim= nb\_scal
- &&CALCUL.IA\_CONVERS S V I dim= 2\*nb\_scal + 1

Soit nb\_scal le nombre de types scalaires : I, R, C, ...

&&CALCUL.SCALAIRE (i) i<sup>ème</sup> type scalaire possible des champs "in" du champ :  
SCAL (i)

&&CALCUL.IA_CONVERS (1)	nb_scal
&&CALCUL.IA_CONVERS (2*(i-1)+2)	adresse dans ZR ou ZC ou ... de l'objet : '&&CALCUL.ORIGINAL.'//SCAL (i)
&&CALCUL.IA_CONVERS (2*(i-1)+3)	adresse dans ZR ou ZC ou ... de l'objet : '&&CALCUL.CONVERTI.'//SCAL (i)

## 6.8 Objets &&CALCUL.TECAEL\_K24 et &&CALCUL.TECAEL\_I

- &&CALCUL.TECAEL\_K24 S V K24
- &&CALCUL.TECAEL\_I S V I

soit nb\_no le nombre de nœuds de la maille associée à l'élément courant.

&&CALCUL.TECAEL_K24 (1)	nom du MAILLAGE
&&CALCUL.TECAEL_K24 (2)	nom du LIGREL
&&CALCUL.TECAEL_K24 (3)	nom de la maille
&&CALCUL.TECAEL_K24 (3+1)	nom du 1er nœud de la maille
&&CALCUL.TECAEL_K24 (3+2)	nom du 2 <sup>ème</sup> nœud de la maille
...	...
&&CALCUL.TECAEL_I (1)	numéro de la maille
&&CALCUL.TECAEL_I (2)	nombre de nœuds de la maille (nb_no)
&&CALCUL.TECAEL_I (2+1)	numéro du 1er nœud de la maille
&&CALCUL.TECAEL_I (2+2)	numéro du 2 <sup>ème</sup> nœud de la maille
...	...

## 6.9 Objets &&CALCUL.NOMOP et &&CALCUL.NOMTE

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

&&CALCUL.NOMOP (iopt)	nom de l'option de numéro iopt
&&CALCUL.NOMTE (ite)	nom du type_element de numéro ite

## 7 Description des COMMONS propres à la routine CALCUL : ICOELXX

### 7.1 COMMONS ICOEL1 et ICOEL2

Description sommaire :

Informations concernant le champ global et le champ local associé au paramètre courant dans la boucle sur l'extraction.

```
COMMON /ICOEL1/IGD,NEC,NCMPMX,IACHIN,IACHLO,IICHIN,IANUEQ,LPRNO
COMMON /ICOEL2/TYPEGD
CHARACTER*8 TYPEGD
C
CES COMMONS SONT MIS A JOUR PAR EXTRAI.
C
IGD : NUMERO DE LA GRANDEUR ASSOCIEE AU CHAMP A EXTRAIRE
C
NEC : NOMBRE D'ENTIERES CODES DE IGD
C
NCMPMX: NOMBRE MAX DE CMPS POUR IGD
C
IACHIN: ADRESSE JEVEUX DE CHIN.VALE
C
IACHLO: ADRESSE JEVEUX DE CHLOC.VALE (&&CALCUL.NOMPAR)
C
IICHIN: NUMERO DU CHAMP CHIN DANS LA LISTE LCHIN.
C
IANUEQ: ADRESSE DE L'OBJET .NUEQ DU PROF_CHNO ASSOCIE EVENTUELLE
C
        -MENT AU CHAMP CHIN. (SI LPRNO=1).
C
LPRNO : 1-> L'OBJET .NUEQ EST A PRENDRE EN COMPTE
C
        (CHAM_NO A PROF_CHNO)
C
        0-> L'OBJET .NUEQ N'EST PAS A PRENDRE EN COMPTE
C
        (CHAM_NO A REPRESENTATION CONSTANTE OU AUTRE CHAMP)
C
TYPEGD: TYPE SCALAIRE DE LA GRANDEUR IGD : 'R', 'I', 'K8', ...
```

### 7.2 COMMON ICOEL3 , ICOELD , ICOELE et ICOELH

Description sommaire :

