Implementation of a calculation of clean modes of a structure

Summary

This document presents a total view of the various approaches available in *Code_Aster* for to calculate the clean modes of vibration of a mechanical structure. These approaches are described on the basis of simplest to implement, for standard studies, and while going gradually worms of the implementations more worked out for advanced studies.

One presents the sequences necessary of the operators of *Code_Aster*, without entering in detail of each operator.

Contents

1 Recalls: formulation of the problem

One considers a mechanical structure represented, within the framework of a modeling by finite elements, by his matrices of stiffness K, of mass M and possibly of damping C. The equation governing the evolution of the structure is written $M\ddot{x}+C\dot{x}+Kx=0$.

One wants to characterize the free vibrations of the mechanical structure, defined by Eigen frequencies $f_i=\frac{\omega_i}{2\pi}$ (ω_i : own pulsation of the n° mode i) and modal deformations x_i associated (and modal depreciation ζ_i if the model contains damping).

In the absence of damping (the simplest case and most frequent), modal calculation consists in finding the couples $\left\{\omega_i, x_i\right\}$ such as $\left(K-\omega_i^2 M\right)x_i=0$.

2 Setting in fact of the case

The setting in data for a calculation of clean modes of vibrations is classical and common to the majority of calculations of mechanics in *Code_Aster* :

- reading of the grid (operator LIRE_MAILLAGE),
- assignment of the characteristics of the model: model of type beam or 3D or…? (AFFE_MODELE),
- definition and assignment of materials (DEFI_MATERIAU and AFFE_MATERIAU) and/or assignment of the characteristics of the elements of structure (AFFE_CARA_ELEM),
- possible imposition of boundary conditions (stage goes away if the structure is completely free). The characteristic of modal calculation in *Code_Aster* currently is that it is necessary in general that the boundary conditions of displacement, if there is, are imposed by dualisation (AFFE_CHAR_MECA, keyword factor DDL_IMPO – more running – or FACE_IMPO or ARETE_IMPO) rather than by loads kinematics (AFFE_CHAR_CINE).

Figure 2-a schematize the setting in data of a problem of modal calculation.

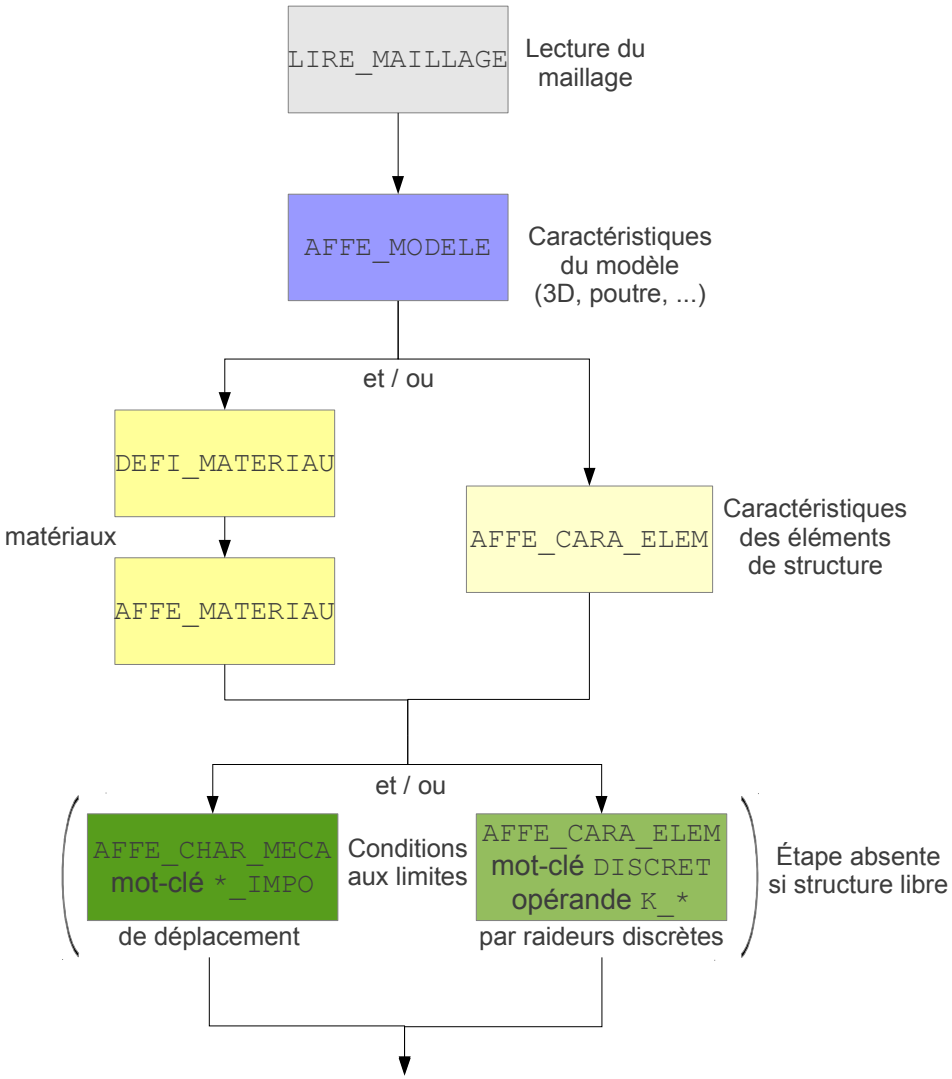

Figure 2-a : Setting in data of a problem of modal calculation.

Note:

- *For a modal calculation, no excitation is necessary, except if one wants to take into account the effect of stiffness geometrical brought by a static loading (advanced study). If necessary, the paragraph 3.2.2.3 indicate the approach to be adopted (advanced study).*
- *For a simple study, the structure is not prestressed: the possible boundary conditions in displacement are generally worthless. If one wants to take into account the prestressing generated by nonworthless displacements, it is necessary to there too adopt the approach indicated at the paragraph 3.2.2.3.*

3 Calculation of the clean modes of vibration of a structure

Starting from the data input seen in the preceding paragraph, one presents here the various opportunities given by *Code_Aster* to calculate the clean modes of a structure, while going from simplest from implementation, with more complicated sequences.

