# Macro-command MACRO_MODE_MECA

## 1 Drank

This macro-command makes it possible to launch **a succession of real computation of eigen modes on a set of contiguous frequential sub-bands**.
The following actions are carried out: obtaining the modes by simultaneous iterations in the sub-bands specified, application of a norm, filtering according to a criterion of value of modal parameter higher than a certain threshold, and finally, concatenation of data structures calculated in only one.

Into term of operator *Code_Aster*, that results in a stage of pre-estimate (*via* INFO_MODE [U4.52.01]) amongst frequencies present in each sub-band. One then will calculate by the command in an effective way these modes MODE_ITER_SIMULT [U4.52.03].
In postprocessing, one controls by default a good amount of frequencies calculated in the total tape *via* a last INFO_MODE. Then the associated modes are normalized (by NORM_MODE [U4.52.11]), are filtered and concaténés (by EXTR_MODE [U4.52.12]).

Compared to a simple call to MODE_ITER_SIMULT , **this macro allows to optimize the costs computation** and **the accuracy of the results** by treating intervals of more reduced size.
**In parallel mode**, one benefits fully from this decomposition in quasi-independent modal computations (cf §3.7). For example, with a computation cut out in 10 sub-bands balanced via a preliminary calibration with INFO_MODE, one can **accelerate computing the time of a factor 10 to 20 by means of about forty processors**. The peak report can also lower few tens of for hundred.

Ideally the intervals should comprise only a few tens of modes and be balanced. The optimal figure depends on the modal solver, his parameter setting and the study. Often about forty modes by sub-band is a good figure into sequential. One can reduce their size in parallel according to the number of processors available.

This operator produces a concept mode_meca .
In a first approach one can be satisfied to inform the parameters: MATR_* and FREQ.

# Contents

# 2 Syntax

```
mod_meca =MACRO_MODE_MECA      (

# Characteristic of computation (left 1/2)
       will ♦MATR_RIGI=matra
[matr_asse_DEPL_R]
       will ♦MATR_MASS=matra
[matr_asse_DEPL_R]
```

| # Parameter setting of operators MODE_ITER_SIMULT and INFO_MODE |
|---|

```
# Pretreatment of the rigid modes (only if METHODE=' TRI_DIAG' )
       ◊OPTION    =          "MODE_RIGIDE"                    [DEFAULT]
                        /   "SANS"

# Choice of the method
       ◊  METHODE =          "TRI_DIAG"
                        /   "JACOBI"
                        /   "SORENSEN"                      [DEFAULT]

# Characteristic of computation (left 2/2)
       ♦CALC_FREQ     = _F   (
# the number of frequencies delimiting the intervals of computation are noted nb_freq.
                        ♦   FREQ=l_f                              [l_R]

# common Parameter setting of methods

         # Characteristic of space of projection
              ◊DIM_SOUS_ESPACE=des                                [I]
              ◊COEF_DIM_ESPACE=mse                                [I]
                    EXCLUDED ("DIM_SOUS_ESPACE", "COEF_DIM_ESPACE")

         # For pre and postprocessings
              ◊PREC_SHIFT=/0.05
[DEFAULT]
                              /p_shift                            [R]
              ◊NMAX_ITER_SHIFT=/3                                 [DEFAULT]
                                  /n_shift                        [I]
              ◊SEUIL_FREQ=/1.E-2                                  [DEFAULT]
                              /f_seuil                            [R]

         # Behavior in the event of empty band
         ◊  STOP_BANDE_VIDE=/"NON"                                [DEFAULT]
                             /"internal

OUI' # Parameter setting from methods

       # If METHODE= `SORENSEN'
                              ◊  PREC_SOREN=/0                    [DEFAULT]
                                         /pso                     [R]
                              ◊  NMAX_ITER_SOREN= /20             [DEFAULT]
                                                 /nso             [I]
                              ◊  PARA_ORTHO_SOREN=/0.717          [DEFAULT]
```

```
                                        /porso              [I]
    # If METHODE= `TRI_DIAG'
                        ◊ PREC_ORTHO=/1.E-12                [DEFAULT]
                                    /po                     [R]
                        ◊ NMAX_ITER_ORTHO=/5                [DEFAULT]
                                         /nio               [I]
                        ◊ PREC_LANCZOS=/1.E-8                [DEFAULT]
                                      /pl                   [R]
                        ◊ NMAX_ITER_QR=/30                  [DEFAULT]
                                      /nim                  [I]
    # If METHODE= `JACOBI'
                       ◊PREC_BATHE=/1.E-10                  [DEFAULT]
                                   /pbat                    [R]
                       ◊   NMAX_ITER_BATHE=/40              [DEFAULT]
                                           /nbat            [I]
                       ◊   PREC_JACOBI=/1.E-2               [DEFAULT]
                                       /pjaco               [R]
                       ◊NMAX_ITER_JACOBI=/12                [DEFAULT]
                                         /njaco             [I]
                       )
```

