Data-processing description of CALC_ESSAI

1 Introduction

The macro-order CALC_ESSAI gather a set of functionalities in correlation calculation-test, controllable by the means of a graphic interface in Tk python:

- expansion of experimental data on a basis of digital deformations (via the macro-order MACRO_EXPANS),
- structural modification a structure known by a base of experimental modes and a simplified digital model,
- identification of efforts on a structure starting from the data of the inter-spectrum and a base of clean modes describing its behavior,
- calculation of spectra filtered and realised (via the macro-order CALC_SPEC),
- visualization of modal deformations, curves and matrices of MAC in Salomé or tools associated with Aster (GMSH, Xmgrace, Tk).

The macro-order is a distance going down from the software MEIDEE, which was used in the Nineties for the identification of turbulent efforts on the tubular structures (tubes type of or pencil steam generator of assemblies).

2 Launching, interactive mode and noninteractive

The macro-order CALC_ESSAI is launched with the routine python calc_essai_ops.py located in the /bibpyt/Macro file. This routine carries out the operations described below

2.1 Recovery of the pre-declared concepts

For the mitres of identification of the efforts and structural modification, it is necessary to declare with the call of the macro-order the name of the concepts at exit of this one, by the use of the keyword RESU_XXX = CO ('NAME'). The names of the pre-declared concepts are transported in all the classes of execution of calculation of the macro-order in the shape of a list (RESU_MODIFSTRU) or of a dictionary (RESU_IDENTIFICATION) containing the names, and, in the case of the mitre of identification, the type of result, a meter (incremented of 1 with each time the user declares a concept outgoing) and the DeclareOut routine.
For the mitre of expansion of data, the results can be declared intéractivement. However, they cannot be used in the continuation of calculation that within the framework of a continuation.
For the mitre of treatment of the signal, the leaving name of the concepts is fixed into hard (see the section 5.5).

2.2 Launching in noninteractive mode

It is not very relevant of launching the macro-order in not-interactive mode, except, possibly, in the case of the identification of efforts, mode which cannot be replaced by the use of another order. Thus, for the expansion of models, it is to better use the macro-order directly MACRO_EXPANS. For the structural modification, one can refer to the sequence of orders describes in U2.07.03 documentation. For the treatment of the signal, one can use the macro-order CALC_SPEC (see U4.32.01).
The not-interactive mode is thus used for the identification of efforts, and, generally, for the validation of the CAS-tests associated with CALC_ESSAI.
In this mode, one returns directly to the CalcEssaiTest routine.

2.3 Launching in interactive mode

Launching is done in calc_essai_ops.py in the routine create_interactive_window. In this routine:

- one imports the CalcEssaiObjets class (administrative of concepts aster),
- one creates an object TabbedWindow (routine create_tab_mess_widget): this object is a window with mitres (see section 4.1) with the window of message in foot,
- one imports all the graphic classes, called InterfaceXxxxxx,
- one creates authorities of each graphic class, which one displays in the window tabs (authority of TabbedWindow) higher definite,
- one creates a file of messages which will be displayed in the window mess defined in foot of the authority tabs.

3 File cata_ce.py: catalogues

The file meidee_cata.py was defined at the time where the methods python to handle the concepts Aster were not also developed. They nevertheless still rather useful, because are particularly adapted to the needs for calculations for CALC_ESSAI.

This file contains:

- the CalcEssaiObjects class: class containing all the concepts Aster which could be useful in calculations,
- the class Results: the concepts Aster sd_resultat are authorities,
- the ModeMeca class: subclass of the preceding one: instancie the concepts Aster mode_meca
- the DynaHarmo class: subclass of Results: instancie the concepts Aster dyna_harmo

- the classes CaraElem, ChampMateriau: the concepts Aster cara_elem and cham_mater instancient
- the InterSpectre class: instancie the concepts Aster table_sdaster of the table_fonction type containing of the frequential functions,
- the class Tempo: the concepts Aster table_sdaster of the type instancient counts function containing of the temporal functions (for the treatment of the signal),
- the Model class: instancie the concepts Aster modele_sdaster.

Only the classes most used in calculations are described here.