3.1 The simplest studies

For the simple studies (one considers here a structure modelled without damping, without prestressing, without fluid interaction – structure, without gyroscopy, without mixed modeling (i.e all the finite elements are of the same dimension: voluminal either surface or telegraphic)), the most ergonomic solution is to use the wizard of modal analysis available in the module Aster of the software Salomé-Méca.

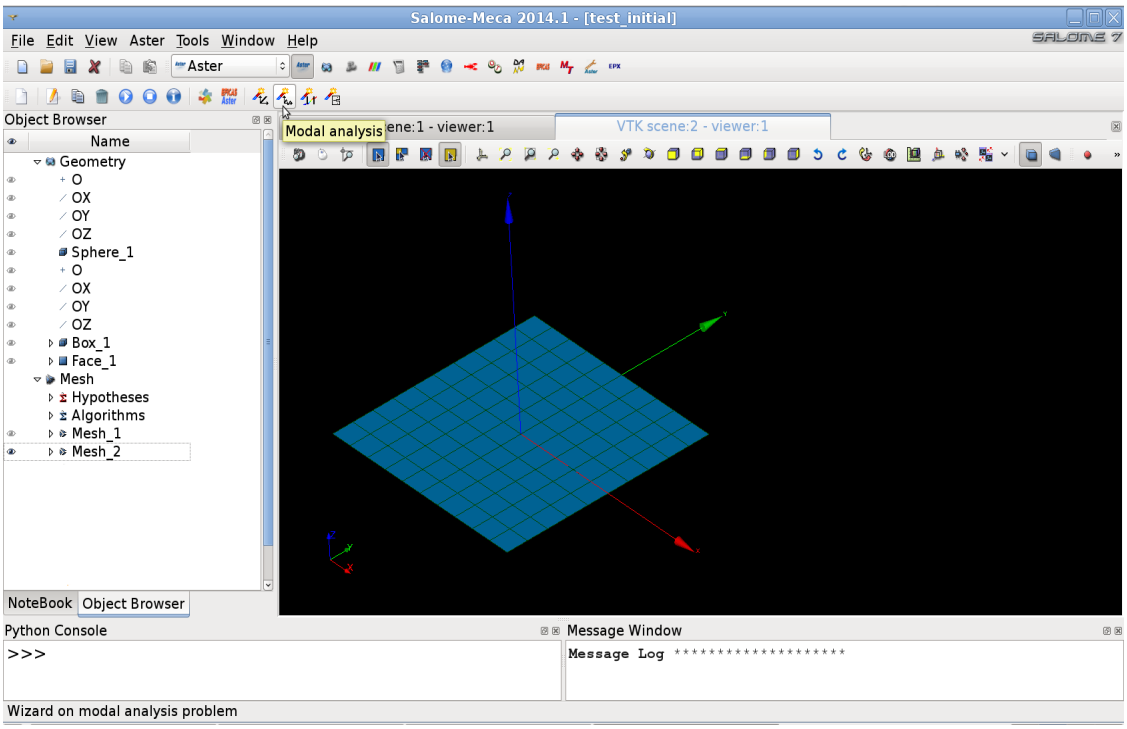

Figure 3.1-a: Localization of the wizard of modal analysis in Salomé-Méca.

The wizard request simply with the user to provide the grid, to inform the data about materials, the possible boundary conditions. And finally the search criterion of the clean modes (for example: NR first modes? or modes on a given frequential tape? or modes around a target frequency?).
The wizard then generates automatically the command file *Code_Aster*, visible in the tree structure of the study, and that the user can open and publish (with the graphic interface Eficas or a text editor) if it is worth for example to modify certain parameters or options.
Finally, the execution of calculation generates automatically a file message (extension .mess) containing the detail of all calculations, as well as a file result (with format MED) containing the clean modes. One can open this file in the ParaViS module in order to visualize the deformations (paragraph 5.1).

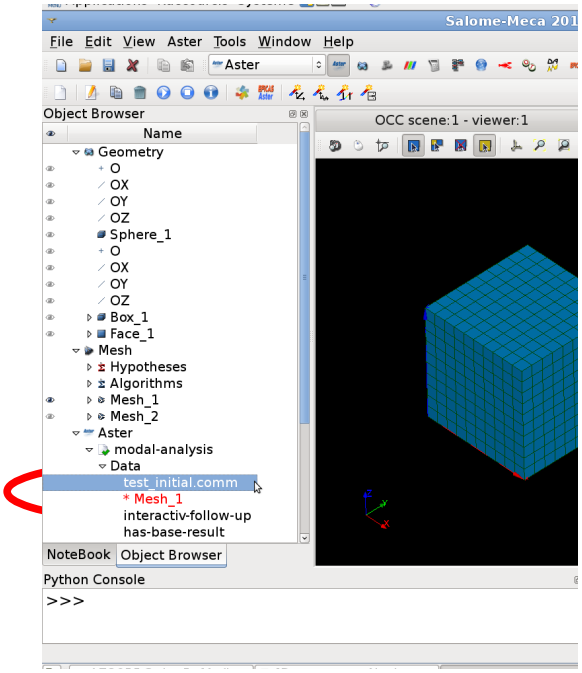

Figure 3.1-b: Command file generated by the wizard.

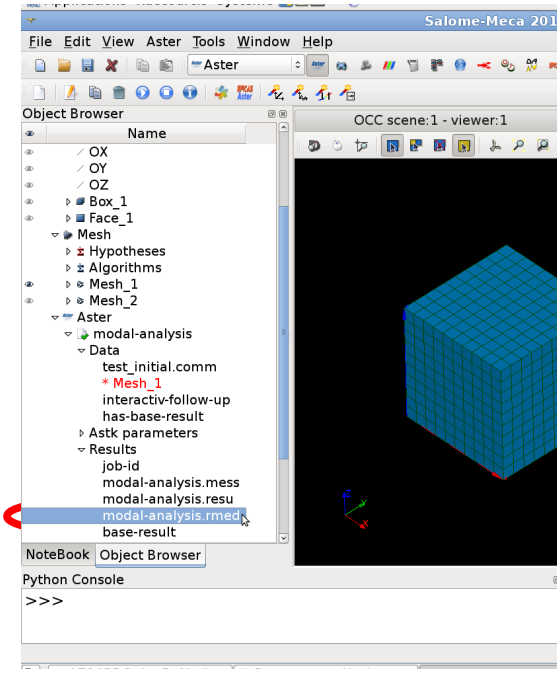

Figure 3.1-c: File result generated by the wizard.

