

Manuel d'Utilisation
Fascicule U4.CB : CABRI
Document : U4.CB.20

Macro commande MACR_CABRI_CALC

1 But

Cette macro permet le calcul thermo-mécanique d'une jonction boulonnée de tuyauterie.
Les principales étapes de la macro commande sont :

- Affectation des modèles mécanique et thermique par la commande AFPE_MODELE.
- Affectation des matériaux par la commande AFPE_MATERIAU.
- Définition du chargement mécanique (pression, effet de fond, torseur d'effort, déformation d'origine thermique) par la commande AFPE_CHAR_MECA.
- Définition du chargement thermique (température du fluide, coefficient d'échange) par la commande AFPE_CHAR_THER_F.
- Réalisation du calcul thermique linéaire et du calcul mécanique linéaire ou non linéaire par les commandes THER_LINEAIRE et STAT_NON_LINE.

Produit un concept de type evol_noli.

Le concept produit par cette macro commande est de type evol_noli, et il est possible de récupérer un certain nombre de concepts créés à l'intérieur de la macro-commande comme les modèles (thermiques et mécaniques), les chargements (thermiques et mécaniques), le champ de matériau et les résultats thermiques (en sus des résultats mécaniques, retournés par le concept sortant de la macro).

Le maillage produit utilisé peut être produit par la macro commande MACR_CABRI_MAIL.

Les longueurs doivent être données en millimètres et les angles en degrés. En tenir compte pour la définition des caractéristiques matériaux (module d'Young en MPa, etc ...).

Table des matières

1 But 1	
2 Syntaxe	4
3 Définition géométrique de la bride	7
3.1 Composants de la jonction boulonnée	7
3.2 Surfaces.....	8
4 Opérandes d'entrées	8
4.1 Mot clé MAILLAGE	8
4.2 Mot clé facteur AFFE_MATERIAU.....	8
4.2.1 Opérande TOUT, GROUP_MA.....	8
4.2.2 Opérande MATER.....	9
4.2.3 Opérande TEMP_REF	9
4.3 Mot clé facteur DEFI_CHAR_THER.....	9
4.3.1 Opérande TEMP_INIT	9
4.3.2 Opérandes COEF_H_FLUI et TEMP_EXT_FLUI.....	9
4.3.3 Opérandes COEF_H_AIR et TEMP_EXT_AIR.....	9
4.3.4 Opérande LIST_INST	10
4.4 Mot clé facteur DEFI_CHAR_MECA.....	10
4.4.1 Opérande PRETENS	10
4.4.2 Opérande PRES_REP	10
4.4.3 Opérande EFFE_FOND	10
4.5 Opérande RELATION	11
4.6 Opérande SOLVEUR.....	11
4.6.1 Opérande METHODE	11
4.6.2 METHODE : 'MULT_FRONT'	11
4.7 Opérande CONVERGENCE	12
4.7.1 Opérande RESI_GLOB_RELA / RESI_GLOB_MAXI	12
4.7.2 Opérande ITER_GLOB_MAXI	13
4.7.3 Opérande ITER_GLOB_ELAS	13
4.7.4 Opérande RESO_INTE	13
4.8 Opérande NEWTON.....	13
4.8.1 Opérande PREDICTION.....	13
4.8.2 Opérande MATRICE	15
4.8.3 Opérande EVOL_NOLI	15
4.9 Opérande INCREMENT	16
4.9.1 Opérandes LIST_INST / EVOLUTION	16
4.9.2 Opérandes NUME_INST_INIT / INST_INIT / NUME_INST_FIN / INST_FIN.....	16
4.9.3 Opérande PRECISION	Erreur ! Signet non défini.

4.9.4 Opérande SUBD_METHODE	17
4.9.5 Opérande SUBD_PAS / SUBD_PAS_MINI / SUBD_COEF_PAS_1	17
4.9.6 Opérande OPTI_LIST_INST / NOM_CHAM / NOM_CMP / VALEUR.....	18
5 Opérandes de sortie	19
5.1 Mot clé MODELE_THER	19
5.2 Mot clé facteur CHAR_THER	19
5.2.1 Opérande CHARGE	19
5.2.2 Opérande TYPE.....	19
5.3 Mot clé RESU_THER	19
5.4 Mot clé MODELE_MECA	19
5.5 Mot clé facteur CHAR_MECA	19
5.5.1 Opérande CHARGE	20
5.5.2 Opérande TYPE.....	20
5.6 Opérande CHAM_MATER.....	20

2 Syntaxe

```

resu [evol_noli] = MACR_CABRI_CALC
(
  ♦ MAILLAGE = mail [maillage]

  ♦ AFFE_MATERIAU = _F(
    ♦ / TOUT = 'OUI' [TXM]
    / GROUP_MA =
      /BRIDE [TXM]
      /GOUJON
      /RONDELLE
      /ECROU
      /JOINT
    ♦ MATER = materiau [mater]
    ♦ TEMP_REF = /20 [DEFAULT]
      /Tref [R]
  ),
  ♦ CHAM_MATER = CO cham_mater [TXM]
  ♦ MODELE_THER = CO modele [TXM]
  ♦ MODELE_MECA = CO modele [TXM]
  ♦ DEFI_CHAR_THER = _F(
    ♦ TEMP_INIT = /25 [DEFAULT]
      /temp_init [R]
    ♦ COEF_H_FLUI = coef_h_flui [fonction]
    ♦ TEMP_EXT_FLUI = temp_ext_flui [fonction]
    ♦ COEF_H_AIR = coef_h_air [fonction]
    ♦ TEMP_EXT_AIR = temp_ext_air [fonction]
    ♦ LIST_INST = list_inst [listr8]
  ),
  ♦ CHAR_THER = _F(
    ♦ CHARGE = CO char_ther [TXM]
    ♦ TYPE = /BRIDE_FLUIDE [TXM]
      /BRIDE_AIR
      /ECROU_GOUJON
      /BRIDE_JOINT
  ),
  ♦ RESU_THER = CO evol_ther [TXM]

  ♦ DEFI_CHAR_MECA = _F(
    ♦ PRETENS = pretens [fonction]
    ♦ PRES_REP = pres_rep [fonction]
    ♦ EFFE_FOND = effe_fond [fonction]
  ),
  ♦ CHAR_MECA = _F(
    ♦ CHARGE = CO char_meca [TXM]
    ♦ TYPE = /BLOC_BAS_GOUJ [TXM]
      /BLOC_BAS_JOINT
      /BLOC_LAT_ALES
      /BLOC_LAT_NALES
      /PLAN_TUBE
      /PRES_FLU
      /EFFET_FOND
      /CONT_JOINT
      /DEFO_THER
      /SERR_ECROU_1
      /SERR_ECROU_2
  ),