Informations concernant la conversion d'un champ local.

```
COMMON /ICOEL3/IAMOD1,IAMOD2,IAORIG,IACONV
C
CE COMMON EST MIS A JOUR PAR CONVER.
C
IAMOD1 : ADRESSE DU MODE-LOCAL AVANT CONVERSION
C
IAMOD2 : ADRESSE DU MODE-LOCAL APRES CONVERSION
C
IAORIG : ADRESSE DU CHAMP-LOCAL '&&CALCUL.ORIGINAL.SCAL'
C
IACONV : ADRESSE DU CHAMP-LOCAL '&&CALCUL.CONVERTI.SCAL'

COMMON /ICOELD/IACVRS
COMMON /ICOELE/IASCAL
C
CES COMMONS SONT MIS A JOUR PAR ALCHLO.
C
CES COMMONS SONT UTILISES PAR CONVER.
C
IASCAL : ADRESSE DE '&&CALCUL.SCALAIRE' V(K8)
C
        V(1),...,V(N) : TYPE_SCALAIRES POSSIBLES DES CHIN.
C
IACVRS : ADRESSE DE '&&CALCUL.IA_CONVERS' V(I)
C
        - DIM(V) = 2*DIM('&&CALCUL.SCALAIRE') +1
C
        - V(1) = DIMENSION DE '&&CALCUL.SCALAIRE'
C
        - POUR I =1,NB_SCALAIRE
C
        - V(1+ 2*(I-1)+1) = ADRESSE DANS ZR, OU ZC, OU ZI,...
C
        DE '&&CALCUL.ORIGINAL.SCAL(I)'
C
        - V(1+ 2*(I-1)+2) = ADRESSE DANS ZR, OU ZC, OU ZI,...
C
        DE '&&CALCUL.CONVERTI.SCAL(I)'

COMMON /ICOELH/PARAM
CHARACTER*8 PARAM
C
CE COMMON EST MIS A JOUR PAR CONVER ET EST UTILISE DANS LES
C
LES TE00IJ DE CONVERSION.
C
PARAM : NOM DU PARAMETRE A CONVERTIR
```

## 7.3 COMMON ICOEL4 et ICOEL7

Description sommaire :

Informations concernant les objets du catalogue des éléments finis "&CATA.XXX".

```
COMMON /ICOEL4/IAOPTT, LGCO, IAOPMO, ILOPNO, IAOPNO, ILOPNO, IAOPDS,  
+ IAOPPA, LGOPPA, IAMLOC, ILMLOC, IACOVE, ILCOVE, IADSGD  
C CE COMMON EST INITIALISE PAR DEBCAL  
C CE COMMON EST UTILISE UN PEU PARTOUT  
C IAOPTT : ADRESSE DE L'OBJET DU CATALOGUE : '&CATA.TE.OPTTE'  
C LGCO : LONGUEUR D'UNE COLONNE DE '&CATA.TE.OPTTE'  
C ( NOMBRE TOTAL D'OPTIONS POSSIBLES DU CATALOGUE)  
C IAOPMO : ADRESSE DE '&CATA.TE.OPTMOD'  
C ILOPNO : ADRESSE DU PT_LONG DE '&CATA.TE.OPTMOD'  
C IAOPNO : ADRESSE DE '&CATA.TE.OPTNOM'  
C ILOPNO : ADRESSE DU PT_LONG DE '&CATA.TE.OPTNOM'  
C IAOPDS : ADRESSE DE '&CATA.OP.DESCOPT(OPT)'  
C IAOPPA : ADRESSE DE '&CATA.OP.OPTPARA(OPT)'  
C LGOPPA : LONGUEUR DE '&CATA.OP.OPTPARA(OPT)'  
C IAMLOC : ADRESSE DE '&CATA.TE.MODELOC'  
C ILMLOC : ADRESSE DU PT_LONG DE '&CATA.TE.MODELOC'  
C IACOVE : ADRESSE DE '&CATA.TE.CONVERS'  
C ILCOVE : ADRESSE DU PT_LONG DE '&CATA.TE.CONVERS'  
C IADSGD : ADRESSE DE '&CATA.GD.DESCRIGD'  
  
COMMON /ICOEL7/IADFNO, IADFCA, LGCONO, LGCOCA  
C CE COMMON EST MIS A JOUR PAR DEBCAL  
C CE COMMON EST UTILISE DANS EXTRAI ET ALCHLO  
C IADFNO : ADRESSE DE '&CATA.TE.MODEFNO'  
C IADFCA : ADRESSE DE '&CATA.TE.MODEFCA'  
C LGCONO : LONGUEUR D'1 COLONNE DE MODEFNO.  
C LGCOCA : LONGUEUR D'1 COLONNE DE MODEFCA.
```

## 7.4 COMMON ICOEL5

Description sommaire :