3.2 More advanced features

3.2.1 Sequence of the orders *Code_Aster*

Starting from the data input, one proceeds as follows:

- calculation of the assembled matrices (operator ASSEMBLY with the options 'RIGI_MECA' and 'MASS_MECA' ; other options exist to calculate more specific matrices, for example geometrical rigidity, damping, the gyroscopy, etc.);
- calculation of the clean modes: operator CALC_MODES, with the keyword OPTION and the keyword factor CALC_FREQ being used to specify the search criterion.

OPTION=' BANDE' or 'PLUS_PETITE' or 'CENTER' or 'ALL' is to be privileged for its performances CPU if one seeks a relatively significant number of modes (up to 50 to 80; beyond, one recommends to cut out research in several sub-bands, cf paragraph 3.2.4.1), in particular with the option of research on a given tape (OPTION=' BANDE') for a better robustness.

On the contrary, OPTION=' PROCHE' or 'SEPARATE' or 'ADJUSTS', more expensive, is to be used rather to calculate some modes with very a good quality, for example if one wants to refine first estimates of clean modes (cf paragraph 3.2.3).

Initially, it is advised to leave the parameters by default of these operators, and to specify only the search criterion of the Eigen frequencies (by default, the first 10 modes).

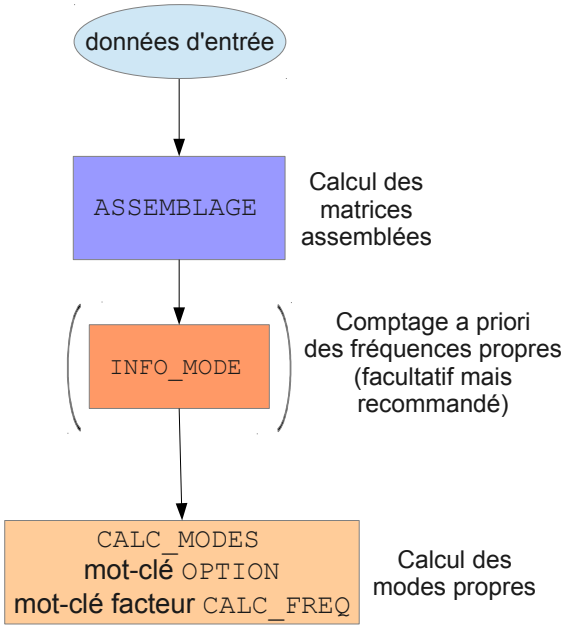

Figure 3.2-a : Calculation of the modes via the assembled matrices.

Example:

calculation of the clean modes on the tape $[20\ ;300]\,Hz$:

```
ASSEMBLY (MODEL = model,
            CHAM_MATER = ch_mat,
            CARA_ELEM = cara_el,
            LOAD = c_limite,
              NUME_DDL = CO ("numbered"), # creation of a classification
of
                                          # DDL
            MATR_ASSE = (
                                       _F (MATRICE= CO ("matr_k"), OPTION=
'RIGI_MECA'),
                                       _F (MATRICE= CO ("matr_m"), OPTION=
'MASS_MECA'),
                         )
          )

modes = CALC_MODES (MATR_RIGI = matr_k,
                    MATR_MASS = matr_m,
                    OPTION = 'BAND',
                    CALC_FREQ =_F (FREQ = (20. , 300.))
                   )
```

3.2.2 Utility of an intermediate calculation of the assembled matrices: some examples

3.2.2.1 Preliminary counting of the Eigen frequencies

Before the calculation itself of the clean modes, it is strongly recommended to carry out a counting of the clean modes contained in one or the wavebands given (in the standard case of real modes; if the modes to be calculated are complex, it will act of a counting around a point of the complex plan). This counting is much faster than the calculation strictly speaking of the clean modes.

Knowledge *a priori* amongst Eigen frequencies contained in the band searches has a double utility of checking and optimization of performances CPU of modal calculation:

- checking: one can check that the number of clean modes calculated by the modal solvor is indeed equal to the counted number of clean modes *a priori* ;
- optimization CPU: if the number of Eigen frequencies counted on the frequential tape of research is too high (a threshold ranging between 50 and 80 is usually noted), the user will be able to cut out his band of research in several sub-bands, thanks to the operator CALC_MODES with the option 'BAND' cut out in sub-bands (cf paragraph 3.2.4.1).

The counting of the clean modes is carried out by the operator INFO_MODE. In a first approach, the user can be satisfied to inform:

- in the standard case of real modes: matrices of the structure with the keyword MATR_* like its (its) band (S) of research with the keyword FREQ ;
- in case of complex modes (structures with damping for example): it is necessary to specify TYPE_MODE=' COMPLEXE' and to inform the disc of research in the plan complexes by RAYON_CONTOUR (and possibly CENTRE_CONTOUR) instead of FREQ.

3.2.2.2 Hysteretic structures with damping

It is necessary to explicitly calculate the assembled matrix of overall rigidity including the hysteretic contribution (complex matrix).
The sequence of the operators is the following:

- calculation of the assembled matrices of hysteretic rigidity (calculated starting from the elastic matrix of rigidity) and of mass (ASSEMBLY with the options 'RIGI_MECA_HYST' and 'MASS_MECA' respectively);
- modal calculation with like entry the matrix of hysteretic rigidity (complex) and stamps it of mass (CALC_MODES).

One will refer to documentation [U2.06.03] for more information on the hysteretic catch in depreciation account in *Code_Aster*.