```

Titre : Macro commande MACR_CABRI_CALC
Auteur(s) : M. ABBAS

Date : 14/06/07
Clé : U4.CB.20-B Page : 5/20

```

◇ RELATION      = /VMIS_ISOT_TRAC                      [TXM]
                  /ELAS
                  /ELAS_VMIS_TRAC

◇ SOLVEUR        = _F (

                  # Factorisation de type "multi-frontale" :

                  /  METHODE = 'MULT_FRONT' ,           [DEFAULT]
                    ◇ RENUM      = / 'METIS' , [DEFAULT]
                      / 'MD' ,
                      / 'MDA' ,
                    ◇ STOP_SINGULIER = / 'OUI' , [DEFAULT]
                      / 'NON' ,
                    ◇ NPREC       = / 8 , [DEFAULT]
                      / nprec , [I]

                  ),

◇ INCREMENT      =_F (
                    ◇ LIST_INST = litps,                [listr8]
                    ◇ EVOLUTION = / 'CHRONOLOGIQUE',    [DEFAULT]
                    ◇ / NUME_INST_INIT = nuini,          [I]
                      / INST_INIT      = instini,        [R]
                    ◇ / NUME_INST_FIN  = nufin,          [I]
                      / INST_FIN       = instfin,        [R]
                    ◇ PRECISION = / 1.0E-3,             [DEFAULT]
                      / prec,          [R]
                    ◇ SUBD_METHODE = / 'AUCUNE',         [DEFAULT]
                      / 'UNIFORME',
                      / 'EXTRAPOLE',
                    ◇ SUBD_PAS = / 1,                   [DEFAULT]
                      / subpas,         [I]
                    ◇ SUBD_PAS_MINI = submini,          [R]
                    ◇ SUBD_COEF_PAS_1 = / 1.,            [DEFAULT]
                      / coefsub,        [R]
                    ◇ OPTI_LIST_INST = / 'INCR_MAXI',    [DEFAULT]
                    ◇ NOM_CHAM      = nomch,            [Kn]
                    ◇ NOM_CMP       = nomcmp,           [Kn]
                    ◇ VALEUR        = val               [R]
                  ),

◇ NEWTON          =_F (
                    ◇ PREDICTION =
                      / 'TANGENTE', [DEFAULT]
                      / 'ELASTIQUE',
                      / 'EXTRAPOL',
                    ◇ EVOL_NOLI = evol_noli,             [evol_noli]
                    ◇ MATRICE   = / 'TANGENTE', [DEFAULT]
                      / 'ELASTIQUE'
                    ◇ REAC_INCR = / 1,                   [DEFAULT]
                      / mf,         [I]
                    ◇ REAC_ITER = / 0,                   [DEFAULT]
                      / it,         [I]
                    ◇ PAS_MINI_ELAS= / 0,                [DEFAULT]
                      / pasmini,    [I]
                  ),

```

```
◇ CONVERGENCE =_F (
    ◇ / RESI_GLOB_RELA      = 1.E-6,          [DEFAULT]
      / | RESI_GLOB_MAXI    = resmax,          [R]
      | RESI_GLOB_RELA      = resrel,          [R]
    ◇ SIGM_REFE             = sigm_refe,        [R]
    ◇ EPSI_REFE             = epsi_refe,        [R]
    ◇ FLUX_THER_REFE        = ther_refe,        [R]
    ◇ RESI_REFE_RELA        = sigm_refe_rela,    [R]
    ◇ ITER_GLOB_ELAS        = / 25,             [DEFAULT]
      / maxelas,           [I]
    ◇ ITER_GLOB_MAXI        = / 10,             [DEFAULT]

    ◇ TYPE_MATR_COMP        = / 'TANG_VIT'      [DEFAULT]
    ◇ RESO_INTE             = / 'IMPLICITE'     [DEFAULT]
)
```