**# For final checks**

```
        ◊ VERI_MODE = _F    (
                        ◊ STOP_ERREUR =        "OUI'        [DEFAULT]
                                      /   "NON"
                        ◊SEUIL    =/rseuil                    [R]
                                   /1.E-6                   [DEFAULT]
                        ◊ STURM TYPE =     "OUI'
[DEFAULT]
                                   /   "NON"                [l_Kn]
                        ◊ PREC_SHIFT =/pshif                   [R]
                                     /   0.005              [DEFAULT]
                        )
```

# Parameter setting of operator NORM_MODE

```
        ♦ NORM_MODE = _F    (
                     ◊ /NORME  =      "EUCL_TRAN"
                                    /   "MASS_GENE"
                                    /   "RIGI_GENE"
                                    /   "TRAN"
                                    /   "TRAN_ROTA"         [DEFAULT]
                                    /   "EUCL"
                   ◊INFO   =/1
[DEFAULT]
                                    /2
                         )
```

# Parameter setting of operator EXTR_MODE

```
       ◊FILTRE_MODE  =_F    (
                    ◊CRIT_EXTR     =      "MASS_EFFE_UN"   [DEFAULT]
                                    /   "MASS_GENE"
                    ◊SEUIL  =/0.001                         [DEFAULT]
                                    /rseuil                    [R]
                    )
```

# Parameter setting of printing

```
◊IMPRESSION   = _F   (
                ◊CUMUL  =      "OUI'                          [DEFAULT]
                             /   "NON"
                ◊CRIT_EXTR     =      "MASS_EFFE_UN"          [DEFAULT]
                                 /   "MASS_GENE"
                ◊TOUT_PARA     =      "OUI'                   [DEFAULT]
                                 /   "NON"
                      )
```

**# Linear solver and parallelism**

◊SOLVEUR=_F (For more details to see the document [U4.50.01]).
# In parallel, one advises particularly parameter setting METHODE=' MUMPS'+RENUM=' QAMD' .

```
◊NIVEAU_PARALLELISME    =              "COMPLET"
[DEFAULT]
                                     /   "PARTIAL"
```

# Activated only in parallel mode (nb_proc>1).
# option "COMPLET" functions some is the direct linear solver if nb_proc=nb_freq-1. With the option "PARTIAL", only SOLVEUR=_F (METHODE=' MUMPS') is licit.

**#**
**Various**

```
◊INFO   =/1
[DEFAULT]
                    /2
                    );
```

# 3 Operands

## 3.1 Operands MATR_RIGI/MATR_MASS/INFO/METHODE/OPTION

They have the same meaning as in command MODE_ITER_SIMULT [U4.52.03].

## 3.2 Key word CALC_FREQ

Plays the same part as in command MODE_ITER_SIMULT [U4.52.03], has the same internal key words with the same default values, except for the few following key keys.

