Opérateur MODI_REPERE

1 But

Cet opérateur permet de modifier le repère dans lequel s'expriment des champs portés par des coques comme des éléments de milieu continu.

L'appel à MODI_REPERE produit généralement un nouveau concept resultat :

```
resuout = MODI_REPERE  (  RESULTAT = resuin … )
```

Le concept resultat en sortie est de même type que le concept en entrée. Pour les changements de repère COQUE_INTR_UTIL et COQUE_UTIL_INTR, il est possible (quoique déconseillé) d'utiliser le même concept resultat en entrée et en sortie. Dans tous les autres cas, ce fonctionnement est interdit.
Un concept produit par MODI_REPERE ne doit plus être utilisé ensuite pour faire des calculs.
Il est de plus nécessaire de prendre soin de bien respecter les règles du paragraphe [§4].

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2 Syntaxe

```
resuout [*] = MODI_REPERE (
              ◊ reuse = resuout,

              /  ♦  RESULTAT = resuin,          /      [evol_elas]
                                                /      [evol_noli]
                                                /      [evol_ther]
                                                /      [mode_flamb]
                                                /      [dyna_trans]
                                                /      [dyna_harmo]
                                                /      [mode_meca]
                                                /      [mult_elas]
                                                /      [base_modale]
                ♦ CHAM_GD  = CHIN,                        [champ]

              # si RESULTAT
                ◊ #  Sélection des numéros d'ordre :
                     /   TOUT_ORDRE =     'OUI'  ,            [DEFAUT]
                     /   NUME_ORDRE =     l_nuor ,            [l_I]
                     /   LIST_ORDRE =     l_nuor ,            [listis]
                     /   NUME_MODE  =     l_numo ,            [l_I]
                     /   NOEUD_CMP  =     l_nomo ,            [l_K16]
                     /   NOM_CAS    =     nocas  ,            [K16]
                     /    ♦   / INST =     l_inst ,            [l_R]
                              / FREQ =     l_freq ,            [l_R]
                              / LIST_INST =  l_inst ,          [listr8]
                              / LIST_FREQ =  l_freq ,          [listr8]
                 ◊   /    ♦ CRITERE    =   /  'RELATIF',          [DEFAUT]
                     ♦ PRECISION      =   /  prec,            [R]
                                      /  1.0D-6,          [DEFAUT]
                   / ♦ CRITERE     =   'ABSOLU',
                     ♦ PRECISION   =   prec,                  [R]
                ♦    MODI_CHAM  =(_F(
                ♦    NOM_CHAM   =  nomch,                     [K16]
                ♦    NOM_CMP=   l_cmp,                    [l_K8]
                ♦    TYPE_CHAM  =  /   'VECT_2D',
                                   /   'VECT_3D',
                                   /   'TENS_2D',
                                   /   'TENS_3D',
                                   /   'COQUE_GENE' ,) , )
                ♦   REPERE=  /   'UTILISATEUR',
                             /   'CYLINDRIQUE',
                             /   'COQUE',
                             /   'COQUE_INTR_UTIL',
                             /   'COQUE_UTIL_INTR',
                             /   'COQUE_UTIL_CYL',
                ♦   AFFE=_F(
                           si REPERE='UTILISATEUR' :
                                  /   ♦   ANGL_NAUT = (alpha,beta,gamma)   [l_R]
                                  /   ♦   VECT_X = (vx1,vx2,vx3)           [l_R]
                                      ♦   VECT_Y = (vy1,vy2,vy3)           [l_R]
                           si  REPERE='CYLINDRIQUE' ou 'COQUE_UTIL_CYL' :
                                      ♦   ORIGINE = (x,y,z)                [l_R]
                                      ♦   AXE_Z=(oz1 oz2 oz3)              [l_R]
                           si  REPERE='COQUE' :
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                              /  ♦  ANGL_REP= ( α,β )                    [l_R]
                              /  ♦  VECTEUR=(x, y, z)                    [l_R]
                    ♦  /TOUT     =  'OUI',                             [TXM]
                       /GROUP_MA =  l_grma,                    [l_gr_maille]
                       /GROUP_NO =  l_grno,                    [l_gr_noeud]
                  )

              # si CHAM_GD
                ♦ REPERE    = 'GLOBAL_UTIL'                            [DEFAUT]
                 ♦  CARE_ELEM = carelem
[cara_elem]

              ◊  INFO   = / 1 ,                                        [DEFAUT]
                          / 2 ,
              ◊  TITRE =   titre,                                      [l_Kn]

 )
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3 Opérandes

3.1 Opérandes RESULTAT et CHAM_GD

/ ♦ RESULTAT = resuin
Nom de la structure de données résultat. Cet argument doit être différent de celui utilisé pour le concept produit par l'opérateur. L'option reuse est autorisée (quoique déconseillée) uniquement lorsque REPERE='COQUE_UTIL_INTR' ou REPERE='COQUE_INTR_UTIL'.