Informations concernant les connectivités des mailles du maillage et des mailles tardives

```
COMMON /ICOEL5/IAMACO, ILMACO, IAMSCO, ILMSCO, IALIEL, ILLIEL  
C CE COMMON EST MIS A JOUR PAR DEBCAL (OU TERLIG)  
C CE COMMON EST UTILISE DANS NUMAIL, EXCHNO, ...  
C IAMACO : ADRESSE DE LA CONNECTIVITE DU MAILLAGE  
C ILMACO : ADRESSE DU POINTEUR DE LONGUEUR DE IAMACO  
C IAMSCO : ADRESSE DE LA CONNECTIVITE DES MAILLES SUPPL. D'1 LIGREL  
C ILMSCO : ADRESSE DU POINTEUR DE LONGUEUR DE IAMSCO  
C IALIEL : ADRESSE DE L'OBJET '.LIEL' DU LIGREL.  
C ILLIEL : ADRESSE DU POINTEUR DE LONGUEUR DE '.LIEL'.
```

## 7.5 COMMON ICOEL8

Description sommaire :

Informations concernant l'ensemble de tous les champs globaux "in".

```
COMMON /ICOEL8/IACHII, IACHIK, IACHIX
C      CE COMMON EST MIS A JOUR PAR DEBCAL
C      CE COMMON EST UTILISE DANS EXTRAI, EXCHNO, EXCART, EXRESL, EXCHML
C      , DCHLMX
C      IACHII : ADRESSE DE '&&CALCUL.LCHIN_I'
C      IACHIK : ADRESSE DE '&&CALCUL.LCHIN_K8'
C      IACHIX : ADRESSE DE '&&CALCUL.LCHIN_EXI'
C
C      '&&CALCUL.LCHIN_EXI' ::= V(L)      (DIM = NIN)
C      V(1) : .FALSE.      : LE CHAMP PARAMETRE N'EXISTE PAS.
C
C      '&&CALCUL.LCHIN_K8' ::= V(K8)      (DIM = NIN*2)
C      V(1) : TYPE_CHAMP : 'CHNO', 'CART', 'CHML' OU 'RESL'.
C      V(2) : TYPE_GD    : 'C', 'R', 'I', 'K8', ...
C
C      '&&CALCUL.LCHIN_I'  ::= V(I)      (DIM = NIN*11)
C      V(1) : IGD      GRANDEUR ASSOCIEE A LCHIN(I)
C      V(2) : NEC      NOMBRE D'ENTRIERS CODES
C      V(3) : NCMPMX   NOMBRE MAX DE CMP POUR IGD
C      V(4) : IADESC  ADRESSE DE .DESC
C      V(5) : IAVALE  ADRESSE DE .VALE
C      V(6) : IAPTMA  ADRESSE DE .PTMA (POUR 1 CARTE)
C      V(7) : IAPTMS  ADRESSE DE .PTMS (POUR 1 CARTE)
C      V(8) : IAPRN1  ADRESSE DU PRNO($MAILLA) (POUR 1 CHAM_NO)
C      V(9) : IAPRN2  ADRESSE DU PRNO(LIGREL) (POUR 1 CHAM_NO)
C      V(10) : IANUEQ ADRESSE .NUEQ (POUR 1 CHAM_NO)
C      V(11) : LPRNO  (DIT SI IANUEQ EST UTILISE POUR 1 CHAM_NO)
```

## 7.6 COMMON ICOEL9

Description sommaire :

Informations concernant les noms des options et des type\_element ainsi que la liste des objets de travail de la routine CALCUL.

```
COMMON /ICOEL9/IANOOP, IANOTE, NBOBTR, IAQBTR, NBOBMX
C      CE COMMON EST MIS A JOUR PAR DEBCAL
C      CE COMMON EST UTILISE DANS TE0000 POUR
C      IANOOP : ADRESSE DANS ZK16 DE '&&CALCUL.NOMOP' V(K16)
C      V(IOP) --> NOM DE L'OPTION IOP
C      IANOTE : ADRESSE DANS ZK16 DE '&&CALCUL.NOMTE' V(K16)
C      V(ITE) --> NOM DU TYPE_ELEMENT ITE
C      CE COMMON EST UTILISE DANS ALCHLO, ALRSLT ET CALCUL POUR :
C      NBOBTR : NOMBRE D'OBJETS DE TRAVAIL '&&CALCUL....' QUI
C      DEVRONT ETRE DETRUIES A LA FIN DE CALCUL.
C      IAQBTR : ADRESSE DANS ZK24 DE L'OBJET '&&CALCUL.OBJETS_TRAV'
C      NBOBMX : LONGUEUR DE L'OBJET '&&CALCUL.OBJETS_TRAV'
```