3.1 Description of the CalcEssaiObjects class

This class is called with the call of the macro-order, and after each calculation creating a new concept Aster in order to update the dictionary of the concepts available, with the use of the method recup_objects.
Description of the principal methods:

3.1.1 recup_objects

This method recovers all the concepts available and arranges them in a dictionary indexed by the names of those, for each type of concepts:

```
self.mode_meca = {'NOM_1': mode_meca_1, 'NOM_2': mode_meca_2}
```

If a particular class were created for the concept in question, then the object `mode_meca_X` will be an authority of this concept. The classes ModeMeca and InterSpectre are used in the current operations. The InterSpectre class has in particular a very useful method to transform the concept inter-spectrum Aster into a matrix python indexed by the frequencies of discretization and the J and sequence numbers I (or, in the place, names of nodes and components I and J).
Several methods relating to the classes in question are called to bind concepts to the nume_ddl, the models, the matrices, the ascending grids…

3.1.2 update

Allows to add a concept to the authority of MeideeObjects without having to entirely recreate it (what can be long being given the number of initializations).

3.1.3 Debug

Routine of posting of the concepts associated with a concept.

3.1.4 Routines get_xxx and get_xxx_name

The routines get_xxx_name are used to recover the list of the names of concepts xxx existing. The routine get_xxx makes it possible to recover the concept starting from its name.

3.2 Description of the ModeMeca class

The concept of mode_meca gathers indifferently the dynamic modes, resulting from **CALC_MODES**, and static modes. The first are indexed by their sequence number and their Eigen frequency, the seconds by their sequence number and the variable **NOM_CMP**, which is the degree of freedom associated with the static deformation.

3.2.1 Classes get_xxx

These classes are used to bind the concept mode_meca with the ascending concepts:

- get_matrices recovers the matrices of mass, stiffness and depreciation associated with the concept. The associated attributes are .mass, .kass and .cass.

- get_nume recovers the nume_ddl by same skew (.numeric extension),
- get_maillages recovers the associated grid, either by the means of the nume_ddl, or directly in the contained information in the .REFE of each sequence number of the result.
- get_modele recovers the associated model, either by the routine dismoi of Aster, or via the associated grid

Caution: certain routines are interdependent; during actualizations, it is advisable (as in the routine update of CalcEssaiObjects) to carry out them in a certain order.

- get_mode_data: recover the whole of the data generalized in the shape of a dictionary; for the static modes, only the fields NUME_MODE, NUME_ORDRE, and NOEUD_CMP are filled, the others are worth Nun ; for the dynamic modes, the fields **FREQ**, **AMOR_REDUIT**, AMOR_GENE, RIGI_GENE and MASS_GENE are filled, the field NOEUD_CMP is worth Nun. The routine show_cara_mod makes it possible to display the whole of the extracted modal characteristics.

3.2.2 Routine extr_matr

Routine allowing to extract the modal deformations in the form from a matrix numpy. The extraction is done with the operator CREA_CHAMP and the routine EXTR_COMP. The encapsulation of these two orders is made in the routine crea_champ at the end of the file ce_cata.py.
The routine nume_ddl_phy makes it possible to manufacture a classification of the active degrees of freedom. These DDL are from the filter which the user could apply by using the operator OBSERVATION to create the base of the modes, or by using the filter of DDL in the graphic interface.

3.3 DynaHarmo class

The routines giving access the concepts ascending of the dyna_harmo are relatively similar to those of the ModeMeca class.
DynaHarmo are used only for the expansion of models, class of calculation which carries out only calculation Aster. It is thus not necessary to extract the data with the format python from them.

3.4 Classes CaraElem, ChampMateriau

No python treatment of these concept was carried out. These classes are thus almost empty. They just make it possible to encapsulate the concept Aster corresponding and to be able to point on this one with its name.

3.5 InterSpectre class

Class allowing to treat the concepts of type inter-spectrum. The inter-spectra are tables containing of the subscripted functions of frequential variable by:

- sequence numbers (indices NUME_ORDRE_I and NUME_ORDRE_J),
- or of the nodes and components (NOEUD_I, NOEUD_J, NOM_CMP_I, NOM_CMP_J).

With the initialization of the authority of the class, one extracts the frequencies from discretization of the functions, which makes it possible to differentiate the tables from type-inter-spectrum of other type of tables (tables containing of the temporal functions, for example, which will be treated by the class Tempo).