### 3.2.1 Operand FREQ

♦FREQ = l_f

List of frequencies (in Hertz) defining the sub-bands which one wants to study $l\_f=(f_i)_i$ (one notes nb_freq the number of frequencies of this list). One then searches the modes in the sub-bands $[\lambda_i, \lambda_{i+1}]$ with $\lambda_*=(2\pi f_*)^2$ and $i=1..nb\_f$ .
This list must comprise at least two values. These values must be arranged by order strictly growing and all positive.

**Note:**

- *Each frequency is treated only once: as a lower limit of the first sub-band for the first of the list, as a higher limit of the sub-bands which follow for the other frequencies. In particular, if this frequency is judged too near to an eigenvalue, it is shifted (cf [U4.52.01] and [R5.01.04]).*
- *The possible shift of a limit of frequency does not take place any more but only once in the initial INFO_MODE. There is not thus more risk of overlapping of intervals shifted as until in v10. One is not thus likely any more to twice calculate by error the same mode.*

◊STOP_BANDE_VIDE=/ **"NON" [DEFAULT]**
/"OUI'

Makes it possible to indicate MODE_ITER_SIMULT for each occurrence if it must stop ("OUI') or continue ("NON") if the sub-band $[\lambda_i, \lambda_{i+1}]$ would not comprise a frequency. In the same way, this parameter decides total behavior the macro one in the event of empty total tape.
Note that, of share particular operation the macro one, the value by default of this parameter is opposed to that retained for MODE_ITER_SIMULT.

## 3.3 Key word VERI_MODE

the internal operands have the same meaning as in of the same key word name in command MODE_ITER_SIMULT. Except with regard to the test of Sturm type, for which particular operation the macro one imposes other values.

◊STURM=/ **"GLOBAL" [DEFAULT]**
/"LOCAL"
/"NON"

Checking known as of STURM TYPE allowing to make sure that the algorithm used in the operator determined the exact number of eigenvalues, sub-band per sub-band ("LOCAL") or only in the tape globale1Cf[1] ("GLOBAL") (cf [U4.52.01] [R5.01.04]). The second alternative is most of the time amply sufficient and much less expensive than the first.

1 . Example of the §4.

However, when the limits provided to the test of Sturm type are close to an eigenvalue, they should be shifted (to preserve the robustness of the process). Sometimes this shift is pronounced too much and it will thus lead the test of Sturm type to include a too large interval comprising not calculated frequencies (and not wished!).
The test then will alert the user sometimes unnecessarily. After being itself assured that they were not multiple frequencies disastrous close to the limits of the tape, one can then disconnect it ("NON") or reduce the parameters of shift (to pass from PREC_SHIFT=5% to 2% for example).

For example, the interval is tested $[100,500]$ and 499.5 and 520 are eigenvalues of the problem. Because of proximity of eigenvalue 499.5 of the maximum limit 500, the test of Sturm type will have to shift the latter. By default it will take value 525. This new tape of test $[100,525]$ is now too important because it includes value 520: the test will conclude, wrongfully, that it there is a problem including a frequency in excess!
*A contrario*, if 500.1 had been an eigenvalue, the test of Sturm type would undoubtedly have made well alert the user!

**Note:**

- *In standard parallel mode ( NIVEAU_PARALLELISME=' COMPLET' ) , there is no possibility of test of local Sturm type. STURM=' GLOBAL' or " LOCAL" carry out the same processing: they check the validity of the test of Sturm type on all the sub-bands of computation.*
- *This test of post-checking is carried out besides other tests (nondisengageable and essential):*

  *Internal tests of convergence [2] with the modal solver (" SORENSEN ", " TRI_DIAG " and " JACOBI" ) flexible via key words PREC_* .*

  *Checking of the residues (cf [R 5.01.01/02] algorithm n°2/n°1) of each calculated mode (cf key word SEUIL_FREQ and SEUIL ).*