3 Définition géométrique de la bride

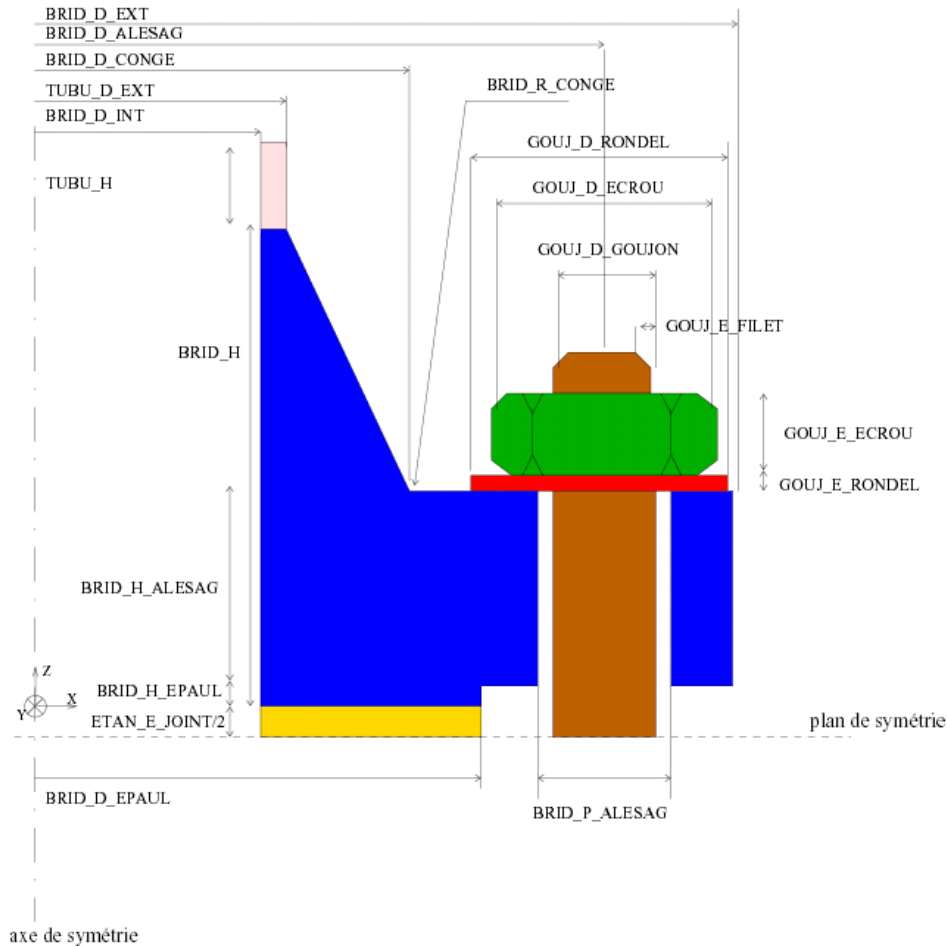


Figure 3-a : Description des différents paramètres de la jonction boulonnée

Le maillage de la bride est possible avec la macro MACR_CABRI_MAIL.

On rappelle les différents groupes de mailles et de nœuds nécessaires pour l'usage de MACR_CABRI_CALC.

3.1 Composants de la jonction boulonnée

La jonction boulonnée est constituée de cinq composants dont les groupes de mailles sont les suivants :

- BRIDE
- ROND
- GOUJON
- JOINT
- ECROU

Et les groupes de nœuds associés :

- P_BRI
- P_GOU
- P_GOUI, surface intérieure du goujon
- P_GOUE, surface extérieure du goujon

3.2 Surfaces

Les surfaces sur lesquelles s'effectuent les échanges thermiques sont les groupes de mailles suivants :

- `M_INT`, surface interne de la bride (échange fluide/bride)
- `M_EXT`, surface externe de la bride (échange air ambiant/bride)

Pour les échanges thermiques entre écrou et goujon, deux groupes de nœuds sont définis, `N_SCEG` et `N_SCGE`.

Pour les échanges thermiques entre bride et joint, deux groupes de nœuds sont définis, `N_SCJB` et `N_SCBJ`.

Les conditions limites mécaniques agissent sur des surfaces définies par ces groupes de nœuds :

- `N_M_GOU`, bas du goujon
- `N_M_JOI`, bas du joint
- `N_M_L_AA`, face latérale avec alésage

Par des groupes de mailles

- `M_L_SA`, face latérale sans alésage
- `M_TUB`, face de coupe inférieure du tube

4 Opérandes d'entrées

4.1 Mot clé **MAILLAGE**

- ♦ `MAILLAGE = mail` [maillage]

On précise ici le maillage de la jonction boulonnée utilisée.

4.2 Mot clé facteur **AFFE_MATERIAU**

- ♦ `AFFE_MATERIAU`

Mot clé facteur permettant d'affecter les matériaux sur des parties du maillage.

4.2.1 Opérande **TOUT, GROUP_MA**

- ♦ `/ TOUT = 'oui'` [TXM]
`/ GROUP_MA = / BRIDE` [TXM]
`/ GOUJON`
`/ RONDELLE`
`/ ECROU`
`/ JOINT`

Ces mots clés permettent d'affecter les matériaux sur toutes les mailles du maillage (`TOUT`), ou sur une partie du maillage (`GROUP_MA`).

Pour les assemblages boulonnés, on peut affecter :

- `BRIDE` : la bride
- `GOUJON` : les goujons
- `RONDELLE` : la rondelle de la goujonnerie
- `ECROU` : l'écrou de la goujonnerie
- `JOINT` : le joint au niveau de l'interface d'étanchéité

4.2.2 Opérande MATER

♦ MATER = materiau [mater]
Nom de matériau que l'on veut affecter

4.2.3 Opérande TEMP_REF

◇ TEMP_REF= /20 [DEFAULT]
/Tref [R]
Température de référence pour laquelle il n'y a pas de déformation thermique (vaut 20°C par défaut)

4.3 Mot clé facteur DEFI_CHAR_THER

◇ DEFI_CHAR_THER
Mot clé facteur permettant d'affecter le chargement thermique.

4.3.1 Opérande TEMP_INIT

◇ TEMP_INIT = /25 [DEFAULT]
/temp_init [R]
Température initiale de la bride. Par défaut, cette température est de 25°.

4.3.2 Opérandes COEF_H_FLUI et TEMP_EXT_FLUI

◇ COEF_H_FLUI = coef [fonction]
◇ TEMP_EXT_FLUI = temp [fonction]
Opérandes permettant de caractériser l'échange thermique entre la bride et le fluide. COEF_H_FLUI est le coefficient d'échange de paroi et TEMP_EXT_FLUI la température du fluide. Ces deux opérandes sont des fonctions du temps.
Les unités sont celles du mot clé 'ECHANGE' d'AFFE_CHAR_THER.

4.3.3 Opérandes COEF_H_AIR et TEMP_EXT_AIR

◇ COEF_H_AIR = coef [fonction]
◇ TEMP_EXT_AIR = temp [fonction]
Opérandes permettant de caractériser l'échange thermique entre la bride et l'air ambiant. COEF_H_AIR est le coefficient d'échange de paroi et TEMP_EXT_AIR la température ambiante. Ces deux opérandes sont des fonctions du temps.
Les unités sont celles du mot clé 'ECHANGE' d'AFFE_CHAR_THER.