3.5.1 Routine make_inte_spec

With the image of the operator LIRE_INTE_SPEC, this routine manufactures a inter-spectrum Aster starting from functions defined in the form of list.
If the concept is connected to a model (with set_model), through a concept result (DynaHarmo or ModeMeca), then it is associated with a classification of the physical degrees of freedom. The inter-spectrum created is thus subscripted by its nodes and components. In the contrary case, the inter-spectrum is associated with a generalized concept. It is thus subscripted by its sequence numbers.

3.5.2 Routine set_model

Allows to associate a concept mode_meca with the definite inter-spectrum. Coherence in the face of the inter-spectrum and the concept associated result must be respected (but not inevitably coherence in DDL, because the inter-spectra are not all the time defined by a physical classification.

3.5.3 Routines extr_inte_spec and extr_freq

Operation reverses routine make_inte_spec. The inter-spectrum created is a matrix numpy with 3 dimensions. Stages of the routine:

- extraction of the couples of sequence numbers or node and component I and J, according to whether the inter-spectrum is physical or modal, with `extr_nume_ordre`,
  - `coupl_ddl = [('N1_DX', 'N1_DY')… ('N10_DZ', 'N10_DZ')]`,
  - or `coupl_ddl = [(1.2)… (10.10)]`,
- extraction amongst frequencies of discretization (`nb_freq`),
- the function `set` allows to recover the sequence numbers in a list comprising only one occurrence of each one; it can be useful to dimension the nonsquare inter-spectra, for example, functions FRF and of coherences created in `CALC_SPEC` ; this functionality is however not used currently by the mitre of treatment of the signal; in this one, one extracts the functions from the inter-spectrum by `RECU_FONCTION`, and one does not carry out matric operations on the matrices; one thus adds temporarily a checking on the equality between many lines and of columns,
- association with a concept result and extraction of the active DDL (`self.nume_phy`),
- checking of coherence enters the size of the nume_phy and inter-spectrum cuts it,
- manufacturing of a list of couple of G-string for indicer in python functions of the inter-spectrum: `NOEUDS_CMP_X` or of `NUME_ORDRE_X`, according to the way in which the inter-spectrum in Aster was manufactured (`coupl_ddl`),
- extraction of the functions to the format python; according to the way in which the inter-spectrum was defined, it may be that the functions are not accessible directly in the base starting from the data from its name; the alternative solution is to use `RECU_FONCTION`,
- for the inter-spectrum subscripted by the names of the nodes and the associated components, one checks coherence with `self.nume_phy` ; if the inter-spectrum is defined only by its sequence numbers, the user must make sure itself of this coherence.

The routine extr_freq functions in an identical way to recover only the frequencies of discretization.

3.6 Class Tempo

The objective of this class is to encapsulate the concepts containing of the function tables of temporal variable. The methods of this class are rather similar to those of the InterSpectre class. The temporal functions are intended to be used by the operator `CALC_SPEC`. It is thus not necessary to extract the functions with the format python.

3.6.1 Method extr_tempo

The temporal samples are indexed by the sequence numbers, and the number of measurement, so as to be able to gather the samples having been measured simultaneously. The extraction is thus done by the method been worth, by searching the keywords `NUME_ORDRE_I`, `NUME_MES` and while returning the names `FUNCTION` associated.

3.7 Model class

This class allows to encapsulate the models Aster. The associated methods allow to extract the ascending grid and the unit from the downward nume_ddl which depend on this model (it can have several of them there. This class has little interest in itself.

4 Description of the generic graphic tools

The graphic routines are organized in the following way:
- generic routines hat and routines: in the file `outils_ihm.py`,
- graphic interfaces specific to each mitre of calculation: in a specific file python: for example, the file `ce_ihm_expansion.py` the graphic class related to the mitre of expansion of data contains, whereas the file ce_calcul_expansion.py contains associated calculations (and can be tested in noninteractive mode without calling on the Tk modules of python).

One details here the operation of the principal classes TabbedWindow and MessageBoxInteractif. For the other classes, to refer to their description heading of those. This lower school allows to display specific menus: selection of a group of nodes in a grid, choice of the parameters for an operation Aster, parameters for a change of reference mark.