  *One makes sure finally that the frequencies exhumed for each sub-band belong well to the selected interval (except for VERI_MODE/PREC_SHIFT %).*

## 3.4 Key word NORM_MODE

Is used to define the arguments for the standardization of the modes. All the modes are normalized in the same way. The arguments are the same ones as for command NORM_MODE [U4.52.11]

## 3.5 Key word FILTRE_MODE

If it is present, is used of the command to introduce the arguments of filtering of the modes inside key keys FILTRE_MODE (an occurrence by subinterval) EXTR_MODE [U4.52.12] producing the result final one. All the modes are filtered with the same criterion.

If it is absent, the call to command EXTR_MODE produces the final result by concatenation without filtering of the eigen modes calculated in each subinterval. There is then $nb\_f-1$ keys key FILTRE_MODE having for argument TOUT_ORDRE=' OUI'.

## 3.6 Key word PRINTING

Makes it possible to possibly display the office plurality of values of a selected modal parameter, for the eigen modes calculated of result final. The internal keywords have the same meaning as in command EXTR_MODE [U4.52.12].

2 tests are expressed in the context of the "problem of work" provides for each sub-band the modal solver. Often this transformed problem is different from the initial problem. The good convergence of this stage thus does not ensure 100% that of the initial problem.

The selected modal parameter can not be the same one as that which was possibly used to filter the calculated modes.

Key word TOUT_PARA makes it possible to display after each modal computation and standardization, the value of all the modal parameters (frequency, masses effective,…).

## 3.7 Factor key word solver

◊SOLVEUR=_F (),

One has access to all the parameters of the direct linear solvers ( METHODE=' LDLT'/"MULT_FRONT"/"MUMPS" ) except those explicitly related to the final stage of descent-increase. This restriction relate to only the two parameters following of the solver MUMPS: POSTTRAITEMENTS and RESI_RELA.

In parallel, one particularly advises the parameter setting [3] METHODE=' MUMPS'+RENUM=' QAMD'.

For more details on the solvers, one will be able to consult the document [U4.50.01].

## 3.8 Operand NIVEAU_PARALLELISME

◊NIVEAU_PARALLELISME=/ **"COMPLET" [DEFAULT]**
/"PARTIAL"

the use of MACRO_MODE_MECA is to be privileged compared to a combination MODE_ITER_SIMULT+NORM_MODE+EXTR_MODE, when one deals with problems **of average or large sizes** (> 0.5M ddls) and/or that one seeks **a good part of their spectrums** (> 50 modes).
One then cuts out computation in several frequential sub-bands (cf operand FREQ). On each one of these sub-bands, a modal solver carries out the associated search for modes. With this intention, this modal solver uses a linear solver intensively.
These two bricks of computation (modal solver and linear solver) are **the dimensioning stages** of computation in term of memory consumption and time. It is on them which it should be put the accent if one wants significantly to reduce the costs computation of this operator.

However, the organization of modal computation on distinct sub-bands offers here an ideal frame of parallelism: **distribution of large computations almost indépendants4Aux**[4]. Its parallelism makes it possible to gain much in time but at the cost of a overcost in mémoire5Du[5].
If one has a sufficient number of processors (> with the number of nonempty sub-bands), one can then engage **a second level of parallelism *via* the linear solver** (if one has chooses METHODE=' MUMPS'). This one will make it possible to continue to gain in time but especially, it will make it possible to compensate for the overcost report of the first level to even decrease the peak sequential report notably.