## 7.7 COMMON ICOELA

Description sommaire :

numéro du GREL courant, numéro de l'élément courant (dans le GREL), caractéristiques de l'ensemble des champs locaux.

```
COMMON /ICOELA/IAWMOL,NPARIO,IAWLOC,IAWTYP,NBELGR,IGR,IEL
C CE COMMON EST INITILISE PAR ALCHLO
C CE COMMON EST MODIFIE PAR MECOE1 (OBJET .IA_CHLOC)
C CE COMMON EST MODIFIE PAR EXTRAI,MONTEE,CALCUL,CONVER,ZECHLO
C (OBJET .MODELO)
C CE COMMON EST MODIFIE UN PEU PARTOUT POUR NBELGR,IGR,IEL
C CE COMMON EST UTILISE DANS EXTRAI,MONTEE,CALCUL,CONVER,
C JEVECH,ZECHLO,TECACH
C
C IGR : NUMERO DU GREL QUE L'ON TRAITE
C NBELGR : NOMBRE D'ELEMENTS DANS LE GREL IGR
C (IGR ET NBELGR SONT MIS A JOUR PAR CALCUL)
C IEL : NUMERO DE L'ELEMENT (DANS LE GREL IGR)
C (IEL EST MIS A JOUR PAR EXTRAI,TE0000,CONVER,MONTEE,...)
C IAWMOL : ADRESSE DANS ZI DE '&&CALCUL.MODELO' V(I)
C V(IPAR) --> MODE LOCAL DU PARAMETRE IPAR (OPT)
C NPARIO : LONGUEUR DE '&&CALCUL.MODELO' (NB DE PARAM IN/OUT (OPT))
C
C IAWLOC : ADRESSE DANS ZI DE '&&CALCUL.IA_CHLOC' V(I)
C CET OBJET CONTIENT DES INFORMATIONS SUR LES CHAMP_LOCAL
C V(7*(IPAR-1)+1) --> ADRESSE DU CHAMP_LOCAL '&&CALCUL.//NOMPAR(IPAR)
C V(7*(IPAR-1)+2) --> LONGUEUR DE L'OBJET '&&CALCUL.//NOMPAR(IPAR)
C L=MAX(TYPE_ELEM PRESENTS)*NBELGR
C V(7*(IPAR-1)+3) --> MODE LOCAL ATTENDU POUR LE PARAMETRE (IPAR)
C V(7*(IPAR-1)+4) --> LONGUEUR UTILE DU CHAMP_LOCAL POUR 1 ELEMENT
C DU TYPE_ELEM ASSOCIE AU GREL IGR : NCMPEL(MODE)
C CETTE LONGUEUR TIEN COMPTE D'UN EVENTUEL ICOEF
C =/ 1 (CF V(7*(IPAR-1)+7)).
C V(7*(IPAR-1)+5) --> TYPE DE CHAMP_LOCAL :
C 1 : CARTE
C 2 : CHAM_NO
C 3 : CHAM_ELEM
C 4 : VECTEUR 2ND MEMBRE
C 5 : MATRICE
C V(7*(IPAR-1)+6) --> NOMBRE DE POINTS DE DISCRETISATION DU CHAMP_LOC
C (0 SI MATRICE)
C V(7*(IPAR-1)+7) --> VALEUR DU COEFFICIENT MULTIPLICATEUR POUR LES
C MODES LOCAUX DE CHAM_ELEM "ZXXXX"(ICOEF).
C
C IAWTYP : ADRESSE DANS ZK8 DE '&&CALCUL.TYPE_SCA' V(K8)
C V(IPAR) --> TYPE_SCALAIRE DU CHAMP_LOCAL
```

## 7.8 COMMON ICOELC

Description sommaire :

Informations concernant l'ensemble de tous les champs globaux "out".

```
COMMON /ICOELC/IACHOI,IACHOK
C CE COMMON EST MIS A JOUR PAR ALRSLT
C CE COMMON EST UTILISE DANS MONTEE ,DCHLMX
C IACHOI : ADRESSE DE '&&CALCUL.LCHOU_I'
C IACHOK : ADRESSE DE '&&CALCUL.LCHOU_K8'
C
C '&&CALCUL.LCHOU_K8' ::= V(K8) (DIM = NIN*2)
C V(1) : TYPE_CHAMP : 'CHML' OU 'RESL'.
C V(2) : TYPE_GD : 'C', 'R'
C
```

```
C      '&&CALCUL.LCHOU_I'  ::= V(I)      (DIM = NOUT*2)
C          V(1) :  ADRESSE DE L_CHOUT(I).DESC
C          V(2) :  ADRESSE DE L_CHOUT(I).VALE (SI CHAM_ELEM)
C
```

## 7.9 COMMON ICOELF

Description sommaire :