4.1 TabbedWindow class

With the call of the macro-order `CALC_ESSAI` in interactive mode, the routine `create_interactive_window` launch:
- a virgin window `TabbedWindow` (in outils.py) in which are displayed the specific contents of each mitre of work.,
- a window of message `MessageBoxInteractif`, in which will be displayed the interactive message displayed in the course of calculation.

Note:
1. Interior of the virgin window `TabbedWindow` is Canvas (the advantage of Canvas on classical Frame is the possibility of adding a vertical bar of run if this one is large),
2. the routine `set_current_tab` is used to display the interface corresponding to each change of mitre; method `setup` associated with the graphic class of the mitre is called to refresh the environment (concepts Aster create); **method `setup` must thus exist in the graphic classes of all the mitres,**
3. the initial size of the windows principal and message are respectively given in the routines `__init` of definition of the aforesaid windows.

4.2 MessageBoxInteractif class

Window empties in which the messages of information are printed during calculations. As specified in the preceding section, the window is created with the call of the macro CALC_ESSAI (authority `mess`), and is transported like argument of all the graphic classes. Example:
```
iface = InterfaceCorrelation (hand, objects, macro, mess, param_visu)
```
The arguments are the following:
- `hand` : the principal window of the IHM, which contains TabbedWindow,
- `objects` : authority of CalcEssaiObjects which contains the existing concepts Aster,
- `macro` : macro-order, obsolete
- `mess` : authority of the window of messages
- `param_visu` : authority of the class IHM InterfaceParametres, which allows to choose the parameters of visualization of the deformations and the curves (Xmgrace + GMSH or Salomé).

4.2.1 Posting of the error messages resulting from exceptions

When one launches an order Aster in the macro one through an interactive order, and that calculation finishes in error (recovered in the form of an exception by Aster), the error message is poster in the interactive window of message. One recovers this error thanks to the following order:
```
try:
    MACRO_EXPANS (MODELE_CALCUL = mcfact_calcul,
                  MODELE_MESURE = mcfact_mesure,
                  NUME_DDL = digital,
```

```
                    RESU_ET = res_ET,
                    RESU_RD = res_RD,
                    RESOLUTION = parameters,
                    ** arguments)
except aster.error, err:
    message = "ASTER ERROR: " + mess.GetText ('I', err.id_message,
                                    err.valk, err.vali, err.valr)
    self.mess.disp_mess (message)
    UTMESS ('WITH', 'CALCESSAI0_7')
    return
```

Note: at the time as of last restitutions carried out, one noticed that this alignment of errors was not functional any more. Posting in interactive window was removed, and replaced by a simple “Error in calculation Aster”.

4.3 ScrollList class

Generic Frame allowing to display a table with bar of run. The attribute been worth is a list filled out by `set_values` correspondent with what is displayed in the table. get_selection returns a list containing the selected indices to the mice and the value of been worth corresponding:

```
[[1, 'N1_DX'], [3, 'N4_DY']]
```

4.3.1 Classes ModeList and ModeFreqList

Subclass of ScrollList, it has two buttons below making it possible to at a stretch select or die-select all the lines of the list. It is used to display sequence numbers and an associated value (standard Eigen frequency or `NOEU_CMP`). `get_selection` turn over only the list of the selected sequence numbers.
The ModeFreqList class, under class of ModeList is to be used preferentially because it makes it possible to correctly format the values of the Eigen frequencies or of `NOEU_CMP`.

4.3.2 StudyList class

Subclass of ScrollList, it makes it possible to display the list of the studies open Salomé. It has a method `to validate` who stores the study current Salomé for the posting of the deformations, MAC and curves.

4.4 VisuSpectre class

This class creates a frame made up of one or more tables for the posting of the coordinates of the inter-spectrum to visualize. This class is used for the mitre of identification of efforts and for the treatment of the signal.
If `choice! = Nun`, one adds a button above allowing for choice of between several options to visualize.

Note: it would be advisable to add in this class of the common methods for the extraction of the functions of the inter-spectrum, and their posting towards the selected software of visualization. Currently, these features are written in the classes of corresponding IHM of the mitres.