3.2.2.3 Taking into account of prestressings

The taking into account of prestressed (boundary conditions nonworthless, static external loadings,…) require to calculate the assembled matrices of mechanical and geometrical rigidity. One can then combine them to form the assembled matrix of overall rigidity which is that used for modal calculation.
The sequence of the operators is the following, in a relatively simple case of static external loading:

- definition of the external loading (operator AFFE_CHAR_MECA, with for example the keyword FORCE_NODALE),
- calculation of the stress field associated with this loading (operator MECA_STATIQUE or STAT_NON_LINE or MACRO_ELAS_MULT to calculate the static answer, then CREA_CHAMP with the keyword OPERATION=' EXTR' to recover the stress field),
- calculation of the assembled matrices of mechanical rigidity, geometrical rigidity associated with the mass and stress field, (ASSEMBLY),
- combination of the matrices of mechanical rigidity and geometrical rigidity to form the matrix of overall rigidity (COMB_MATR_ASSE),
- modal calculation with like entry the matrix of overall rigidity and the matrix of mass (CALC_MODES).

Example:

CAS-test SDLL101 presents an example of calculation of the modes of a beam subjected to static forces.

3.2.2.4 Taking into account of the gyroscopy (revolving machines)

Besides the other assembled matrices (of stiffness, mass and possibly damping other than gyroscopic), it is necessary to calculate the gyroscopic matrix of damping with the option

'MECA_GYRO'. The operator CALC_MODE_ROTATION then allows to calculate the clean modes of the structure for various number of revolutions defined by the user under the keyword VITE_ROTA.
One can then plot the diagram of Campbell (evolution of the Eigen frequencies according to the number of revolutions) of the revolving structure thanks to the operator IMPR_DIAG_CAMPBELL.

Note:

The operator CALC_MODE_ROTATION is actually an macro-order calling CALC_MODES : if need be, the user can thus carry out the various elementary stages of CALC_MODE_ROTATION "with the hand" but in a way much less ergonomic. CAS-test SDLL129 illustrates the approach in the case of a rotor with stages whose characteristics depend on the number of revolutions.

3.2.2.5 Use of the modes for a dynamic calculation on modal basis

It is there too necessary to have access to the assembled matrices: their projection on a modal basis provides the generalized matrices usable for a dynamic calculation, with performances CPU much better than the direct use of the assembled matrices. This method of reduction of model is described in the use and reference materials [R5.06.01] [U2.06.04].

3.2.3 To improve quality of the clean modes

One draws attention to the fact that **the quality of a modal calculation depends above all on quality on the data input and physical modeling**. One can in particular quote:

- the choice of the boundary conditions: is they representative of reality? Their influence is strong on the result of calculation;
- the smoothness of the grid: a study of convergence of the grid is necessary, as for any digital study;
- the choice of modeling: by elements of structure (beam, hull,…) or in 3D? For example, for a hurled structure, a modeling in beam will be generally better than a modeling 3D even with a good smoothness of grid.

If the data input and modeling are fixed, it is possible to improve "data-processing" quality of the result thanks to the keyword AMELIORATION=' OUI' who automatically carries out a chaining of two modal calculations: the first which gives a first estimate of the clean modes (Eigen frequencies, modal deformations,…) with the search criterion asked by the user (for example with CALC_FREQ/OPTION=' BANDE' or 'PLUS_PETITE' who use the method of subspace), the second by the method of the powers opposite (CALC_FREQ/OPTION=' PROCHE') who refines the estimate of preceding calculation.

Rq 1: The first estimate is generally already good and satisfactory. For more complicated models, it however is advised to carry out this improvement.

Rq 2: this automatic functionality is limited to the cases where the structure is represented by real symmetrical matrices, and does not comprise multiple modes or several modes of rigid body. For the other cases, or if for example one wants to improve only certain modes among all those calculated, one can of course make the sequence of calculations manually.
The sequence of the orders is then the following:

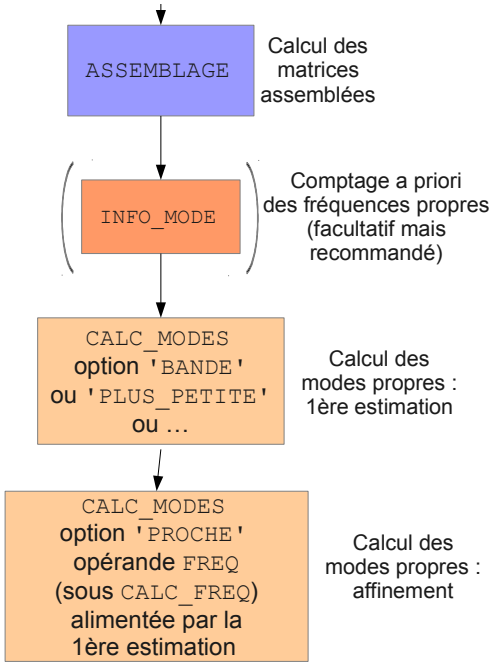

Figure 3.2-b : Improvement of the quality of the clean modes (made manually).

Example:

The first calculation of modes on the tape [0; 2000] Hz with the order below:

```
mode1 = CALC_MODES (MATR_RIGI = k_asse,
                    MATR_MASS = m_asse,
                    OPTION = 'BAND',
                    CALC_FREQ =_F (FREQ= (0. , 2000.)),
                   )
```

give the following, visible results in the file MESSAGE :

```
-----------------------------------------------------------------------

   FREQUENCIES CALCULEES INF. AND SUP. ARE:
      FREQ_INF:  4.65661E+01
      FREQ_SUP:  1.60171E+03


-----------------------------------------------------------------------
       MODAL CALCULATION:  METHOD OF SIMULTANEOUS ITERATION
                       METHOD OF SORENSEN

    NUMBER     FREQUENCY (HZ)     ERROR NORMALIZES
         1       4.65661E+01        1.82405E-07
         2       2.91827E+02        3.47786E-09
         3       8.17182E+02        9.83625E-11
         4       1.60171E+03        4.31692E-11
 NORMALIZES AVERAGE ERROR:  0.46506E-07


-----------------------------------------------------------------------

          CHECKING A POSTERIORI OF THE MODES

   IN the INTERVAL (4.64496E+01, 1.60571E+03)
   IT THERE WITH GOOD    4 FREQUENCY (S)
```

```
-------------------------------------------------------------------------
```