3 to at least reduce the cost in time of the phase of analysis (sequential) of MUMPS. This parameter setting is done however to the detriment of memory consumption. But this overcost quickly proves compensated by the distribution of the data on the processors which parallelism implies.
4 expensive communications of eigenvectors near
5 makes of buffers MPI required by the communications of eigenvectors at the end of the MODE_ITER_SIMULT

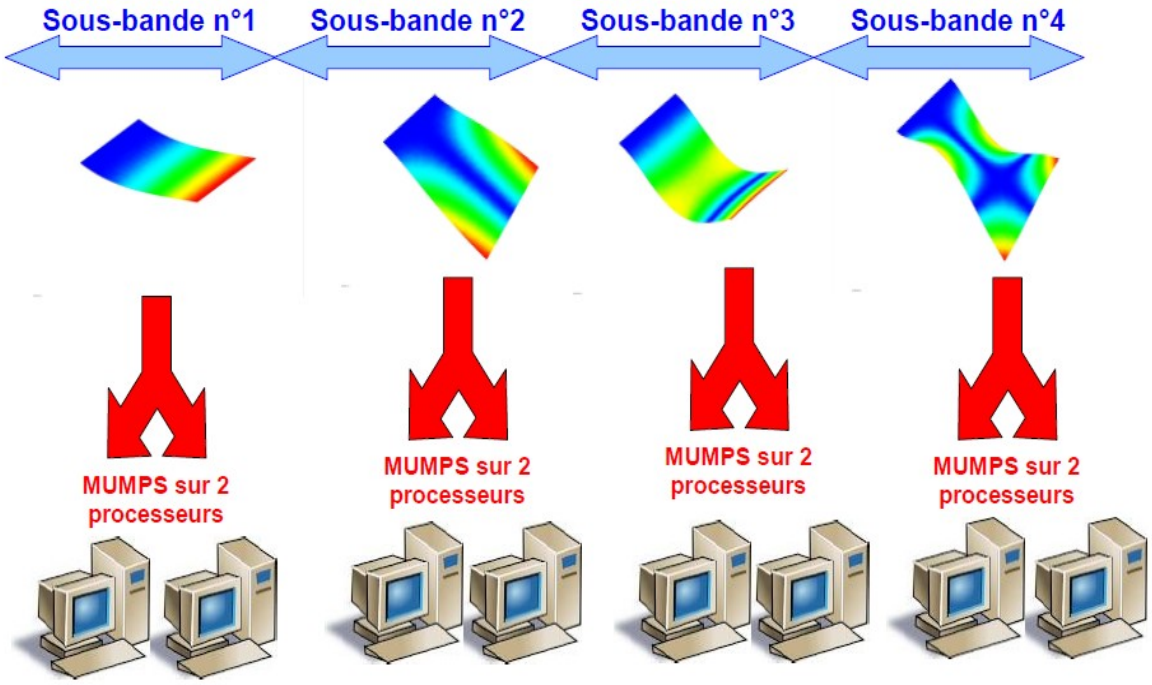
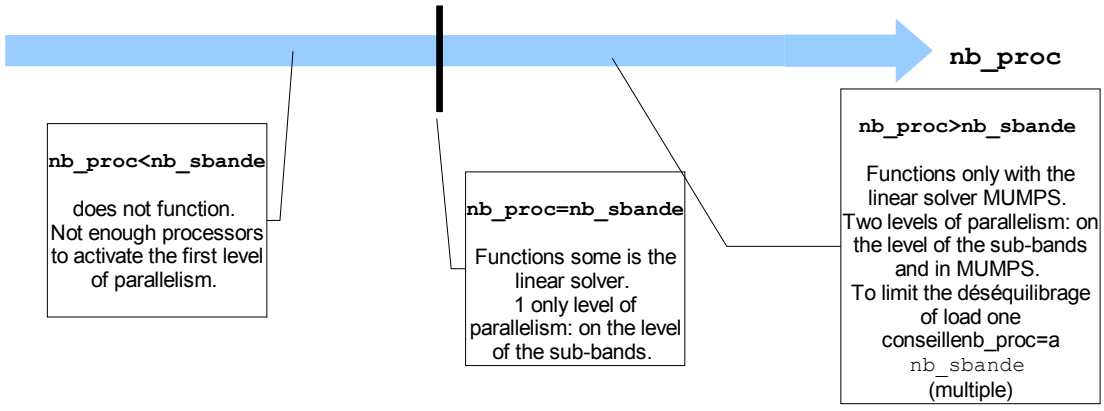