♦ CHAM_GD = chin
Nom du champ en entrée. Ce champ est très particulier et il ne peut être produit que par la commande PROJ_CHAMP avec la méthode sous-points appliquée à un champ de contrainte 3D. La commande PROJ_CHAMP projette ce champ sur tous les points de la famille RIGI des éléments finis supportant des sous-points. Voir les opérandes REPERE et CARA_ELEM de cette commande, ainsi que la documentation de la commande PROJ_CHAMP.

3.2 Sélection des numéros d'ordre et des instants

3.2.1 Opérandes TOUT_ORDRE / NUME_ORDRE / LIST_ORDRE /

♦ / TOUT_ORDRE = 'OUI' (valeur par défaut)
Ce mot clé indique que l'on applique le changement de repère pour tous les numéros d'ordre du concept résultat
Exemple : tous les instants pour un résultat de type evol_*.

/ NUME_ORDRE = l_nume
Le changement de repère se fera pour les valeurs de numéro d'ordre l_nume fournies.

/ TOUT_INST = 'OUI'
Ce mot clé indique que l'on veut changer le repère pour tous les instants.

/ LIST_ORDRE = l_ord
Ce mot clé indique que l'on veut modifier le repère aux numéros d'ordre décrits dans le concept l_ord de type listis.

/ NUME_MODE = l_numo
Ce mot clé indique que l'on veut modifier le repère des modes désignés par leurs numéros de mode dans la liste l_numo.

/ NOEUD_CMP = l_nomo
Ce mot clé indique que l'on veut modifier le repère des modes statiques désignés par leurs DDL dans la liste l_nomo.

/ NOM_CAS = nocas
Ce mot clé indique que l'on veut modifier le repère d'un résultat statique désigné par le nom de son cas de charge nocas.

3.2.2 Opérandes INST / LIST_INST / FREQ / LIST_FREQ

◊ / INST = l_inst
Ce mot clé indique que l'on veut modifier le repère aux instants l_inst.

/ LIST_NST = li_inst
Ce mot clé indique que l'on veut modifier le repère aux instants décrits dans le concept li_inst de type listr8.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

/ FREQ = l_freq
Ce mot clé indique que l'on veut modifier le repère aux fréquences l_freq.

/ LIST_FREQ = li_freq
Ce mot clé indique que l'on veut modifier le repère aux fréquences décrites dans le concept li_freq de type listr8.

3.3 Mot-clé facteur MODI_CHAM

Ce mot-clé facteur permet de définir les champs et les composantes à calculer. Il peut de plus être répété plusieurs fois. On peut traiter plusieurs champs à la fois.

3.3.1 Opérande NOM_CHAM

Nom symbolique du champ à traiter.

3.3.2 Opérande NOM_CMP

Noms des composantes que l'on veut traiter (voir [U2.01.04]). Voir également le paragraphe [§4].

3.3.3 Opérande TYPE_CHAM

Cet opérande **obligatoire** permet de spécifier le type de champ à traiter. Les différents types sont les suivants :
/ 'VECT_2D' signifie que l'on traite un champ de vecteurs en dimension 2,
/ 'VECT_3D' signifie que l'on traite un champ de vecteurs en dimension 3,
/ 'TENS_2D' signifie que l'on traite un champ de tenseurs symétriques en dimension 2,
/ 'TENS_3D' signifie que l'on traite un champ de tenseurs symétriques en dimension 3,
/ 'COQUE_GENE' signifie que l'on traite un champ de quantités généralisées en dimension 3 (déformations ou efforts). Ce type n'est valable que pour le repère 'COQUE'.