One can then refine for example the first two clean modes, while launching the second calculation starting from the Eigen frequencies previously calculated:

```
mode2 = CALC_MODES (MATR_RIGI = k_asse,
                    MATR_MASS = m_asse,
                    OPTION = 'NEAR',
                    CALC_FREQ =_F (FREQ = (46.6,291.8)),
                   )
```

what gives

```
-------------------------------------------------------------------------
    MODAL CALCULATION:  METHOD OF ITERATION OPPOSITE
                                                          OPPOSITE
NUMBER   FREQUENCY (HZ) DAMPING  NB_ITER  PRECISION    ERROR NORMALIZES
    1     4.65661E+01    0.00000E+00      3    3.33067E-16     3.99228E-08
    2     2.91827E+02    0.00000E+00      3    2.22045E-16     1.23003E-09
```

It is observed that the standard of error is slightly improved (certainly slightly but it is here about a very simple case).

Note:

One can also automate the recovery of the Eigen frequencies resulting from the first estimate to feed the second calculation, thanks to the language Python:

```
# recovery of the list of the Eigen frequencies estimated in the variable
Python f_estimation:
f_estimation = mode1.LISTE_VARI_ACCES () ['FREQ']

mode2 = CALC_MODES (MATR_RIGI = k_asse,
                    MATR_MASS = m_asse,
                    OPTION = 'NEAR',
                    CALC_FREQ =_F (FREQ = f_estimation),
                   )
```

3.2.4 Optimization of performances CPU

3.2.4.1 Cutting of the frequential band of research

If one searches many clean modes (either because the band of research is very broad, or because the modal density is strong), the performances of modal calculation will be better by cutting out the band of total research $[f_{min}; f_{max}]$ in several (n) sub-bands: $[f_{min}; f_2]$, $[f_2; f_3]$,..., $[f_n; f_{max}]$. That is done thanks to the operator CALC_MODES with the option 'BAND' and by specifying frequential cutting with the keyword FREQ= (fmin, f2,…, fn, fmax). To define the sub-bands, the user can be based on counting *a priori* Eigen frequencies provided by the operator INFO_MODE who can also function by sub-bands.

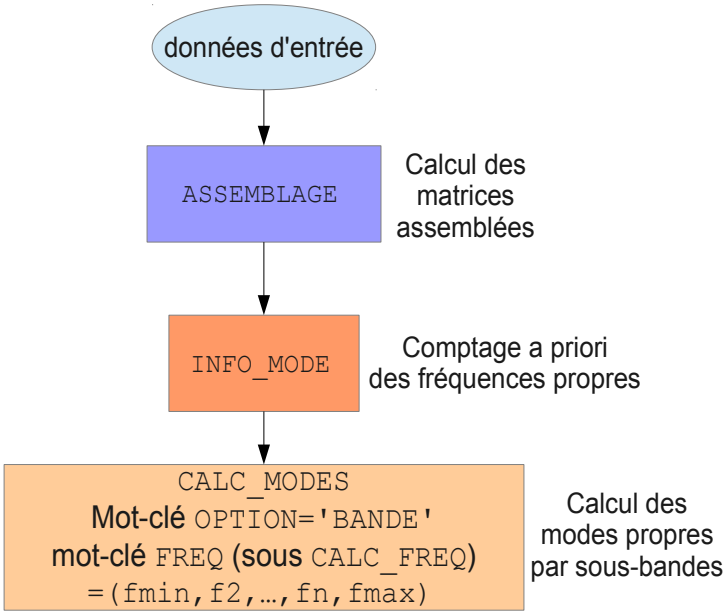

Figure 3.2-c : Calculation of the clean modes by cutting in sub-bands.

Example:

identical to the paragraph 3.2.1 by cutting out the band $[20\ ;300]\,Hz$ in three sub-bands:

```
ASSEMBLY (MODEL = model,
            CHAM_MATER = ch_mat,
            CARA_ELEM = cara_el,
            LOAD = c_limite,
            NUME_DDL = CO ("numbered"), # creation of a classification of
                                        # DDL
            MATR_ASSE = (
                         _F (MATRICE= CO ("matr_k"), OPTION= 'RIGI_MECA'),
                         _F (MATRICE= CO ("matr_m"), OPTION= 'MASS_MECA'),
                        )
          );

nb_modes = INFO_MODE (MATR_RIGI = matr_k,
                      MATR_MASS = matr_m,
                      FREQ = (20. , 300.),
                     );

modes = CALC_MODES (MATR_RIGI = matr_k,
                    MATR_MASS = matr_m,
                    CALC_FREQ =_F (FREQ = (20. , 100. , 200. , 300.))
                   );
```

Note:

- *There is a profit in performance CPU even when the sub-bands are treated sequentially (what is the case by default). The implementation of parallelism (cf following paragraph) makes it possible to improve even more the performances.*
- *For optimal performances, it is advised to have the most balanced possible sub-bands (either with a number of modes searched by relatively uniform sub-band).*

3.2.4.2 Parallelism

For modal calculation, parallelism can be levelled in work two:

- parallelization of the modal calculations carried out on each sub-band, in the operators INFO_MODE and CALC_MODES with the option 'BAND' cut out in several sub-bands ;
- parallelism on the level of linear solvor MUMPS, in the operator CALC_MODES when it is used with OPTION=' BANDE' or 'PLUS_PETITE' or 'CENTER' or 'ALL'.