*Figure 3.8-1. Example of distribution of computations of MACRO_MODE_MECA on 8 processors with a cutting in 4 frequential sub-bands.*

This double level of parallelism (activated by default *via* key word NIVEAU_PARALLELISME=' COMPLET') then makes it possible to benefit, at best, of the two aspects.
When one truly wishes to gain in peak report because computation does not pass on the machine and that one test, without success, all the other arms of levier6Découper[6], one can knowingly choose to limit parallelism only to the level of the solver[7]: NIVEAU_PARALLELISME=' PARTIEL'. That functions only with the parallel linear solver MUMPS.

**The functional rules** are the following ones, by noting nbproc the number of parameterized processors (mitre option/mpi_nbcpu of Astk) and nb_sbande the number of nonempty sub-bands (=nb_freq-1):

•With NIVEAU_PARALLELISME=' COMPLET' (**default**): very large saving of time/improvement or average deterioration of the peak RAM report.




6 besides sub-bands, to use modal solver SORENSEN, to reduce the size of the space of projection *via* COEF_DIM_ESPACE, to use the linear solver MUMPS in OUT_OF_CORE and/or with METIS…
7 linéaire7C' is this kind of parallelism which is deployed in the rest of the code.

*Figure 3.8-2. Perimeter of use with NIVEAU_PARALLELISME=' COMPLET'.*

•With NIVEAU_PARALLELISME=' PARTIEL' : gain moderated in time/important gain on the peak RAM report.

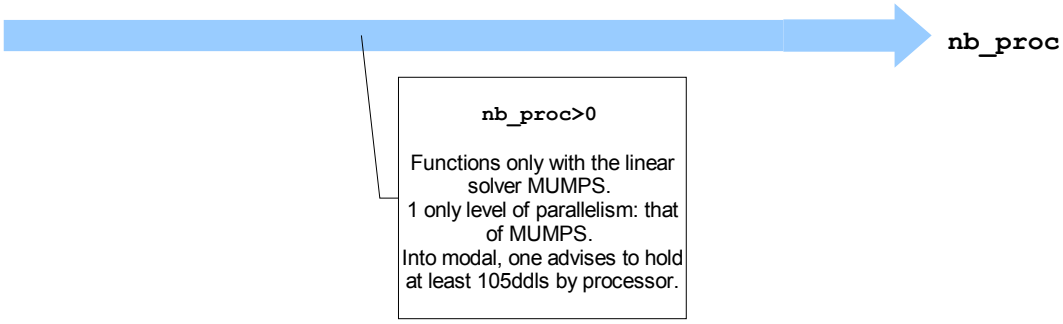



*Figure 3.8-3. Perimeter of use with NIVEAU_PARALLELISME=' PARTIEL'.*

For **an optimal use** of this activation of parallelism, he is thus advised of:

•To build relatively balanced sub-bands of computation. With this intention, one can thus, as a preliminary, gauge the spectrum studied *via* one or more call to INFO_MODE [U4.52.01]. If possible in parallel mode. Then to launch computation MACRO_MODE_MECA parallel according to the number of selected sub-bands and amongst processors available.

•To take sub-bands finer than into sequential, between 10 and 20 modes instead of 40 to 80 modes into sequential. The quality of the modes and the robustness of computation will be some increased. The peak report will be decreased by it. It however remains to have sufficient processors available (and with enough memory).

•To select a number of processors which is a multiple amongst sub-bands (not vacuums). Thus, one reduces the déséquilibrages of loads which harm the performances.

