Operator NORM_MODE

1 Drank

the role of the command is to normalize eigen modes according to a criterion chosen by the user.

The operators of modal computation MODE_ITER_INV [U4.52.04] and MODE_ITER_SIMULT [U4.52.03] produce a concept of the type mode_meca or mode_meca_c of which the real eigen modes or complex is standardized in such way that largest of the components which is not a LAGRANGE multiplier , that is to say equal to 1.

Operator NORM_MODE allows the user to choose another method of standardization, for example generalized mass, stiffness generalized…

According to standardization chosen, the modal parameters (participation factor, effective mass,…) are reactualized.

D-entering operator.

2 Syntax

```
m_out=  NORM_MODE    (

       ◊reuse=m_out            ,
       ♦MODE=m_in              ,                          /   [mode_meca]
                                                        /   [mode_meca_c]
                                                        /   [mode_flamb]

       ◊/NORME=/                   "MASS_GENE",
                          /   "RIGI_GENE",
                          /   "TRAN",
                          /   "TRAN_ROTA",
                          /   "EUCL",
                          /   "EUCL_TRAN",
          /◊NOEUD=no                    ,                       [node]
             ◊GROUP_NO=grno          ,                       [group_no]
              # So NOEUD or GROUP_NO
             ♦NOM_CMP=cmp         ,                        [kN]
          /SANS_CMP     =s_cmp    ,                           [l_Kn]
          /AVEC_CMP     =a_cmp    ,                           [l_Kn]

       ◊MODE_SIGNE=_F              (
             ♦/  NOEUD=no      ,                           [node]
                /GROUP_NO=grno   ,                         [group_no]
             ♦NOM_CMP=cmp         ,                        [kN]
             ◊SIGNE=/             "POSITIF",                  [DEFAULT]
                          /   "NEGATIF",
                          )

       ◊MASSE=masse            ,                               [matr_asse_depl_r]
                                                        or   [matr_asse_gene_r]
                                                        or   [matr_asse_pres_r]
       ◊RAIDE=masse            ,                               [matr_asse_depl_r]
                                                        or   [matr_asse_depl_c]
                                                        or   [matr_asse_gene_r]
                                                        or   [matr_asse_pres_r]
       ◊AMOR=masse             ,                               [matr_asse_depl_r]
                                                        or   [matr_asse_gene_r]

       ◊TITER=t             ,                                 [l_Kn]

       ◊   INFO=/1             ,                              [DEFAULT]
                      /2  ,

                              );

       if m_in is of type [ mode_meca ]
       then m_out is of type [ mode_meca ]
       idem with [ mode_meca_c ]
       idem with [ mode_flamb ]
```

3 Operands

3.1 Operand MODE

♦MODE =m_in

Name of the concept of the mode_* type which one wants to change the standardization of the eigen modes. If m_out is identical to m_in and if the key word "reuse" is activated with the value m_out, the renormalization is done out of core.

3.2 Operand NORMALIZES

◊ /NORME =

Symbolic name of the selected norm.

"MASS_GENE":

The modes are standardized with the unit generalized mass.

"RIGI_GENE":

The modes are normalized with the unit generalized stiffness.

"TRAN":

The modes are normalized to 1. for largest of the components of translation: (components: DX, DY, DZ).

"TRAN_ROTA" :

The modes are normalized to 1. for largest of the components of translation and rotation (components: DX, DY, DZ, DRX, DRY, DRZ).

"EUCL" :

The modes are standardized with the euclidian norm of the components which are not LAGRANGE multipliers (component: LAGR).

"EUCL_TRAN" :

The modes are standardized with the euclidian norm of the components which are of the components of translation (components: DX, DY, DZ).

3.3 Operands NOEUD or GROUP_NO and NOM_CMP

◊/ NOEUD = No
/GROUP_NO = grno

Name of the node no or of the group of a node $grno$ where one standardizes.

| *Caution: the nodes group $grno$ must contain a single node.*

♦NOM_CMP = cmp

Name of the component of standardization to the node no or with the group of a node $grno$. This operand is compulsory so NOEUD or GROUP_NO is indicated.

The modes are normalized with $1.$ for the component cmp node no or of the group of a node $grno$.

3.4 Operands AVEC_CMP / SANS_CMP

◊/AVEC_CMP = a_cmp

a_cmp list of the names of the components used for standardization.

The modes are normalized with 1. for largest of the components of the list a_cmp some is the node.

/SANS_CMP = s_cmp

s_cmp list of the names of the components which are not used for standardization.
The modes are normalized with 1. for largest of the components which is not in the list s_cmp.