To implement parallelism, it is necessary:

- to have a version of *Code_Aster* built with a parallel compiler (for example: OpenMPI,…). On the waiter centralized Aster4, parallel versions already exist: STAxx_impi;
- to select in ASTK a parallel version of *Code_Aster* ;

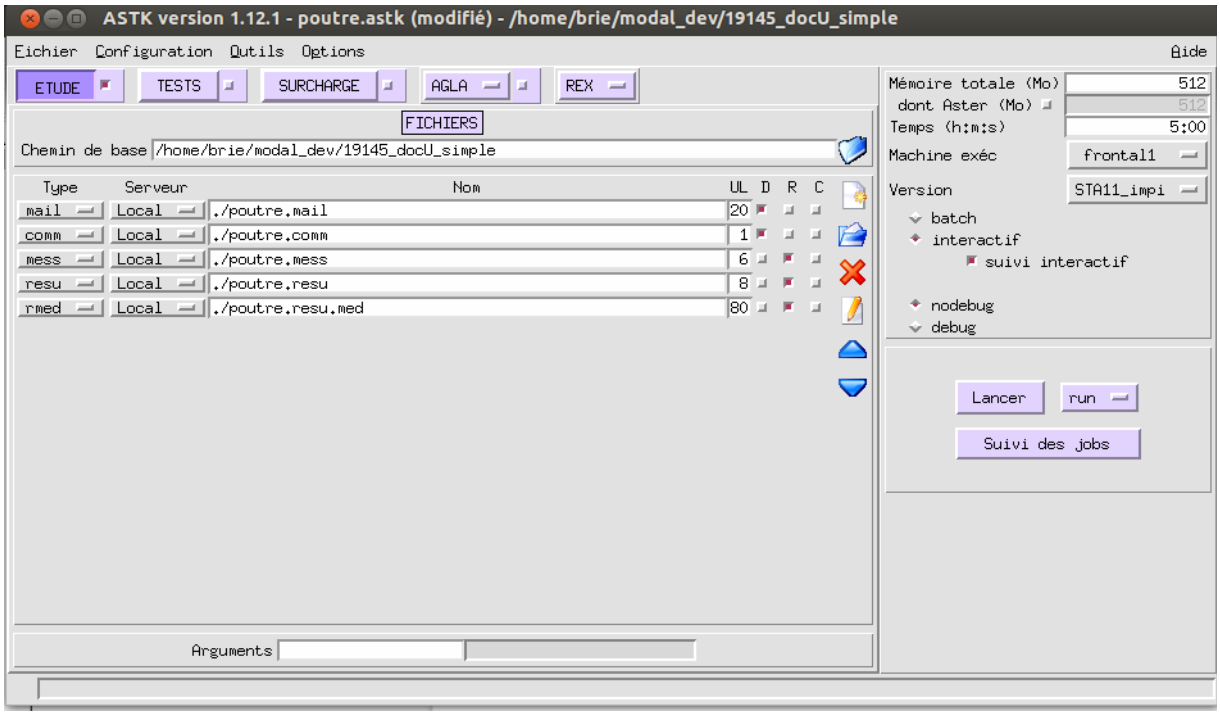

Figure 3.2-d : Selection in ASTK of a parallel version of *Code_Aster* (example on the waiter centralized Aster4).

- to specify in ASTK the number of processors and nodes of calculation to be exploited; it is necessary to use at least as many processors as of frequential sub-bands nonempty, and one advises to use a multiple number of processors amongst nonempty sub-bands (for example: if there are 5 nonempty sub-bands, to use 10 or 15 or… processors);

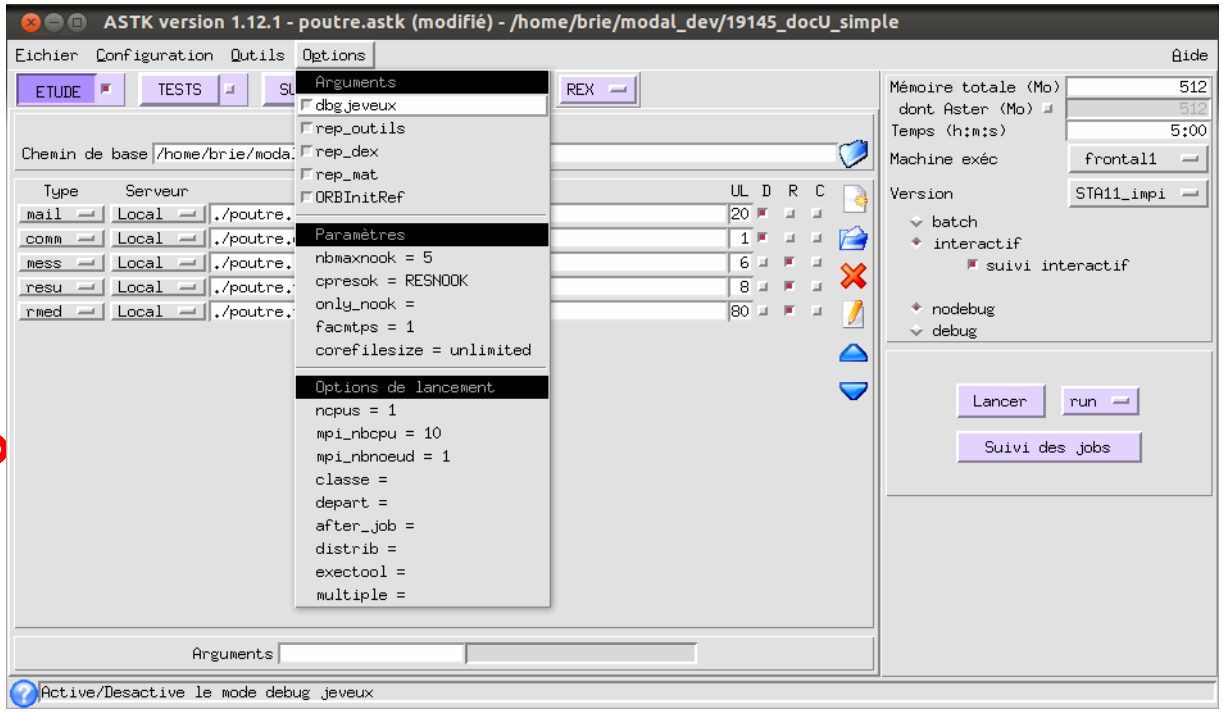

Figure 3.2-e : Declaration in ASTK amongst processors and nodes of calculation to be exploited.