4.3.4 Opérande LIST_INST

◇ LIST_INST = list_inst [listr8]

Permet de définir les instants de calcul qui déterminent les intervalles de temps pris pour intégrer l'équation différentielle pour le problème thermique. Par défaut, cette liste est la suivante :

```
list_inst = DEFI_LIST_REEL(DEBUT=0.0,  
                           INTERVALLE=( _F(JUSQU_A=1.0,  
                                           NOMBRE=1, ),  
                                           _F(JUSQU_A=11.0,  
                                           NOMBRE=10, ),  
                                           _F(JUSQU_A=600.0,  
                                           NOMBRE=10, ),  
                                           _F(JUSQU_A=610.0,  
                                           NOMBRE=30, ),  
                                           _F(JUSQU_A=1800.0,  
                                           NOMBRE=30, ),  
                                           _F(JUSQU_A=7200.0,  
                                           NOMBRE=10, ), ),
```

4.4 Mot clé facteur DEFI_CHAR_MECA

◆ DEFI_CHAR_MECA

Mot clé facteur permettant d'affecter le chargement mécanique.

4.4.1 Opérande PRETENS

◇ PRETENS = pretens [fonction]

Fonction permettant de simuler le serrage initial de la bride. Par défaut, cette fonction est :

```
pretens=DEFI_FONCTION(NOM_PARA='INST',  
                      VALE=(0.0,0.0,1.0,-0.02, ),  
                      PROL_DROITE='CONSTANT',  
                      PROL_GAUCHE='CONSTANT', )
```

Le serrage est défini par le déplacement de la bride (donc en mm).

4.4.2 Opérande PRES_REP

◇ PRES_REP = pression [fonction]

Fonction permettant d'appliquer la pression du fluide à l'intérieur de la bride. Par défaut, cette fonction est :

```
pression = DEFI_FONCTION(NOM_PARA='INST',  
                         VALE=(0.0,0.0,1.0,0.0,11.0,16.0, ),  
                         PROL_DROITE='CONSTANT',  
                         PROL_GAUCHE='CONSTANT', )
```

4.4.3 Opérande EFFE_FOND

◇ EFFE_FOND = effe_fond [fonction]

Fonction permettant de simuler l'effet de fond sur la tuyauterie. Par défaut, cette fonction est :

```
effe_fond=DEFI_FONCTION(NOM_PARA='INST',  
                        VALE=(0.0,-0.0,1.0,-0.0,11.0,-20.607059, ),  
                        PROL_DROITE='CONSTANT',  
                        PROL_GAUCHE='CONSTANT', )
```

4.5 Opérande RELATION

```

◇ RELATION      =  /VMIS_ISOT_TRAC      [TXM]
                   /ELAS
                   /ELAS_VMIS_TRAC

```

Cet opérande permet de définir les relations de comportement. On peut avoir soit un comportement incrémental de type Von Mises avec écrouissage isotrope (VMIS_ISOT_TRAC), soit un comportement élastique pur (ELAS), soit un comportement élastique non linéaire (ELAS_VMIS_TRAC).

Dans le cas d'un comportement non linéaire (VMIS_ISOT_TRAC et ELAS_VMIS_TRAC), il est nécessaire que le matériau affecté aux mailles soit cohérent avec DEFI_MATERIAU.

4.6 Opérande SOLVEUR

4.6.1 Opérande METHODE

```

◇ METHODE =

```

Ce mot clé permet de choisir la méthode de résolution des systèmes linéaires :

```

/ 'MULT_FRONT'  méthode directe "multi-frontale". Le stockage est 'MORSE'.
                  Cette méthode est parallélisée et peut être exécutée sur plusieurs
                  processeurs (via l'interface Astk).

```

Pour la macro-commande, seule la méthode MULT_FRONT est possible.

4.6.2 METHODE : 'MULT_FRONT'

```

◇ RENUM =

```

Cet argument permet de renuméroter les nœuds du modèle :

```

/ 'MD'          ("Minimum Degré") cette numérotation des nœuds minimise le remplissage de la
                  matrice lors de sa factorisation.

/ 'MDA'         ("Minimum Degré Approché") cette numérotation est en principe moins optimale
                  que 'MD' en ce qui concerne le remplissage mais elle est plus économique à
                  calculer. Elle est toutefois préférable à 'MD' pour les gros modèles (≥ 50 000 ddls).

/ 'METIS'       Autre méthode de numérotation basée sur une dissection emboîtée. C'est la
                  méthode la plus efficace (en temps CPU et en mémoire).

```

```

◇ STOP_SINGULIER = 'OUI' / 'NON'

```

Lorsqu'au terme de la factorisation, on constate qu'un terme diagonal d' est devenu très petit (par rapport à ce qu'il était avant la factorisation d), c'est que la matrice est presque singulière.

$$\text{Soit } n = \log \left| \frac{d}{d'} \right|$$

ce nombre n indique que sur une équation (au moins) on a perdu n chiffres significatifs.

Si $n > \text{nprec}$ (mot clé NPREC ci-dessous), on considère que la matrice est singulière.

Si l'utilisateur a indiqué : `STOP_SINGULIER = 'OUI'`, le code s'arrête en erreur fatale, sinon l'exécution se poursuit avec émission d'une alarme.

Remarque :

Toute perte importante de chiffres significatifs lors d'une factorisation est un indicateur d'un problème mal posé. Plusieurs causes sont possibles (liste non exhaustive) :

- *des conditions aux limites de blocage de la structure insuffisantes,*
- *des relations linéaires redondantes,*
- *des données numériques très hétérogènes (termes de pénalisation trop grands), ...*

◇ `NPREC = nprec`

C'est le nombre qui sert à déterminer si la matrice est singulière (ou non) (cf. mot clé `STOP_SINGULIER` ci-dessus).

4.7 Opérande CONVERGENCE

◇ `CONVERGENCE`

Si aucun des deux opérandes suivants n'est présent, alors tout se passe comme si : `RESI_GLOB_RELA = 1.E-6`.