Informations concernant l'ensemble des objets liés à l'initialisation des type\_element : "&INEL.XXX"

```
COMMON /ICOELF/NBOBJ, IAINEL, ININEL
C      NBOBJ :  NOMBRE D'OBJETS '&INEL.XXXX' CREE PAR L'INITIALISATION
C              DU TYPE_ELEM
C      ININEL :  ADRESSE DANS ZK24 DE L'OBJET '&&CALCUL.NOM_&INEL'
C              QUI CONTIENT LES NOMS DES OBJETS '&INEL.XXXX'
C      IAINEL :  ADRESSE DANS ZI DE L'OBJET '&&CALCUL.IAD_&INEL'
C              QUI CONTIENT LES ADRESSES DES OBJETS '&INEL.XXXX'
C      CE COMMON EST INITIALISE PAR DEBCAL
C      CE COMMON EST UTILISE PAR CALCUL ET JEVETE
```

## 7.10 COMMON ICOELG

Description sommaire :

Informations concernant la maille sous-jacente à l'élément courant

```
COMMON /ICOELG/ICAELI, ICAELK
C      CE COMMON EST CREE PAR DEBCAL.
C      IL EST UTILISE PAR TECAEL
C      ICAELK EST L'ADRESSE D'UN VECTEUR DE K24 CONTENANT :
C      V(1) :  NOM DU MAILLAGE      (K8)
C      V(2) :  NOM DU LIGREL        (K19)
C      V(3) :  NOM DE LA MAILLE     (K8)
C      V(3+ 1) :  NOM DU 1ER NOEUD DE LA MAILLE
C      V(3+ 1) :  NOM DU DER NOEUD DE LA MAILLE
C      ICAELI EST L'ADRESSE D'UN VECTEUR DE IS CONTENANT :
C      V(1) :  NUMERO DE LA MAILLE
C      V(2) :  NOMBRE DE NOEUDS DE LA MAILLE (NBNO)
C      V(2+ 1) :  NUMERO DU 1ER NOEUD DE LA MAILLE
C      V(2+NBNO) :  NUMERO DU DER NOEUD DE LA MAILLE
C      V(2+NBNO +1) :  NUMERO DU GREL
C      V(2+NBNO +2) :  NUMERO DE L'ELEMENT DANS LE GREL
```



## 8 Usage des COMMONS : ICOELXX

### 8.1 Liste des routines utilisant les COMMONS ICOELXX

Ces routines ne doivent pas être appelées en dehors de la routine CALCUL

ALCHLO	EXRESL	NBPARA
ALRSLT	EXTRAI	NOPARA
CALCUL	GRDEUR	NUCALC
CONVER	INPARA	OPCONV
DCHLMX	JEVECH	TE0000
DEBCAL	JEVETE	TECACH
DIGDE2	MECOE1	TECAEL
EXCART	MECOEL	TRIGD
EXCHML	MODATT	ZECHLO
EXCHNO	MONTEE	
EXPAND	MOYENN	

### 8.2 Usage des COMMONS ICOELXX

Common block cross-reference list:

```
Common Block ICOEL1 used in:
EXCART      EXCHML      EXCHNO      EXRESL      EXTRAI      TRIGD
Common Block ICOEL2 used in:
CONVER      EXCHML      EXPAND      EXRESL      EXTRAI      MOYENN
TRIGD
Common Block ICOEL3 used in:
CONVER      EXPAND      MOYENN      (+ routines TEOOIJ de conversion)
Common Block ICOEL4 used in:
CALCUL      CONVER      DEBCAL      DIGDE2      EXCART      EXCHNO
EXTRAI      GRDEUR      INPARA      JEVECH      MECOE1      MODATT
MONTEE      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:
CALCUL      DCHLMX      DEBCAL      EXCART      EXCHML      EXCHNO
EXRESL      EXTRAI
Common Block ICOEL9 used in:
ALCHLO      ALRSLT      CALCUL      DEBCAL      TE0000
Common Block ICOELA used in:
ALCHLO      CALCUL      CONVER      EXCART      EXCHML      EXCHNO
EXPAND      EXRESL      EXTRAI      JEVECH      MECOE1      MONTEE
MOYENN      TE0000      TECACH      TECAEL      ZECHLO
Common Block ICOELC used in:
ALRSLT      DCHLMX      MONTEE
Common Block ICOELD used in:
ALCHLO      CONVER
Common Block ICOELE used in:
ALCHLO      CONVER
Common Block ICOELF used in:
CALCUL      DEBCAL      JEVETE
Common Block ICOELG used in:
DEBCAL      TECAEL
Common Block ICOELH used in:
CONVER      (+ routines TEOOIJ de conversion)
```