Voir également le paragraphe §4.1.

3.4 Mot-clé simple REPERE

Ce mot-clé permet de sélectionner un type de repère parmi ceux listés ci-dessous. Pour définir le repère choisi, il faut utiliser le mot-clé facteur AFFE.
- repère 'UTILISATEUR' : pour les éléments de milieu continu.
- repère 'CYLINDRIQUE' : pour les éléments de milieu continu.
- repère 'COQUE' : définit un repère utilisateur sur les éléments coque.
- repère 'COQUE_UTIL_INTR' : permet de passer du repère utilisateur au repère intrinsèque sur les éléments coque.
- repère 'COQUE_INTR_UTIL' : permet de passer du repère intrinsèque au repère utilisateur sur les éléments coque.
- Repère 'COQUE_UTIL_CYL' : permet de passer du repère utilisateur sur les éléments coque à un repère global cylindrique.

Remarque :
- *Dans le cas où l'utilisateur a spécifié que le concept est ré-entrant (par le mot réservé reuse), les seuls choix possibles pour le mot-clé simple REPERE sont 'COQUE_UTIL_INTR' ou 'COQUE_INTR_UTIL'. Les champs où la transformation est pertinente sont transformés, tandis que les autres champs sont gardés en l'état ;*
- *Pour une coque, le repère étant défini par élément, le changement de repère 'COQUE' ne s'applique qu'aux champs par élément (cham_elem).*

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Le repère intrinsèque est un repère propre à un élément fini de plaque ou de coque, où sont effectués les calculs élémentaires et qui sert d'intermédiaire au repère utilisateur dans lequel sont exprimés les champs de contraintes et de déformations. Il est défini de la façon suivante :

- Pour un triangle (*cf.* Figure 3.4-1) dont les sommets sont numérotés de s_1 à s_3 : il s'agit du repère formé par $x = s_1 s_2$, $y = z \wedge x$, $z = s_1 s_2 \wedge s_1 s_3$
- Pour un quadrangle (*cf.* Figure 3.4-2) dont les sommets sont numérotés de s_1 à s_4 et dont les milieux des côtés sont numérotés de m_1 (milieu de $s_1 s_2$) à m_4 (milieu de $s_4 s_1$) : il s'agit du repère formé par $x = m_4 m_2$, $y = z \wedge x$, $z = m_4 m_2 \wedge m_1 m_3$

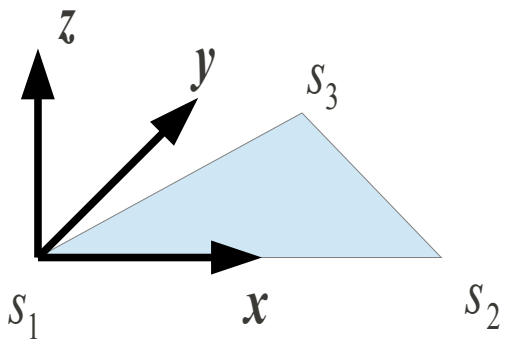


Figure 3.4-1: Repère intrinsèque pour une maille de type TRIA3

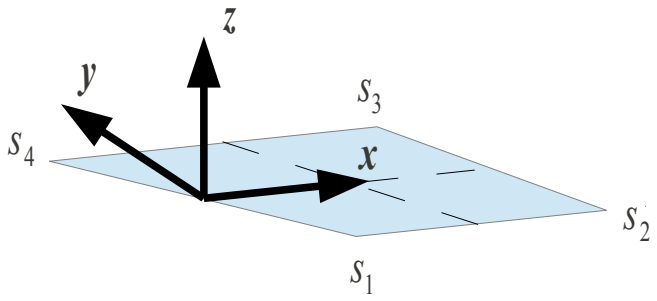


Figure 3.4-2: Repère intrinsèque pour une maille de type QUAD4

3.5 Mot-clé facteur AFFE

Ce mot-clé facteur définit le repère précédemment choisi :

- repère 'UTILISATEUR' :
  - soit défini par la donnée de 3 angles nautiques (en degrés) ,
    ANGL_NAUT = (alpha,beta,gamma)
  - soit en 3D par la donnée de 2 vecteurs de base avec VECT_X et VECT_Y à partir desquels 3 angles nautiques sont construits tels que :

pour alpha
$$\begin{cases} si\ VECT_X(1) = 0\ alors\ alpha = 0 \\ sinon\ alpha = \arctan\left(VECT_X(2) / VECT_X(1)\right) \end{cases}$$

pour beta
$$\begin{cases} si\ \sqrt{VECT_X(1)^2 + VECT_X(2)^2} = 0\ alors\ beta = 0 \\ sinon\ beta = -\arctan\left(VECT_X(3) / \sqrt{VECT_X(1)^2 + VECT_X(2)^2}\right) \end{cases}$$