- In the command file *Code_Aster* :
- to use the basic mode (parallelization of the sub-bands only) with INFO_MODE or CALC_MODES with the option 'BAND' cut out in several sub-bands, there is nothing to make: the keyword by default activate the parallelization of the sub-bands;
- to use the advanced mode (parallelization of the sub-bands for INFO_MODE and CALC_MODES with the option 'BAND' cut out in several sub-bands, and of the linear solvor for all the modal operators): to use linear solvor MUMPS (keyword factor SOLVEUR, operand METHODE=' MUMPS' ; one also recommends to parameterize the operands RENUM=' QAMD' and GESTION_MEMOIRE=' IN_CORE').

Example:

identical to the paragraph 3.2.4.1 by paralleling at the same time calculations on the sub-bands and the linear solvor:

```
modes = CALC_MODES (MATR_RIGI = matr_k,
                    MATR_MASS = matr_m,
                    CALC_FREQ = F (FREQ = (20. , 100. , 200. , 300.)),
                    NIVEAU_PARALLELISME = 'COMPLETE',
                    SOLVEUR = F (METHOD = 'MUMPS',
                                 RENUM = 'QAMD',
                                 GESTION_MEMOIRE = 'IN_CORE'),
                   );
```

Note:

For optimal performances, it is advised to have the most balanced possible sub-bands (i.e.: with a number of modes searched by relatively uniform sub-band).

The implementation of parallelism is presented in a more detailed way in documentation credits [U2.08.06] and documentations of use of INFO_MODE [U4.52.01] and of CALC_MODES with the option 'BAND' cut out in several sub-bands [U4.52.02].

3.2.4.3 Reduction of model: calculation by under-structuring

When the digital model comprises a high number of degrees of freedom or that the studied structure is an assembly of components with a grid separately, one can use methods of reduction of model per under-structuring, which rests on a geometrical partitioning of the total structure. On great models, these methods present better performances CPU that a direct calculation.

- dynamic Under-structuring

This method has a field of application very general. Documentation [U2.07.05] details its implementation.

- cyclic Under-structuring

This method has a field of application much more restrictive than the preceding one: it makes it possible to treat only structures with cyclic repetitivity (for example: aubagée wheel,…). CAS-test SDLV301 gives an example of implementation.

4 Parameters contained in a modal computation result

The execution of the operator of modal calculation is accompanied by the automatic impression of certain parameters in the file RESULT :

```
-----------------------------------------------------------------------
THE NUMBER OF DDL

   TOTAL IS:                      234

   LAGRANGE IS:                120

THE NUMBER OF ACTIVE DDL IS:         54
-----------------------------------------------------------------------
THE SELECTED OPTION IS: CENTER

THE VALUE OF SHIFT IN FREQUENCY IS:  5.00000E+01
-----------------------------------------------------------------------

 INFORMATION ON CALCULATION REQUIRES:
 MANY RESEARCH MODES     :                      1

 THE DIMENSION OF REDUCED SPACE IS:                       0
 IT IS LOWER THAN THE NUMBER OF MODES, ONE TAKES IT EQUALIZES WITH
                    4



     ============================================
     =       METHOD OF SORENSEN (CODE ARPACK)   =
     =       VERSION:  2.4                      =
     =          DATE:  07/31/96                 =
     ============================================
     MANY RESTARTINGS                       =     2
     MANY PRODUCTS OP*X                      =     7
     MANY PRODUCTS B*X                       =    20
     MANY REORTHOGONALISATIONS  (STAGE 1)  =     6
     MANY REORTHOGONALISATIONS  (STAGE 2)  =     0
     MANY RESTARTINGS OF IN NULL V0      =     0
```

```
-------------------------------------------------------------------------

   FREQUENCIES CALCULEES INF. AND SUP. ARE:
      FREQ_INF:  5.22037E+01
      FREQ_SUP:  5.22037E+01


-------------------------------------------------------------------------
       MODAL CALCULATION:  METHOD OF SIMULTANEOUS ITERATION
                           METHOD OF SORENSEN
```

Eigen frequencies

NUMBER	FREQUENCY (HZ)	ERROR NORMALIZES
2	5.22037E+01	7.02498E-10
3	6.74211E+01	9.12843E-10

```
 NORMALIZES AVERAGE ERROR:  0.80767E-09
```

position of the mode in the total spectrum

```
-------------------------------------------------------------------------

          CHECKING A POSTERIORI OF THE MODES

   IN the INTERVAL (5.20730E+01, 5.23340E+01)
   IT THERE WITH GOOD    1 FREQUENCY (S)
-------------------------------------------------------------------------
```

If the calculated modes are complex, there is in more one column giving modal depreciation:

```
-------------------------------------------------------------------------
THE NUMBER OF DDL

   TOTAL IS:                        74

   LAGRANGE IS:                  44

THE NUMBER OF ACTIVE DDL IS:         8
-------------------------------------------------------------------------
 INFORMATION ON CALCULATION REQUIRES:
 MANY RESEARCH MODES     :                     5


  with the dealt problem being quadratic, the space of research is doubled

    Method QZ in CALC_MODES: One finds a number of eigenvalues
    17 different amongst ddls physical credits 8!

 your problem is strongly deadened.
 value (S) clean (S) real (S)                 :  14
 value (S) clean (S) complex (S) with combined:  10
 value (S) clean (S) complex (S) without combined:  0
       MODAL CALCULATION:  GLOBAL METHOD OF THE TYPE QR
                           ALGORITHM QZ_SIMPLE
```

Eigen frequencies (deadened)

NUMBER	FREQUENCY (HZ)	DAMPING	ERROR NORMALIZES
1	5.52718E+00	8.68241E-03	4.00918E-13
2	1.08852E+01	1.71010E-02	7.31808E-14
3	1.59105E+01	2.50000E-02	5.40182E-14
4	2.04500E+01	3.21394E-02	4.03817E-14
5	2.43661E+01	3.83022E-02	3.48265E-14

```
 NORMALIZES AVERAGE ERROR:  0.12067E-12
```

modal depreciation

```
 Caution: for the moment, there is no checking of the type STURM (counting
of the good number of the calculated eigenvalues)
 when one is in the complex plan:
       modal problem generalized with complex MATR_RIGI,
    or modal problem generalized with matrix (S) nonsymmetrical (S),
    or quadratic modal problem (préence of keyword MATR_AMOR).

-------------------------------------------------------------------------

          CHECKING A POSTERIORI OF THE MODES

-------------------------------------------------------------------------
```

Moreover, the data-processing structure of data produced during a modal calculation can contain the following parameters:

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 4.1 : list of the modal parameters.

