
Operator NORM_MODE

1 Goal

The role of the order is to normalize clean modes according to a criterion chosen by the user.

The operator of modal calculation `CALC_MODES` [U4.52.02] produced a concept of the type `mode_meca` or `mode_meca_c` of which clean modes real or complex are standardized in such way that largest of the components which is not a multiplier of `LAGRANGE`, that is to say equal to 1.

The operator `NORM_MODE` allows the user to choose another method of standardisation, for example generalized mass, generalized rigidity...

According to the standardisation chosen, the modal parameters (factor of participation, 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]

    ◇ / NORMALIZES = / 'MASS_GENE',
                                / 'RIGI_GENE',
                                / 'TRAN',
                                / 'TRAN_ROTA',
                                / 'EUCL',
                                / 'EUCL_TRAN',
    / GROUP_NO = grno,                               [group_no]
    # If GROUP_NO
    ◆ NOM_CMP = cmp,                                 [KN]
    / SANS_CMP = s_cmp,                             [1_Kn]
    / AVEC_CMP = a_cmp,                             [1_Kn]

    ◇ MODE_SIGNE = _F (
        ◆ GROUP_NO= grno,                           [group_no]
        ◆ NOM_CMP = cmp,                             [KN]
        ◇ SIGN = / 'POSITIVE',                       [DEFECT]
                / 'NEGATIVE',
        )

    ◇ MASS      = mass,                               [matr_asse_depl_r]
                                                    or [matr_asse_gene_r]
                                                    or [matr_asse_pres_r]
    ◇ STIFF     = mass,                               [matr_asse_depl_r]
                                                    or [matr_asse_depl_c]
                                                    or [matr_asse_gene_r]
                                                    or [matr_asse_pres_r]
    ◇ AMOR      = mass,                               [matr_asse_depl_r]
                                                    or [matr_asse_gene_r]

    ◇ TITLE = T,                                     [1_Kn]

    ◇ INFORMATION = / 1,                             [DEFECT]
                    / 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 type `mode_*` which one wants to change the standardisation of the clean modes. If `m_out` is identical to `m_in` and if the keyword 'reuse' is activated with the value `m_out`, the renormalisation is done in place.

3.2 Operand **NORMALIZES**

◇ / `NORMALIZES =`

Reference symbol of the selected standard.

'`MASS_GENE`' :

The modes are standardized with the unit generalized mass.

'`RIGI_GENE`' :

The modes are normalized with unit generalized rigidity.

'`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`, `MARTINI`, `DRZ`).

'`EUCL`' :

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

'`EUCL_TRAN`' :

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

3.3 Operands **GROUP_NO** and **NOM_CMP**

◇ `GROUP_NO = grno`

Name of group of a node *grno* where one standardizes.

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

◆ `NOM_CMP = cmp`

Name of the component of standardisation to the node *no* or with the group of a node *grno*. This operand is obligatory if `GROUP_NO` is well informed.

The modes are normalized with 1. for 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 the standardisation.

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 components which are not used for the standardisation.

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

3.5 Keyword factor MODE_SIGNE

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

◆ GROUP_NO = grnd

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

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

◆ NOM_CMP = cmp

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

◇ SIGN = / 'POSITIVE'
/ 'NEGATIVE'

Imposed sign of the component: 'POSITIVE' or 'NEGATIVE'.

3.6 Case of the standardisation of one collection of modes resulting from DEFI_BASE_MODAL

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

```
◇ MASS = mass [matr_asse_depl_r]
or [matr_asse_gene_r]
or [matr_asse_pres_r]
◇ STIFF = mass [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 matrices of mass and rigidity (and possibly of damping) on which rests the modal base were lost, or modes can be resulting from various sets of matrices. It is thus necessary to recall them to the operator `NORM_MODE`.

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

```
◇ AMOR = mass [matr_asse_depl_r]
or [matr_asse_gene_r]
```

3.7 Operand TITLE

◇ TITLE = T

Title associated with the concept produced by this operator [U4.03.01].

3.8 Operand INFORMATION

◇ INFORMATION = 1 or 2

For each mode, the name of the old standard and the name of the new standard are indicated in the file MESSAGE . The printed names of the standards correspond to the keyword described in the paragraphs 3.2 , 3.3 , 3.4 .

4 Formulation of the rules of standardisation

The various standards used as well as the definition of the various modal parameters are listed in the reference material [R5.01.03].

4.1 Real clean modes

For the modes of the type `mode_meca_r` (real clean 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 matrix of mass and the matrix of rigidity of the mechanical system.

For modelings 'MECANIQUE', one defines the components of the clean vector:

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

One calls:

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

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 clean 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 standardisation obtained by the keyword 'TRAN_ROTA'.

With the keyword '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, the standardisation is by default:

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

what is equivalent to the standardisation obtained by the keyword 'TRAN'.

The standardisation by default leads to the following generalized parameters:

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

The standardisation with the unit generalized mass is obtained by the keyword 'MASS_GENE':

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

That with unit generalized rigidity is obtained by the keyword 'RIGI_GENE':

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

Standardisation of the mode suitable for the euclidian norm 'EUCL' is obtained naturally by:

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

Standardisation of the mode suitable for the 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 clean modes

For the modes of the type `mode_meca_c` (complex clean modes) resulting from a resolution of a quadratic problem to the eigenvalues $\lambda^2 M + \lambda C + K = 0$ where C is the matrix of damping 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 clean mode is normalized with the unit generalized mass ('MASS_GENE'), if Φ_i satisfied:

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

with unit generalized rigidity ('RIGI_GENE'), if Φ_i satisfied:

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

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

5 Updated modal parameters

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

Heading of the parameter in Code_Aster	Definition
FREQ	Eigen frequency (deadened, if necessary)
AMOR_GENE	Generalized modal damping
AMOR_REDUIT	Reduced modal damping
FACT_PARTICI_D* (* = X or Y or Z)	Factor of participation of the mode in the direction D*
MASS_EFFE_D* (* = X or Y or Z)	Effective modal mass in the direction D*
MASS_EFFE_UN_D* (* = X or Y or Z)	Unit effective modal mass in the direction D*
MASS_GENE	Generalized mass of the mode
OMEGA2	Own pulsation (deadened, if necessary) squared
RIGI_GENE	Generalized stiffness of the mode

Table 5.1 : list of the modal parameters.

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

The operator `NORM_MODE` calculate or updates the following modal parameters, which depend on the selected standardisation: `FACT_PARTICI_D*`, `MASS_GENE` and `RIGI_GENE`. It also enriches the structure of data with the parameters `MASS_EFFE_UN_D*` (independent of the standardisation).

The other parameters are independent of the standardisation.

6 Examples for real modes

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

- to normalize a clean vector x to unit generalized rigidity is equivalent so that x satisfies

$$x^T K x = 1$$

Standardisation with duplication of the concept `mode_meca` :

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

- to normalize a clean vector x to the unit generalized mass is equivalent so that x satisfies

$$x^T M x = 1$$

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

```
Mo = NORM_MODE ( reuse = Mo,  
                 MODE = Mo,  
                 = 'MASS_GENE NORMALIZES'  
                 );
```