pour gamma
$$\begin{cases} si\ VECT_Y(2) = 0\ alors\ gamma = 0 \\ sinon\ gamma = \arctan\left(VECT_Y(3) / VECT_Y(2)\right) \end{cases}$$

- repère 'CYLINDRIQUE' ou 'COQUE_UTIL_CYL' : défini par la donnée de l'origine du repère et l'axe Oz

  ORIGINE = (x, y, z) coordonnées de l'origine O du repère
  AXE_Z = (oz1 , oz2 , oz3) coordonnées d'un vecteur définissant l'axe oz (axe du cylindre).

  Ce repère ne peut être défini qu'une seule fois.

- repère ' COQUE' : il est défini soit par la donnée de deux angles en degrés (mot-clé ANGL_ REP) ou bien à l'aide du mot-clé VECTEUR. *Cf.* mot-clé COQUE dans AFFE_CARA_ELEM [U4.42.01].

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Dans le cas ANGL_REP, les angles servent à définir le vecteur précédent qui, projeté sur le plan de la coque, donnera l'axe Ox du nouveau repère. Connaissant la normale en tout point à la coque, on en déduit aisément le nouveau repère.

Remarques :

- *On peut définir un repère variable en utilisant plusieurs occurrences du mot-clé AFFE et en donnant par exemple des angles nautiques différents sur différents groupes de nœuds ou de mailles. Cette possibilité est réservée aux changements de repères portant sur des éléments de milieux continus ('UTILISATEUR' et 'CYLINDRIQUE') ;*
- *Pour les changements de repère sur des éléments de structure ('COQUE' , 'COQUE_UTIL_INTR' , 'COQUE_INTR_UTIL' , 'COQUE_UTIL_CYL'), le repère ne peut pas être variable, et une seule occurrence du mot-clé AFFE est autorisée.*

3.5.1 Opérandes TOUT, GROUP_MA, GROUP_NO

Permettent d'appliquer sur tout le maillage ou de restreindre le changement de repère à certaines mailles ou certains nœuds, ou bien de définir le repère de façon différente sur certaines mailles ou certains nœuds.
Attention, pour les champs par élément, GROUP_NO est interdit.

3.6 Opérande REPERE

Cet opérande est obligatoire dès que CHAM_GD est renseigné. Il permet de changer le repère du champ en entrée. Actuellement une seule possibilité existe : passage du repère global au repère local de l'élément de coque REPERE='GLOBAL_UTIL'. On suppose donc que le champ donné sous CHAM_GD est défini dans le repère **global**.

3.7 Opérande CARA_ELEM

Cet opérande est obligatoire dès que CHAM_GD est renseigné. Il permet de donner le repère local des éléments de structure et donc de faire le passage du repère global de définition du champ de départ dans le repère local des éléments.

3.8 Opérande INFO

Cet opérande permet d'afficher la structure du concept résultat (resuout).

3.9 Opérande TITRE

Voir [U4.03.01].

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4 Notes d'utilisation

4.1 Définitions et précautions d'utilisation

Un concept produit par MODI_REPERE ne doit plus être utilisé ensuite pour faire des calculs, Code_Aster ne mémorisant pas le repère dans lequel figurent les champs. Seuls des impressions ou des tracés sont licites.