To reduce the peak report of a computation, one has several lever arms: to reduce the size of the sub-bands, to use the linear solver MUMPS (possibly in OUT_OF_CORE [U4.50.01]) and/or to parallel only this brick of computation (NIVEAU_PARALLELISME=' PARTIEL').

To use MACRO_MODE_MECA **in parallel effectively** , one thus proposes to proceed in **three stages:**

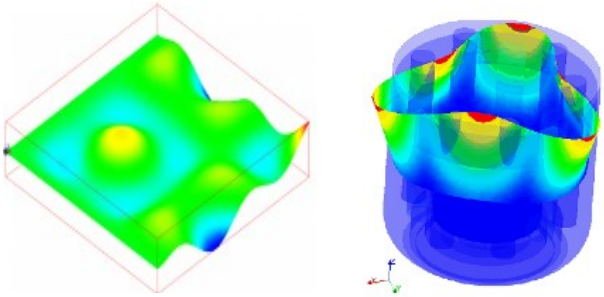
•**preliminary modal** Pre-calibrations *via* INFO_MODE. If possible, in parallel mode (potential Gains in time x70 on hundreds of processors. Gain in peak RAM report until x2).

•To examine the produced results and to break up computation into sub-bands of size modest (e.g. 20 modes) and balanced, according to the number of processors available.

•To launch in mode POURSUITE , computation MACRO_MODE_MECA parallel itself.

| **Benchmark perf016a** (N=4.0M, 50 modes) cutting in **8 sub-bands** | **Time elapsed** | **Peak report RAM** |
|---|---|---|
| 1 processor | 5524s | 16.9Go |
| 8 processors | 1002s | 19.5Go |
| 32 processors | 643s | 13.4Go |
| cutting in **4 sub-bands** | | |

| | | |
|---|---|---|
| 1 processor | 3569s | 17.2Go |
| 4 processors | 1121s | 19.5Go |
| 16 processors | 663s | 12.9Go |



| **seismic Study** (N=0.7M, 450 modes) cutting in **20 sub-bands** | **Time elapsed** | **Peak report RAM** |
|---|---|---|
| 1 processor | 5200s | 10.5Go |
| 20 processors | 407s | 12.1Go |
| 80 processors | 270s | 9.4Go |
| cutting in **5 sub-bands** | | |
| 1 processor | 4660s | 8.2Go |
| 5 processors | 1097s | 11.8Go |
| 20 processors | 925s | 9.5Go |

*Tables 3.8-1. Some results of tests of MACRO_MODE_MECA parallel with the parameters by default (+ SOLVEUR=MUMPS in IN_CORE and RENUM=' QAMD'). Code_Aster v11.3.11 on machine IVANOE (1 or 2 processes MPI by node).*

**Note:**

- *In mode NIVEAU_PARALLELISME=' COMPLET' , if the number of processors is not a multiple amongst sub-bands (not vacuums), one distributes the remainder of processors by privileging the first sub-bands. A message informs the user of the potential déséquilibrage of load and the sub-optimal character of computation.*
- *In mode NIVEAU_PARALLELISME=' COMPLET', one disabled the parallelism of elementary computations and the assemblies who can take place in NORM_MODE . Their cost is marginal in any case. This deactivation is temporary and just restricted with MACRO_MODE_MECA .*
- *In mode NIVEAU_PARALLELISME=' COMPLET', one communicates all the eigenvectors exhumed at the end of the MODE_ITER_SIMULT. Thus the distinction8La[8] between values STURM=' LOCAL' or " GLOBAL" does not take any more place to be functionally. It is not serious because the mode by default to be privileged is mode " GLOBAL".*

For implementation the practical of parallelism, one will refer to the documents credits [ U2.08.06] on parallelism, and to the paragraph dedicated of [U2.06.01] on modal computation.

8 distinction between the two modes is just here of a data-processing nature: in case "GLOBAL", the test of Sturm type is brought in work to the level of the file PYTHON the macro one, whereas in case "LOCAL", it is operated in F77 of MODE_ITER_SIMULT.