4.7.1 Opérande `RESI_GLOB_RELA` / `RESI_GLOB_MAXI`

◇ | `RESI_GLOB_RELA = resrel`

L'algorithme continue les itérations globales tant que :

$$\max_{i=1,\dots,nb_ddl} |F_i^n| > \text{resrel} \max |\mathbf{L}|$$

où F^n est le résidu de l'itération n et \mathbf{L} le vecteur du chargement imposé et des réactions d'appuis (Cf. [R5.03.01] pour plus de détails).

Lorsque le chargement et les réactions d'appui deviennent nuls, c'est-à-dire lorsque \mathbf{L} est nul (par exemple dans le cas d'une décharge totale), on passe du critère de convergence relatif au critère de convergence absolu `RESI_GLOB_MAXI`. Cette opération est transparente pour l'utilisateur (message d'alarme émis dans le fichier .mess). Lorsque le vecteur \mathbf{L} redevient différent de zéro, on repasse automatiquement au critère de convergence relatif `RESI_GLOB_RELA`.

Si cet opérande est absent, le test est effectué avec la valeur par défaut, sauf si `RESI_GLOB_MAXI` est présent.

◇ | `RESI_GLOB_MAXI = resmax`

L'algorithme continue les itérations globales tant que :

$$\max_{i=1,\dots,nb_ddl} |F_i^n| > \text{resmax}$$

où F^n est le résidu de l'itération n (Cf. [R5.03.01] pour plus de détails).

Si cet opérande est absent, le test n'est pas effectué.

Si `RESI_GLOB_RELA` et `RESI_GLOB_MAXI` sont présents tous les deux, les deux tests sont effectués.

4.7.2 Opérande ITER_GLOB_MAXI

```
◇ ITER_GLOB_MAXI = / 10 [DEFAULT]
                  / maglob
```

Nombre d'itérations maximum effectué pour résoudre le problème global à chaque instant (10 par défaut). Ce test est toujours effectué.

4.7.3 Opérande ITER_GLOB_ELAS

```
◇ ITER_GLOB_ELAS = / 25 [DEFAULT]
                  / maxelas
```

Nombre d'itérations maximum effectué avec la matrice élastique lorsqu'on utilise le mot clé PAS_MINI_ELAS du mot clé facteur NEWTON (voir [§3.8.2]) pour résoudre le problème global à chaque instant (25 par défaut).

On rappelle que PAS_MINI_ELAS permet de passer de la matrice tangente à la matrice élastique lorsque le pas de temps est ou devient (par le redécoupage) inférieur à une certaine valeur précisée sous PAS_MINI_ELAS.

4.7.4 Opérande RESO_INTE

```
◇ RESO_INTE = / 'IMPLICITE' [DEFAULT]
```

Permet de préciser le type de schéma d'intégration pour résoudre le système d'équations non linéaires formé par les équations constitutives des modèles de comportement à variables internes.

Pour la macro-commande, on se limite au schéma implicite.

4.8 Opérande NEWTON

```
◇ NEWTON
```

Précise les caractéristiques de la méthode de résolution du problème incrémental non linéaire (méthode de NEWTON-RAPHSON).

4.8.1 Opérande PREDICTION

```
◇ PREDICTION =
/ 'TANGENTE'
/ 'ELASTIQUE'
/ 'EXTRAPOL'
/ 'DEPL_CALCULE'
```

La phase de prédiction (Cf. [R5.03.01]) a pour but de calculer une estimation du champ de déplacements afin de permettre à la méthode de NEWTON de converger plus rapidement.

Lorsque le mot clé est absent, c'est la matrice tangente en vitesse (option RIGI_MECA_TANG dans le fichier .mess) qui est utilisée si l'on a choisi pour la méthode de NEWTON une MATRICE: 'TANGENTE', et c'est la matrice élastique (option RIGI_MECA dans le fichier .mess) qui est utilisée si on a choisi MATRICE: 'ELASTIQUE'.

```
/ 'TANGENTE'
```

On utilise la matrice tangente du problème en vitesse (option RIGI_MECA_TANG dans le fichier .mess).

```
/ 'ELASTIQUE'
```

On utilise la matrice élastique (option RIGI_MECA dans le fichier .mess).

/ 'EXTRAPOL'

On calcule l'estimation de l'incrément de déplacement à partir de l'incrément total obtenu comme solution au pas de temps précédent (pondéré par le rapport des pas de temps). On projette cette estimation sur l'ensemble des champs cinématiquement admissibles (i.e. satisfaisant les conditions aux limites de DIRICHLET) selon la norme donnée par la matrice élastique, qui doit donc être calculée. Cette fonctionnalité est intéressante dans le cas de l'utilisation de schémas d'intégration locale explicite de type RUNGE-KUTTA qui ne fournissent pas de matrice tangente : dans ce cas la méthode de NEWTON utilise une matrice élastique, mais le nombre d'itérations nécessaires peut être élevé. L'utilisation de l'extrapolation peut améliorer les performances.

/ 'DEPL_CALCULE'

Permet de proposer comme déplacement pour la prédiction à chaque pas de temps, le déplacement donné par une histoire mécanique précisée sous le mot clé `EVOL_NOLI` (§3.8.3).

Utilité :

- *supposons qu'on réalise un premier calcul avec un maillage grossier. On souhaite réaliser le même calcul mais sur un maillage plus fin. On peut supposer que la solution en déplacement pour ce second calcul n'est pas éloignée de celle du premier calcul et donc qu'une bonne prédiction du déplacement pour ce second calcul est la projection des déplacements du calcul 1 sur les nœuds du nouveau maillage (la projection des déplacements sur le nouveau maillage doit être réalisée préalablement avec l'opérateur `PROJ_CHAMP` [U4.72.05]). Ce mot clé permet de réaliser ce mode de prédiction.*
- *cela permet de réduire la place mémoire et de conserver ces résultats en vue d'une poursuite ultérieure. Pour un gros calcul, on peut stocker uniquement les déplacements à tous les instants aux formats IDEAS ou MED dans `IMPR_RESU`. Si on veut recalculer les contraintes et variables internes, on fait un `LIRE_RESU` au format adéquate puis on utilise `DEPL_CALCULE` avec `ITER_GLOB_MAXI` : 0 (on effectue une seule itération) et `ARRET` : NON (il n'y a pas convergence, on ne vérifie pas l'équilibre). Il est toutefois nécessaire pour des raisons de syntaxe de donner un chargement (éviter les chargements dirichlet qui imposent une résolution linéaire) ainsi qu'un critère de convergence, même si ces informations ne sont pas prises en compte.*