4.5 ParamModel classesCouples, ParamModeLMME and ParamProjMesuModal

These classes allow to display in the principal window or in of TopLevel of the frame allowing to control the parameters of calculation for the use of CALC_MODES or PROJ_MESU_MODAL.

4.6 Class Window Observation

This class generates Frame to apply to a concept result of the mode_meca type the macro-order OBSERVATION. It is used in the mitre of identification of efforts and in the display window of the results.
This Frame is made up of two frameworks of colors green and blue, in which one displays the groups of nodes contained in the selected experimental model, and DDL available in each group. A button allowing for choice of the change of associated reference mark is used.

4.6.1 DispObs class

This class opens an external window TopLevel when one clicks on the button “Observations” in the mitre “Parameters and visualization”.
One notes that in the method set_resu, it is checked if the concept result chosen upstream were created by the macro-order OBSERVATION. If it is the case, then one preselects the selected DDL and the parameters of change of reference mark (see the method set_selected in the GroupNoWidget class below).

4.6.2 Classes SelectionNoeuds, SelectionMailles and _SelectionBase

The classes SelectionNoeuds and SelectionMailles display the lines of groups of nodes and meshs, and are subclasses of _SelectionBase. This class allows to establish the link between the DDL selected by the user and the construction of the keyword in the macro one OBSERVATION. They are based on the class below.

4.6.3 GroupNoWidget class

This class displays the lists of groups of nodes and meshs, with the associated DDL (method set_data).
get_selected is a method which turns over in the form of dictionary the options chosen by the user.
set_selected is used when the concept mode_meca to be visualized was already created by OBSERVATION. In this case, one will seek in the context the keywords which were used during the creation of the concept, and one modifies the boxes of the interface consequently. That must make it possible to modify the existing concept without having to reproduce all the operations.

Note: method set_selected is not optimal, and it would be advisable to modify it:

- one added at the end of the routine the order “return Nothing”, which generates an error; it was however noticed that if one turns over really nothing (order “return” only), then when the window is opened, the check boxs are not it,
- if one wants to modify a concept already created by OBSERVATION, it would be necessary to set out again of the initial result, and not of the concept already project, because on the basis of this last, one can only still withdraw points, but one cannot modify some.

5 Validation of the graphic tools and tree of call

In this section, one proposes, for each mitre:

- a procedure of recettage of the graphic tools,
- the description of the tree of call for the aforementioned procedures

The test of the graphic tools will have to be carried out on all the distributions on which one ensures the follow-up quality of Code_Aster (in particular Gauge). It will be checked that the connections towards the tools for visualization are correctly carried out:

- posting on XMGrace and GMSH,
- posting on Salomé: to open a transfer of Salomé, and to check in the mitre “Parameters of visualization” that the option “Salomé” is well selected.

5.1 Display window of the deformations and MAC

When one opens **CALC_ESSAI** , one arrives in first on a window allowing to parameterize the visualization of the computation results. Two options are possible:

- Gmsh/Xmgrace: the deformations are visualized in GMSH, the curves in Xmgrace. MAC are visualized in a specific Tk window,
- Salomé: all the computation results are displayed in Salomé; for MAC, one created a script Salome allowing to see MAC in the shape of a constant field by elements at the points of Gauss.

5.1.1 Test of visualization: CAS-test sdls112a

The validation of the features of this mitre is validated in section 5.3 , at the same time as the features of expansion of data.

5.1.2 Tree of calculation

Class InterfaceParametres (file ce_ihm_parametres.py) :

It is a subclass of the class Frame (Tk) in which is manufactured the IHM, called with the launching of CALC_ESSAI. It is made up of two principal frameworks:

- a framework allowing to define the choice of visualization (GMSH/Xmgrace or Salomé); to note that distant visualization on Salomé is not available; when a study Salomé is opened, its name appears in the window; to think of clicking on validating to choose the study in which the results will be displayed
- a framework allowing to visualize the modal deformations, criterion of MAC, or to simulate FRF “blow of hammer” by giving a modal base and a point of excitation (and to visualize them).

When a result is selected, the routine **check_state** check in particular if the models of results 1 and 2 are the same ones. If it is the case, the button of MAC is activated. The calculation of the matrix of MAC is carried out in a file of calculation: **ce_calcul_expansion.py** (routine **calc_mac_mode**).
The routine **view_frf** allows to visualize the FRF in a window annexes (class **DispFRFDialogist**, in **outils_ihm.py**), that one describes below.