# 4 Example

Is the following sequence:

```
    mode=MACRO_MODE_MECA    (
          MATR_RIGI=rigi, MATR_MASS=masse,

          CALC_FREQ=_F (   FREQ= (1. , 3. , 5.),)
          VERI_MODE=_F (),

          NORM_MODE=_F (NORME=' TRAN_ROTA',),

          FILTRE_MODE=_F (CRIT_EXTR='
MASS_EFFE_UN'),

          IMPRESSION=_F (CUMUL=' OUI',
                         CRIT_EXTR=' MASS_EFFE_UN')
                               );
```

One will thus seek all the modes understood in the total tape $[1.,5.]$ by cutting out it in two frequential sub-bands: $[1.,3.]$ and $[3.,5.]$.
Once interpreted, the macro-command consists with the sequence of usual commands describes below.

## 4.1 Stage n°1

# Determination amongst frequencies in each sub-bands

```
      table1=INFO_MODE (MATR_RIGI=rigi, MATR_MASS=masse,
                        FREQ= (1. , 3. , 5.))
```

# Computation amongst theoretical frequencies of the tape globale9On[9]: nbmodeth
# If the total tape is empty: ALARME or ERREUR_FATALE following the value of CALC_FREQ/STOP_BANDE_VIDE.

## 4.2 Stage n°2

# Computation and standardization of the modes in each sub-bands
# to save the costs computation, one re-uses the array generated précédemment10Pour[10] and, by default, one locally does not make with each sub-band the test of Sturm type of post-checking.
# If the local sub-band is empty: ALARME or ERREUR_FATALE following the value of CALC_FREQ/STOP_BANDE_VIDE.

```
      mode_1=MODE_ITER_SIMULT ( MATR_RIGI=rigi,
MATR_MASS=masse,
                 CALC_FREQ=_F (OPTION=' BANDE',
                               FREQ= (1. , 3.),
TABLE_FREQ=table1),),
                               VERI_MODE (STURM=' NON'));
      mode_1=NORM_MODE (MODE=mode_1, reuse=mode1,
                 NORME=' TRAN_ROTA',);
```

9 sum right numbers of frequencies calculated previously and stored in table1.
10 not to remake the test of specific Sturm type of pretreatment to each sub-band.

```
      mode_2=MODE_ITER_SIMULT ( MATR_RIGI=rigi,
MATR_MASS=masse,
                CALC_FREQ=_F (OPTION=' BANDE',
                              FREQ= (3. , 5.),
TABLE_FREQ=table1),),
                              VERI_MODE (STURM=' NON'));
      mode_2=NORM_MODE (MODE=mode_2, reuse=mode2,
                NORME=' TRAN_ROTA',);
```

# Checking by a test of total Sturm type of a good amount of calculated frequencies
# Determination of smallest (resp. great) frequency of the first (resp. last) nonempty sub-band: `freq_ini` (resp. `freq_fin` ).
# Computation amongst frequencies understood in the interval: `[freq_ini,freq_fin]` : `nbmodeef` .

```
      table2=INFO_MODE (MATR_RIGI=rigi, MATR_MASS=masse,
                        FREQ= (freq_ini, freq_fin))
```

# If this number of modes is different from the number of modes envisaged initially: `ERREUR_FATALE` .

## 4.3 Stage n°3

# Filtering, concatenation and printing of the calculated modes.

```
      mode=EXTR_MODE (FILTRE_MODE=_F (MODE=mode_1,
                                      CRIT_EXTR=' MASS_EFFE_UN'),
                      FILTRE_MODE=_F (MODE=mode_2,
                                      CRIT_EXTR=' MASS_EFFE_UN'),
                      IMPRESSION=_F (CUMUL=' OUI',
                                      CRIT_EXTR=' MASS_EFFE_UN'),
                      );
```