4.8.2 Opérande MATRICE

```
◇ MATRICE
/ 'TANGENTE'
◇ REAC_INCR = / 1 [DEFAULT]
               / mf
◇ REAC_ITER = / 0 [DEFAULT]
               / it
```

La matrice utilisée pour les itérations globales de la méthode est la matrice tangente [R5.03.01] qui est réévaluée tous les `mf` incréments de temps (`mf` positif ou nul) et toutes les `it` itérations de NEWTON pour un incrément de temps donné (précisément aux itérations de numéro `it`, `2it`, `3it`...). Donc à la première itération de NEWTON, on ne réassemble la matrice tangente que si `it` vaut 1 : sinon on garde la matrice utilisée dans la phase de prédiction. Par convention si `it` vaut 0 la matrice n'est pas réévaluée durant tout le pas de temps.

```
◇ PAS_MINI_ELAS = 0. [DEFAULT]
                  pasmini [I]
```

Permet de passer de la matrice tangente à la matrice élastique lorsque le pas de temps est ou devient (par le redécoupage) inférieur à `pasmini`.

Comme la convergence avec la matrice élastique est plus lente que celle avec la matrice tangente, le mot clé `ITER_GLOB_ELAS` sous le mot clé facteur `CONVERGENCE` permet de définir un nombre d'itérations maximal spécifique à l'utilisation de la matrice élastique et différent de celui associé à l'utilisation de la matrice tangente.

Utilité :

Cette option peut être utile lorsque le redécoupage automatique du pas de temps (cf. [§ 3.7.4]) ne suffit pas à faire converger un calcul. Par exemple, dans le cas de lois adoucissantes, la matrice tangente peut devenir singulière et il vaut donc mieux utiliser la matrice élastique pour converger.

```
/ 'ELASTIQUE'
```

La matrice utilisée correspond au calcul élastique : elle n'est évaluée qu'une fois à l'instant initial, en début d'algorithme.

Cette matrice "élastique" est calculée en utilisant le module d'YOUNG donné sous le mot clé `ELAS` de l'opérateur `DEFI_MATERIAU`, et non pas la pente à l'origine de la courbe de traction donnée sous le mot clé `TRACTION` (et qui sert, elle, dans l'expression de la relation de comportement).

4.8.3 Opérande EVOL_NOLI

```
◇ EVOL_NOLI = evol_noli
```

Nom du concept de type `evol_noli` qui servira dans la prédiction par `DEPL_CALCULE`.

4.9 Opérande INCREMENT

◇ INCREMENT

Par rapport à la syntaxe de ce mot clé décrite dans le document [U4.51.03], l'opérande est légèrement modifié.

Le mot clef `LIST_INST` est rendu facultatif (alors qu'il est obligatoire dans [U4.51.03]), ceci afin de permettre à l'utilisateur d'entrer une liste d'instant de calcul personnalisé mais aussi de ne pas le faire et de laisser la macro utiliser la liste par défaut suivante :

```
list_inst = DEFI_LIST_REEL(DEBUT=0.0,
                           INTERVALLE=( _F(JUSQU_A=1.0,
                                             NOMBRE=2, ),
                                           _F(JUSQU_A=11.0,
                                             NOMBRE=20, ),
                                           _F(JUSQU_A=600.0,
                                             NOMBRE=20, ),
                                           _F(JUSQU_A=610.0,
                                             NOMBRE=20, ),
                                           _F(JUSQU_A=1800.0,
                                             NOMBRE=20, ),
                                           _F(JUSQU_A=7200.0,
                                             NOMBRE=20, ), ), ), );
```

INCREMENT :

Définit les intervalles de temps pris dans la méthode incrémentale.

Les instants ainsi définis n'ont de sens physique que pour des relations de comportement où le temps intervient explicitement (visco-élastiques ou visco-plastiques par exemple). Dans les autres cas, ils permettent seulement d'indicer les incréments de charge et de paramétrer l'évolution d'un éventuel champ de température.

4.9.1 Opérandes LIST_INST / EVOLUTION

◇ LIST_INST = litps

Les instants de calcul sont ceux définis dans le concept `litps` par l'opérateur `DEFI_LIST_REEL` [U4.34.01].

◇ EVOLUTION = / 'CHRONOLOGIQUE' [DEFAULT]

Le mot clé 'CHRONOLOGIQUE' permet de vérifier si la liste d'instant donnée par l'utilisateur est strictement croissante (si non message d'erreur),

4.9.2 Opérandes NUME_INST_INIT / INST_INIT / NUME_INST_FIN / INST_FIN

◇ / NUME_INST_INIT = nuini / INST_INIT = instini

L'instant initial du calcul (qui donc n'est pas (re)calculé) est désigné soit par sa valeur (`INST_INIT`), soit par son numéro d'ordre dans la liste d'instant `litps` (`NUME_INST_INIT`). Pour pouvoir accéder par valeur, il est nécessaire que la liste soit ordonnée (`EVOLUTION : 'CHRONOLOGIQUE'` ou `'RETROGRADE'`).

En l'absence des mots clés `INST_INIT` ou `NUME_INST_INIT`, le défaut est calculé de la manière suivante :

- si un état initial est précisé (opérande `ETAT_INIT`) et s'il définit un instant correspondant (par `EVOL_NOLI` ou `INST_ETAT_INIT`) alors l'instant initial est celui défini par l'état initial,
- s'il n'y a pas d'état initial (opérande `ETAT_INIT`) ou qu'il ne définit pas d'instant correspondant (les champs sont donnés dans `ETAT_INIT` sans préciser `INST_ETAT_INIT`), alors on prend le premier instant de la liste d'instant `litps` (`NUME_INST_INIT : 0`), ou le dernier lorsque l'évolution est rétrograde.