3.5 Factor key word MODE_SIGNE

This factor key word makes it possible to impose for all the modes the sign of a component of a node stipulated by the user. This factor key word can be used only for the real modes (generalized problem).

```
♦ /NOEUD = Nd
  /GROUP_NO = grnd
```

Name of the node or the group of a node where the sign of a component is imposed.

| *Caution: the nodes group* $grnd$ *must contain a single node.*

```
♦NOM_CMP = cmp
```

Name of the component of the node nd or of the group of a node $grnd$ where the sign is imposed.

```
◊SIGNE = "POSITIF"
        /"NEGATIF"
```

Signs imposed component: "POSITIF" or "NEGATIF".

3.6 Case of the standardization of a collection of modes resulting from DEFI_BASE_MODALE

If one wants to normalize a collection of modes (modal base) resulting from DEFI_BASE_MODALE , the two matrixes should be informed, of mass and stiffness, making it possible to bring up to date the modal parameters:

```
◊MASSE=masse          [matr_asse_depl_r]
                  or  [matr_asse_gene_r]
                  or  [matr_asse_pres_r]
◊RAIDE=masse          [matr_asse_depl_r]
                  or  [matr_asse_depl_c]
                  or  [matr_asse_gene_r]
                  or  [matr_asse_pres_r]
```

Indeed, in this case, information on the mass matrixes and of stiffness (and possibly of damping) on which rests modal base were lost, or modes can be resulting from various sets of matrixes. It is thus necessary to recall them to operator NORM_MODE.

In the case of a base of complex modes, one needs moreover give one damping matrix if one wants compared to to normalize generalized mass or with the generalized stiffness (cf paragraph 4.2).

```
◊AMOR=masse           [matr_asse_depl_r]
                  or  [matr_asse_gene_r]
```

3.7 Operand TITER

```
◊TITER =t
```

Titer associated with the product concept by this operator [U4.03.01].

3.8 Operand INFO

◊INFO = 1 or 2

For each mode, the name of the old norm and the name of the new norm is indicated in the message file . The printed names of the norms correspond to the key word described in paragraphs 3.2 , 3.3 , 3.4 .

4 Formulation of the rules of standardization

the various norms used as well as the definition of the various modal parameters are listed in documentation of reference [R5.01.03].

4.1 Real eigen modes

For the modes of the type `mode_meca_r` (real eigen modes) the problem generalized with the eigenvalues associated is: $(K-\omega^2 M)x=(K-(2\pi f)^2 M)x=0$
where K, M are respectively the mass matrix and the stiffness matrix of the mechanical system.

For the modelizations "`MECANIQUE`", one defines the components of the eigenvector:

- component components of u^T
- translation of rotation u^R
- components of the different λ
- LAGRANGE multipliers component (pressure and fluid potential) p_f

One calls:

- u^{TR} components of translation and rotation,
- u components other than LAGRANGE multipliers.

what leads to:

$$u^*=\begin{bmatrix} u \\ \lambda \end{bmatrix}=\begin{bmatrix} u^T \\ u^R \\ p_f \\ \lambda \end{bmatrix}$$

For the models with components of translation and rotation, the eigen mode ϕ_i provided by the algorithms of modal analysis is by default:

$$\Phi_i=\frac{u^*}{max\, u}=\frac{u^*}{max\, u^{TR}}=\Phi_i^{TR}$$

what is equivalent to the standardization obtained by key word "`TRAN_ROTA`".

With key word "`TRAN`" the mode obtained is defined by:

$$\Phi_i=\frac{u^*}{max\, u^T}=\Phi_i^T$$

For the models with components of translation only, standardization is by default:

$$\Phi_i^T=\frac{u^*}{max\, u}=\frac{u^*}{max\, u^T}$$

what is equivalent to the standardization obtained by key word "`TRAN`".