These parameters are mathematically defined in reference material [R5.01.03]. The user has access there by printing the contents of the structure of data with the operator IMPR_RESU with FORMAT=' RESULTAT' with the option TOUT_PARA=' OUI'.

5 Postprocessings of the clean modes

5.1 Visualization

The modal deformations calculated by one of the methods described previously can be exported in various formats in order to be visualized in platforms of mechanical calculation: format MED for the platform Salomé, format UNV,…

The user can thus graphically characterize the calculated modes: mode of inflection? mode in a given plan? local mode? etc.

Example:

impression with format MED.

```
modes = CALC_MODES (...)     # modal calculation

IMPR_RESU (FORMAT = 'MED',
           RESU=_F (RESULT = modes));
```

One can then open the file created in the platform Salomé to visualize the modal deformation, to animate it,…

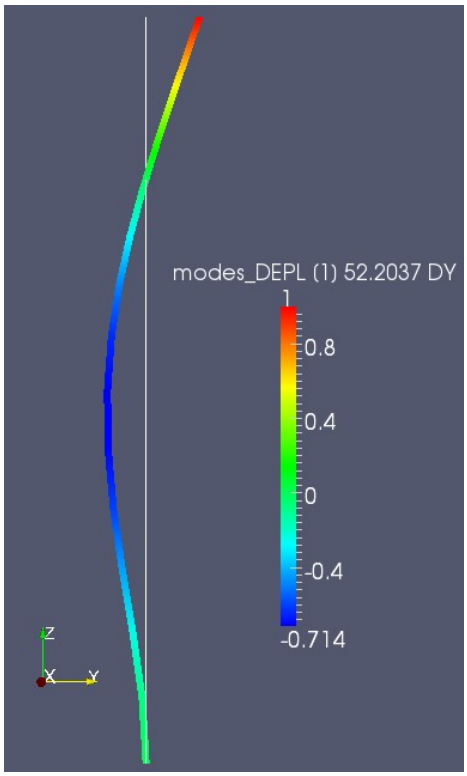

Figure 5.1-a : Visualization of a mode in Salomé (ParaViS module): here mode of inflection of order 2 of a beam.

5.2 Standardisation of the modes

The modal deformations are defined except for a multiplicative factor (cf formulation of the modal problem in the paragraph 1).

By default, modes calculated by the operator CALC_MODES are normalized so that the largest physical component is equal to 1. The user can modify this standardisation, that is to say directly in the operator CALC_MODES thanks to the keyword factor NORM_MODE, that is to say later thanks to the operator NORM_MODE [U4.52.11]. In both cases, the code also calculates 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* (which is them independent of the standardisation). These parameters (definite in the paragraph 4) can be useful in particular to eliminate from a modal base certain nondesired modes (cf paragraph 5.3).

Example:

normalizes compared to the mass.
1st possibility:

```
modes = CALC_MODES (MATR_RIGI = matr_k,
                    MATR_MASS = matr_m,
                    OPTION = 'BAND',
                    CALC_FREQ =_F (FREQ = (20. , 300.)),
                    NORM_MODE =_F (STANDARD = 'MASS_GENE')
                   )
```

2nde possibility:
```
modes = CALC_MODES (MATR_RIGI = matr_k,
                    MATR_MASS = matr_m,
                    OPTION = 'BAND',
                    CALC_FREQ =_F (FREQ = (20. , 300.)),
                   )

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

5.3 Filtering of the modes according to a criterion

From the point of view of a transitory calculation of answer for example, the user can choose to preserve in his modal base of projection, only certain modes considered to be important in the dynamic response or filling a given criterion. That can be fact is directly in CALC_MODES with the keyword factors FILTRE_MODE and IMPRESSION, that is to say thanks to the operator EXTR_MODE [U4.52.12]. One can thus filter the modes according to various options: starting from their number in the total spectrum, of their generalized mass, etc.

Example:

elimination of the modes whose unit effective mass is lower than 5 %, and posting in the file RESULT office plurality of the unit effective masses of the preserved modes.

1st possibility:
```
modes = CALC_MODES (MATR_RIGI = matr_k,
                    MATR_MASS = matr_m,
                    OPTION = 'BAND',
                    CALC_FREQ =_F (FREQ = (20. , 300.)),
                    NORM_MODE =_F (STANDARD = 'MASS_GENE'),
                    FILTRE_MODE =_F (CRIT_EXTR = 'MASS_EFFE_UN',
                                     THRESHOLD = 0.05),
                    =_F IMPRESSION (CRIT_EXTR = 'MASS_EFFE_UN',
                                    OFFICE PLURALITY = 'YES'),
                   )
```

2nde possibility:
```
modes = CALC_MODES (MATR_RIGI = matr_k,
                    MATR_MASS = matr_m,
                    OPTION = 'BAND',
                    CALC_FREQ =_F (FREQ = (20. , 300.)),
                    NORM_MODE =_F (STANDARD = 'MASS_GENE'),
                   )

modes_f = EXTR_MODE (FILTRE_MODE =_F (MODE = modes,
                                      CRIT_EXTR = 'MASS_EFFE_UN',
                                      THRESHOLD = 0.05),
                     =_F IMPRESSION (CRIT_EXTR = 'MASS_EFFE_UN',
                                     OFFICE PLURALITY = 'YES'),
                    );
```

Note:

- *The operator EXTR_MODE allows to filter with a threshold applied to one or of the directions given (keyword SEUL_X for example for direction DX), which is not possible directly in CALC_MODES currently.*