Titre : Macro commande MACR_CABRI_CALC
 Auteur(s) : M. ABBAS

Date : 14/06/07
 Clé : U4.CB.20-B Page : 17/20

```

◇ / NUME_INST_FIN = nufin
  / INST_FIN      = instfin

```

L'instant final (dernier pas calculé) est désigné de la même manière que l'instant initial (soit NUME_INST_FIN, soit INST_FIN), sauf qu'il n'est pas possible de faire référence à l'instant de l'état initial.

A - Exemple simple (par défaut)

```

LIST = DEFI_LIST_REEL( DEBUT =0.,
                      INTERVALLE =_F(JUSQU'A = 10., NOMBRE =10)),

U = STAT_NON_LINE (INCREMENT =_F ( LIST_INST =LIST,
                                   INST_FIN =4.)) ,
U = STAT_NON_LINE( reuse=U,
                  INCREMENT =_F (LIST_INST =LIST),
                  ETAT_INIT =_F (EVOL_NOLI =U)) ,

```

1^{er} STAT_NON_LINE : effectue le calcul pour les instants 1, 2, 3 et 4s.

2nd STAT_NON_LINE : effectue le calcul pour les instants 5, 6, 7, 8, 9 et 10s, l'état initial correspondant au temps 4s. (par défaut INST_INIT=INST_ETAT_INIT=INST=4.).

B - Exemple pour montrer l'intérêt de INST_INIT

```

LIST = DEFI_LIST_REEL( DEBUT =0.,
                      INTERVALLE =_F (JUSQU'A = 10., NOMBRE =10)),

U = STAT_NON_LINE(INCREMENT =_F( LIST_INST =LIST,
                                   INST_FIN =4.)) ,
U = STAT_NON_LINE( reuse = U,
                  INCREMENT =_F ( LIST_INST =LIST,
                                   INST_INIT =8.),
                  ETAT_INIT =_F ( EVOL_NOLI =U)) ,

```

1^{er} STAT_NON_LINE : effectue le calcul des instants 1 à 4s.

2nd STAT_NON_LINE : effectue le calcul pour les instants 9 et 10s (ne fait rien pour t=5, 6, 7 et 8s), l'état initial correspondant au temps t=4s (par défaut INST=4.).

4.9.3 Opérande PRECISION

```

◇ PRECISION = prec Cf. [U4.71.00].

```

4.9.4 Opérande SUBD_METHODE

```

◆ SUBD_METHODE = / 'AUCUNE' [DEFAULT]
                  / 'UNIFORME'
                  / 'EXTRAPOLE'

```

Ce mot-clef indique la méthode de subdivision des pas de temps qui s'active lorsque le problème non-linéaire ne converge pas suivant les autres critères de INCREMENT. Par défaut, aucune méthode n'est activée ('AUCUNE'). Si on donne SUBD_METHODE = 'UNIFORME', en cas de non-convergence, l'intervalle de temps est découpé en SUBD_PAS en appliquant un ratio de SUBD_COEF_PAS_1 sur la première division.

Si on active la subdivision de type 'EXTRAPOLE', le code tente de déterminer automatiquement le nombre de subdivisions et le ratio à appliquer pour converger avec le nombre d'itérations demandé.

4.9.5 Opérande SUBD_PAS / SUBD_PAS_MINI / SUBD_COEF_PAS_1

```

◇ SUBD_PAS = subpas
◇ SUBD_PAS_MINI = submini
◇ SUBD_COEF_PAS_1 = coefsub

```

Permet de réaliser un redécoupage automatique du pas de temps lorsque l'algorithme de Newton ne converge pas.

Le pas de temps est redécoupé en subpas sous pas. Par défaut il n'y a pas de redécoupage (subd_pas : 1). La subdivision automatique s'arrête lorsque les nouveaux pas créés sont plus petits que SUBD_PAS_MINI. Les nouveaux pas créés sont de taille identique, excepté le premier qui est égal à cette taille multipliée par COEF_SUBD_PAS_1 (par défaut 1). Ceci permet de mieux prendre en compte les problèmes de décharge de la structure (changement de matrice tangente) sans utiliser la matrice élastique (PREDICTION : 'ELASTIQUE' ou MATRICE : 'ELASTIQUE' sous l'opérande NEWTON).

Remarque concernant le mot clé DECOUPE sous SOLVEUR :

Lors de calcul de flambage élastoplastique, il peut arriver que la matrice tangente du système soit singulière au cours des itérations de Newton. En redécoupant le pas de temps, on peut passer ces points durs. Sous l'opérande SOLVEUR, le mot clé DECOUPE sous STOP_SINGULIER sert à gérer ces points durs. Il est alors nécessaire de renseigner les mots clés relatifs au redécoupage pour que la méthode DECOUPE soit activée.

4.9.6 Opérande OPTI_LIST_INST / NOM_CHAM / NOM_CMP / VALEUR

◇ OPTI_LIST_INST	=	'INCR_MAXI'	[DEFAULT]
◇ NOM_CHAM	=	'TEMP'	[DEFAULT]
◇ NOM_CMP	=	'TEMP'	[DEFAULT]
◇ VALE	=	valeur	

Ces opérandes n'ont d'intérêt que lorsqu'on réalise un calcul thermomécanique. Permet de créer si besoin est une nouvelle liste de pas de temps mécanique de sorte qu'entre chaque incrément de temps, l'incrément de température soit inférieur à une valeur donnée par l'utilisateur et renseignée par le mot clé VALE.