Standardization by default leads to the following generalized parameters:

- generalized stiffness ${}^T\Phi_i K \Phi_i = \gamma_i$
- generalized mass ${}^T\Phi_i M \Phi_i = \mu_i$
- from where the own pulsation $\omega_i^2 = \dfrac{\gamma_i}{\mu_i}$

standardization with the unit generalized mass is obtained by key word "MASS_GENE":

$$\Phi_i^M = \frac{\Phi_i}{\sqrt{\mu_i}} \text{ from where } {}^T\Phi_i^M M \Phi_i^M = 1. \text{ and } {}^T\Phi_i^M K \Phi_i^M = \omega_i^2$$

That with the unit generalized stiffness is obtained by key word "RIGI_GENE":

$$\Phi_i^K = \frac{\Phi_i}{\sqrt{\gamma_i}} \text{ from where } {}^T\Phi_i^K M \Phi_i^K = \frac{1}{\omega_i^2} \text{ and } {}^T\Phi_i^K K \Phi_i^K = 1.$$

the standardization of the eigen mode to euclidian norm "EUCL" is obtained naturally by:

$$\Phi_i^{\|u\|} = \frac{u^*}{\|u\|} = \frac{u^*}{\sqrt{\sum_j (u_j)^2}}$$

The standardization of the eigen mode to euclidian norm "EUCL_TRAN" is:

$$\Phi_i^{\|u^T\|} = \frac{u^*}{\|u^T\|} = \frac{u^*}{\sqrt{\sum_j (u_j^T)^2}}$$

4.2 Complex eigen modes

For the modes of the type mode_meca_c (complex eigen modes) resulting from a resolution of a quadratic problem to the eigenvalues $\lambda^2 M + \lambda C + K = 0$ where C is the damping matrix of the mechanical system, one normalizes the modes Φ compared to the associated linearized problem:

$$\left(\lambda \begin{bmatrix} 0 & M \\ M & C \end{bmatrix} + \begin{bmatrix} -M & 0 \\ 0 & K \end{bmatrix}\right)\begin{pmatrix} \lambda \Phi \\ \Phi \end{pmatrix} = 0$$

The eigen mode is normalized with the unit generalized mass ("MASS_GENE"), if Φ_i satisfied:

$$\left(\lambda\, {}^T\Phi_i \;\; {}^T\Phi_i\right)\begin{bmatrix} 0 & M \\ M & C \end{bmatrix}\begin{pmatrix} \lambda \Phi_i \\ \Phi_i \end{pmatrix} = 1.$$

with the unit generalized stiffness ("RIGI_GENE"), if Φ_i satisfied:

$$\left(\lambda\, {}^T\Phi_i \;\; {}^T\Phi_i\right)\begin{bmatrix} -M & 0 \\ 0 & K \end{bmatrix}\begin{pmatrix} \lambda \Phi_i \\ \Phi_i \end{pmatrix} = 1.$$

For the other norms, the definitions are equivalent to those defined for the real modes, it is enough to replace the scalar product by the hermitian product.

5 Updated modal parameters

a concept of the mode_meca type can contain, for each mode, the following modal parameters (visible for example by printing the concept with command IMPR_RESU with FORMAT=' RESULTAT' with option TOUT_PARA=' OUI'):

Heading of the parameter in Code_Aster	Definition
FREQ	Eigenfrequency (damped, if necessary)
AMOR_GENE	Modal damping generalized
AMOR_REDUIT	Modal damping reduces
FACT_PARTICI_D* (* = X or Y or Z)	Participation factor of the mode in the direction D*
MASS_EFFE_D* (* = X or Y or Z)	Masses modal effective in the direction D*
MASS_EFFE_UN_D* (* = X or Y or Z)	Masses modal effective unit in the direction D*
MASS_GENE	Generalized mass of clean
mode	OMEGA2 Pulsation (deadened, if necessary) to square
RIGI_GENE	Stiffness generalized of the mode

Table 5.1 : list modal parameters.

These parameters are mathematically defined in documentation of reference [R5.01.03].

Operator NORM_MODE calculates or updates the following modal parameters, which depend on selected standardization: FACT_PARTICI_D*, MASS_GENE and RIGI_GENE. He also enriches data structure with parameters MASS_EFFE_UN_D* (independent of standardization).
The other parameters are independent of standardization.

6 Examples for real modes

For the modes of the type `mode_meca` (real eigen modes) resulting from a resolution of a problem generalized with the eigenvalues $(K-\lambda M)x=0$:

- to normalize an eigenvector x with the unit generalized stiffness is equivalent so that x Standardization

  $$x^T K x=1$$

  with duplication satisfies with the concept `mode_meca`:

```
mo_2    = NORM_MODE (  MODE =mo_1    ,
                          NORME=' RIGI_GENE'
                    );
```

- to normalize an eigenvector x with the unit generalized mass is equivalent so that x Standardization

  $$x^T M x=1$$

  with the unit generalized mass satisfies, with crushing of the concept `mode_meca` :

```
Mo   = NORM_MODE        (   reuse = Mo,
                            MODE  = Mo,
                            NORM = "MASS_GENE"
                        );
```