Selon chaque type de champ, il est nécessaire de bien spécifier après NOM_CMP le nombre exact de composantes et dans l'ordre suivant :

- dans le cas des vecteurs en dimension 2 ou 3 (2 ou 3 composantes) : X , Y , (Z) ,
- dans le cas d'un tenseur en dimension 2 (4 composantes) : XX , YY , ZZ , XY ,
- dans le cas d'un tenseur en dimension 3 (6 composantes) : XX , YY , ZZ , XY , XZ , YZ
- dans le cas d'une quantité généralisée en dimension 3 (8 composantes : deux tenseurs en dimension 2, un vecteur en dimension 2) : XX , YY , XY , XX , YY , XY , X , Y

Remarques :

- *Lorsqu'un nœud N du maillage se trouve sur l'axe Oz (du repère cylindrique), on cherche le nœud moyen des centres géométriques des mailles contenant le nœud N pour le calcul de la matrice de passage en repère cylindrique. Si ce nœud moyen se trouve également sur l'axe Oz , le calcul s'arrête en erreur fatale.*
- *Lorsque toutes les composantes d'un nœud ne sont pas présentes dans le champ à traiter, on n'écrit rien dans le champ transformé pour ce nœud. Ceci se rencontre par exemple avec les éléments de la modélisation COQUE_3D pour lesquels les nœuds situés au milieu des faces n'ont pas de degré de liberté de translation. Le champ de déplacement issu de MODI_REPERE n'est donc pas calculé pour ces nœuds milieux de faces.*
- *Cette liste de composantes est exhaustive. Il n'est pas licite d'effectuer un changement de repère sur des champs possédant d'autres composantes. Par exemple, un champ de déformation thermique (de type EPVC_NOEUD) est certes un tenseur de dimension 3 mais ses composantes ne sont pas XX , YY , ZZ , XY , XZ , YZ , mais $EPTHER_L$, $EPTHER_T$, $EPTHER_N$, $EPSECH$, $EPHYDR$, $EPTOT$. Ces composantes ne correspondant pas à un repère, cela n'aurait pas de sens de les changer de repère.*

4.2 Correspondances cylindriques

Pour l'expression des contraintes en repère cylindrique on fait les correspondances suivantes (par simplification on note $T=\theta$) :

Contraintes en repère cartésien		Contraintes en repère cylindrique
vecteur	X	R
	Y	Z
	Z	T
tenseur	XX	RR
	YY	ZZ
	ZZ	TT
	XY	RZ
	XZ	RT
	YZ	ZT

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

5 Exemples

5.1 Calcul des contraintes en repère cylindrique

```
RESU1=MODI_REPERE(
                  RESULTAT     = RESU,
                  NUME_ORDRE   = 1,
                  MODI_CHAM    =  (
                      _F(NOM_CHAM     = 'SIGM_ELNO',
                         NOM_CMP      =('SIXX','SIYY','SIZZ','SIXY',),
                         TYPE_CHAM    = 'TENS_2D',),),
                  REPERE       = 'CYLINDRIQUE',
                  AFFE = _F(ORIGINE   = (0.0, 0.0, 0.0,),
                            AXE_Z     = (0.0, 0.0, 1.0,), ),
      )
```

5.2 Calcul des contraintes et efforts généralisés sur des coques dans un repère variable défini par l'utilisateur

```
RESU2=MODI_REPERE(
                  RESULTAT     = RESU,
                  NUME_ORDRE   = 1,
                  MODI_CHAM    = (
                      _F(NOM_CHAM  = 'SIGM_ELNO',
                         NOM_CMP   = ('SIXX','SIYY','SIZZ',
                                      'SIXY','SIXY','SIXY',),
                         TYPE_CHAM = 'TENS_3D',),
                      _F(NOM_CHAM  = 'EFGE_ELNO',
                         NOM_CMP   = ('NXX','NYY','NXY',
                                      'MXX','MYY','MXY','QX','QY',),
                         TYPE_CHAM = 'COQUE_GENE',),
                  ),
                  REPERE = 'COQUE',
                  AFFE = (
                      _F(ANGL_REP = (30.0, 30.0,),GROUP_MA='GRMA1',),
                      _F(ANGL_REP = (45.0, 45.0,),GROUP_MA='GRMA2',),
                  ),
      )
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)