

Opérateur MODI_REPERE

1 But

Cet opérateur permet de faire des post-traitements en repère cylindrique ou dans un repère défini par l'utilisateur.

L'appel à MODI_REPERE se fait nécessairement à partir d'un nouvel argument :

```
resuou = MODI_REPERE ( RESULTAT = resuin ... )
```

Le concept `resultat` est de même type que le concept `donnee`.

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].

2 Syntaxe

```

resuout [*] = MODI_REPERE (
    ♦ RESULTAT =      resuin ,
                                /      [evol_elas]
                                /      [evol_noli]
                                /      [evol_ther]
                                /      [mode_flamb]
                                /      [dyna_trans]
                                /      [dyna_harmo]
                                /      [mode_meca]
                                /      [mult_elas]
                                /      [base_modale]

    ♦ # Sélection des numéro d'ordre :
      / TOUT_ORDRE =      'OUI' ,
      / NUME_ORDRE =      l_nuor ,
      / LIST_ORDRE =      l_nuor ,
      / NUME_MODE =      l_numo ,
      / NOEUD_CMP =      l_nomo ,
      / NOM_CAS =      nocas ,
      / ♦ / INST =      l_inst ,
          / FREQ =      l_freq ,
          / LIST_INST =      l_inst ,
          / LIST_FREQ =      l_freq ,
    ♦ / | CRITERE =      / 'RELATIF',
      | PRECISION =      / prec,
          / 1.0D-6,
      / ♦ CRITERE =      'ABSOLU',
      ♦ PRECISION =      prec,
    ♦ MODI_CHAM =(_F(
      ♦ GROUP_MA =      l_grma,
      ♦ GROUP_NO =      l_grno,
      ♦ MAILLE =      l_maille,
      ♦ NOEUD =      l_noeud,
      ♦ NOM_CHAM =      nomch,
      ♦ NOM_CMP =      l_cmp,
      ♦ TYPE_CHAM =      / 'VECT_2D' ,
          / 'VECT_3D' ,
          / 'TORS_3D' ,
          / 'TENS_2D' ,
          / 'TENS_3D' ,
      ) , )
    ♦ DEFI_REPERE=_F(
      REPERE=
          / 'UTILISATEUR' ,
              ♦ ANGL_NAUT = (alpha,beta,gamma) [l_R]
          / 'CYLINDRIQUE' ,
              ♦ ORIGINE = (x,y,z) [l_R]
              ♦ AXE_Z=(oz1 oz2 oz3) [l_R]
          )
    ♦ INFO = / 1 ,
      / 2 ,
    ♦ TITRE =      titre,

)

```

3 Opérandes

3.1 Opérande RESULTAT

♦ `RESULTAT = resuin`

Nom de la structure de données résultat à calculer en repère cylindrique. Cet argument doit impérativement être différent de celui utilisé pour le concept produit par l'opérateur.

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_num`

Le changement de repère se fera pour les valeurs de numéro d'ordre `l_num` 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_nomo`

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

`/ 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 extraire aux instants `l_inst`.

`/ LIST_INST = li_inst`

Ce mot clé indique que l'on veut extraire aux instants décrits dans le concept `li_inst` de type `listr8`.

`/ FREQ = l_freq`

Ce mot clé indique que l'on veut extraire aux fréquences `l_freq`.

`/ LIST_FREQ = li_freq`

Ce mot clé indique que l'on veut extraire 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.

3.3.1 Opérandes GROUP MA, GROUP NO, NOEUD, MAILLE

Permettent de restreindre le changement de repère à certaines mailles ou certains nœuds. Attention, pour les champs par éléments, NOEUD et GROUP NO sont interdits.

3.3.2 Opérande NOM CHAM

Nom symbolique du champ à traiter (voir fascicule [U5.01]).

3.3.3 Opérande NOM CMP

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

3.3.4 Opérande TYPE_CHAM

Cet opérateur **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 à 2 composantes,
- / 'VECT_3D' signifie que l'on traite un champ de vecteurs à 3 composantes,
- / 'TORS_3D' signifie que l'on traite un champ de torseurs à 6 composantes,
- / 'TENS_2D' signifie que l'on traite un champ de tenseurs symétriques d'ordre 2,
- / 'TENS_3D' signifie que l'on traite un champ de tenseurs symétriques d'ordre 3.

Voir également le paragraphe [§4].

3.4 Mot-clé facteur DEFI REPERE

Ce mot-clé facteur permet le choix d'un repère parmi :

- repère 'UTILISATEUR' : défini par la donnée de 3 angles nautiques (en degrés) :

$$\text{ANGL NAUT} = (\alpha, \beta, \gamma)$$

- repère 'CYLINDRIQUE' : 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 mot-clé ne peut quant à lui être utilisé qu'une seule fois.

3.5 Opérande INFO

Cet op rande permet d'afficher la structure du concept r sultat (`resuout`).

3.6 Opérande TITRE

Voir [U4.03.01].

4 Notes d'utilisation

4.1 Définitions et précautions d'utilisations

Un concept produit par MODI_REPERE ne doit plus être utilisé ensuite pour faire des calculs, 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 à 2 (ou 3) composantes) : X, Y, (Z),
- dans le cas d'un torseur à 6 composantes : 3 translations X, Y, Z, et 3 rotations RX, RY, RZ,
- dans le cas d'un tenseur d'ordre 2 (4 composantes) : XX, YY, ZZ, XY,
- dans le cas d'un tenseur d'ordre 3 (6 composantes) : XX, YY, ZZ, XY, XZ, YZ.

Remarques :

- *Lorsqu'un noeud N du maillage se trouve sur l'axe Oz (du repère cylindrique), on cherche le noeud moyen des centres géométriques des mailles contenant le noeud N pour le calcul de la matrice de passage en repère cylindrique. Si ce noeud 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.*

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

5 Exemples

5.1 Calcul des déplacements en repère cylindrique

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