La création de cette nouvelle liste se fait de la façon suivante :

- Liste d'instants initial en mécanique : T_i
- Liste d'instants thermique : θ
- Nouvelle liste d'instants mécanique final à créer si besoin : T_f
- On insère entre chaque intervalle de la liste initiale mécanique T_i , les instants thermiques inclus dans cet intervalle. On récupère alors pour chaque intervalle une liste d'instants τ = $[\tau_0, \tau_1, \tau_2, \dots, \tau_N]$
- Construction de la liste final T_f

a) Initialisation : $\tau_f = \tau_0$

b) 1^{er} Test :

Si $T(\tau_j) - T(\tau_f) > \text{valeur}$ avec $T(t)$ la température au temps t et τ_f le dernier instant inséré dans la nouvelle liste T_f , alors on garde dans la nouvelle liste T_f , l'instant τ_{j-1}

c) 2^{ème} Test :

Si $T(\tau_j) - T(\tau_{j-1}) > \text{valeur}$ alors on redécoupe uniformément cet intervalle de façon à satisfaire la condition sur l'incrément de température.

Exemple : Si $T(\tau) = [T(\tau_1) = 20^\circ\text{C}, T(\tau_2) = 30^\circ\text{C}, T(\tau_3) = 55^\circ\text{C}, T(\tau_4) = 65^\circ\text{C}]$ avec VALE = 15°C

Initialisation : $\tau_f = \tau_1$

Intervalle 1 :

1^{er} Test = 2^{ème} Test : $T(\tau_2) - T(\tau_1) = 10^\circ\text{C} < 15$ donc on a $T_f = [\tau_1]$

Intervalle 2 :

1^{er} Test : $T(\tau_3) - T(\tau_f) = 35^\circ\text{C} > 15$ donc on a $T_f = [\tau_1, \tau_2]$ et $\tau_f = \tau_2$

2^{ème} Test : $T(\tau_3) - T(\tau_2) = 25^\circ\text{C} > 15$ donc on a $T_f = [\tau_1, \tau_2, \tau_3]$ tel que $T(\tau_3) = 42.5^\circ\text{C}$, τ_3 et $\tau_f = \tau_3$

Intervalle 3 :

1^{er} Test = 2^{ème} Test : $T(\tau_4) - T(\tau_3) = 10^\circ\text{C} < 15^\circ\text{C}$

d'où la liste finale suivante :

$$T_f = [\tau_1, \tau_2, T_3 \text{ tel que } T(T_3) = 42.5^\circ\text{C}, \tau_3, \tau_4]$$

5 Opérandes de sortie

5.1 Mot clé **MODELE_THER**

◇ `MODELE_THER` = `CO modele` [TXM]

Ce mot clé permet de nommer éventuellement le modèle thermique afin de le réutiliser par exemple pour faire un autre calcul (n'utilisant pas `MACR_CABRI_CALC`) ou du post-traitement.

5.2 Mot clé facteur **CHAR_THER**

◇ `CHAR_THER`

Mot clé facteur permettant de récupérer les concepts créés pour le chargement thermique.

5.2.1 Opérande **CHARGE**

◆ `CHARGE` = `CO char_ther` [TXM]

Cet opérande permet de nommer le chargement thermique afin de le réutiliser par exemple pour faire un autre calcul (n'utilisant pas `MACR_CABRI_CALC`) ou du post-traitement. La nature de ce chargement thermique est défini par l'opérande `TYPE`.

5.2.2 Opérande **TYPE**

◆ `TYPE` =
 `/BRIDE_FLUIDE`
 `/BRIDE_AIR`
 `/ECROU_GOUJON`
 `/BRIDE_JOINT` [TXM]

Description du chargement thermique contenu dans le concept nommé par `CHAR` :

- `BRIDE_FLUIDE` : Échanges thermiques internes entre le fluide et la bride.
- `BRIDE_AIR` : Échanges thermiques externes entre bride et air ambiant.
- `ECROU_GOUJON` : Échanges thermiques entre écrou et goujon.
- `BRIDE_JOINT` : Échanges thermiques entre bride et joint.

5.3 Mot clé **RESU_THER**

◇ `RESU_THER` = `CO evol_ther` [TXM]

Ce mot clé permet de nommer éventuellement les résultats du calcul thermique afin de le réutiliser en dehors de `MACR_CABRI_CALC`.

5.4 Mot clé **MODELE_MECA**

◇ `MODELE_MECA` = `CO modele` [TXM]

Ce mot clé permet de nommer éventuellement le modèle mécanique afin de le réutiliser plus tard.

5.5 Mot clé facteur **CHAR_MECA**

◇ `CHAR_MECA`

Mot clé facteur permettant de récupérer les concepts créés pour le chargement mécanique.

5.5.1 Opérande CHARGE

♦ CHARGE = CO char_meca [TXM]

Cet opérande permet de nommer le chargement mécanique afin de le réutiliser par exemple pour faire un autre calcul (n'utilisant pas MACR_CABRI_CALC) ou du post-traitement. La nature de ce chargement mécanique est défini par l'opérande TYPE.

5.5.2 Opérande TYPE

♦ TYPE =
/BLOC_BAS_GOUJ [TXM]
/BLOC_BAS_JOINT
/BLOC_LAT_ALES
/BLOC_LAT_NALES
/PLAN_TUBE
/PRES_FLU
/EFFET_FOND
/CONT_JOINT
/DEFO_THER
/SERR_ECROU_1
/SERR_ECROU_2

Description du chargement mécanique contenu dans le concept nommé par CHAR :

- BLOC_BAS_GOUJ : Blocage bas du goujon.
- BLOC_BAS_JOINT : Blocage bas du joint.
- BLOC_LAT_ALES : Blocage latéral, face latérale avec alésage.
- BLOC_LAT_NALES : Face latérale sans alésage.
- PLAN_TUBE : Condition de planéité de la face de coupe du tube.
- PRES_FLU : Pression due au fluide.
- EFFET_FOND : Effet de fond.
- CONT_JOINT : Contact zone de joint.
- DEFO_THER : Déformation thermique.
- SERR_ECROU_1 et SERR_ECROU_2 : Serrage écrou/goujon (pre-tensionnement).

5.6 Opérande CHAM_MATER

♦ CHAM_MATER = CO cham_mater [TXM]

Nom du concept de type cham_mater retournant le champ de matériau affecté dans la macro-commande.