When one chooses a set of software for visualization, one creates an authority associated with the choice carried out:

- class **CalcEssaiGmsh**, and CalcEssaiXmgrace for the posting of the curves,
- class **CalcEssaiSalome** ; and **CalcEssaiSalomeCourbes** for the posting of the curves.

The list of the studies open Salomé is brought up to date with the button “To bring up to date”, which returns towards the method **visu_study_list**. Method called (method of the class **CalcEssaiSalome**), uses a common script with **STANLEY** (**SalomeGetStudies.py**).

3 methods **visu_resu**, **visu_mac** and **visu_courbe** return respectively towards the methods of the two classes quoted above.

Classes CalcEssaiLogiciel (file ce_ihm_parametres.py) :

Class hat for the piloting of the external software. In particular allows to define the files where the results are printed, and to do them **IMPR_RESU** necessary.

Classes CalcEssaiGmsh (file ce_ihm_parametres.py) :
Class for the posting of the deformations in Gmsh. To note, for the posting of MAC: this one is done in a window Tk python, which is defined in **outils_ihm.py** : one opens a window annexes TopLevel, and one displays the frame **MacWindowFrame** inside.

Classes CalcEssaiSalome (file ce_ihm_parametres.py) :
Class for the posting of the deformations in Salomé. Routines **studylist** and **Show** return towards scripts Salomé defined for **STANLEY** , via the operator **EXEC_LOGICIEL** .
For the visualization of MAC, one manufactures in **make_mac_salome** (in **ce_calcul_expansion.py**) a square grid, and one assigns to each mesh a constant field by element, whose values are worth respectively:
- the value of MAC,
- the sequence number of the first list,
- the frequency of the first list,
- the sequence number of the second list,
- the frequency of the second list.

The routine Show displays the deformation, by using a script Salome called **salomeScriptMac** .

Notice important: a bug was announced in Salomé (card REX 19375), and the poster carried out is not correct. To correct this one, it is necessary to make a right click on the sight + edict + OK (without anything to change with the parameters).

Classes DispFRFDialogue (file outils_ihm.py) :
Class allowing to display a TopLevel window (news fenestrates depend on the principal Tk window), to display FRF resulting from a concept dyna_harmo, or to calculate a harmonic result starting from a base of modes and an excitation “hammer”.
- the whole of the concepts is initialized in the shape of a list with two components, for each of the two columns,
- when certain parameters are modified, all the IHM is updated; thus, if the selected concept is one mode_meca, one reveals Frame to choose the design assumptions of the associated harmonic result; if the concept is already a harmonic result, then one does not need to display this window
- some attributes of the class:
  - self.dyna : contains the dyna_harmo, to calculate, or already calculated; third is used by the mitre of structural modification, to calculate internal displacement given as starter (definite on a super-mesh self.sumail),
  - self.ddls : for each column, the list of the ddl associated with a kind of field (self.champ_choisi) ; NB: one checks for each field that the DDL exists, but it is not checked if the component is nonempty for the selected node; if the component does not exist, one returns an error message in the interactive window (at the time of RECU_FONCTION)
  - self.var_type_resu : for each of the two columns, allows to say if the selected result is one dyna_harmo or one mode_meca, and thus to display, or not, the window of calculation of a harmonic result,
  - self.param_calc : allows to gather in a dictionary the variables entered the IHM by the user for the calculation of a harmonic result: the node and direction of excitation, the waveband and the frequential resolution,
  - self.param_disp : for each of the two columns, parameters to display the FRF: node and direction of visualization, and the field to be visualized.
- calc_dyna_line_harm : method of calculating of the harmonic result, on physical basis: calculation is false for the experimental results (and does not have, in addition, interest) because the experimental physical matrices are not known
- choix_champ : look in .TACH structure of data if the fields corresponding to displacements, speeds and accelerations are not empty; if they are not it,
- choix_dd L: use CREA_CHAMP and EXTR_COMP to recover the field for the first sequence number to the format python; the components of this field available are stored in the .comp of the structure of data result,

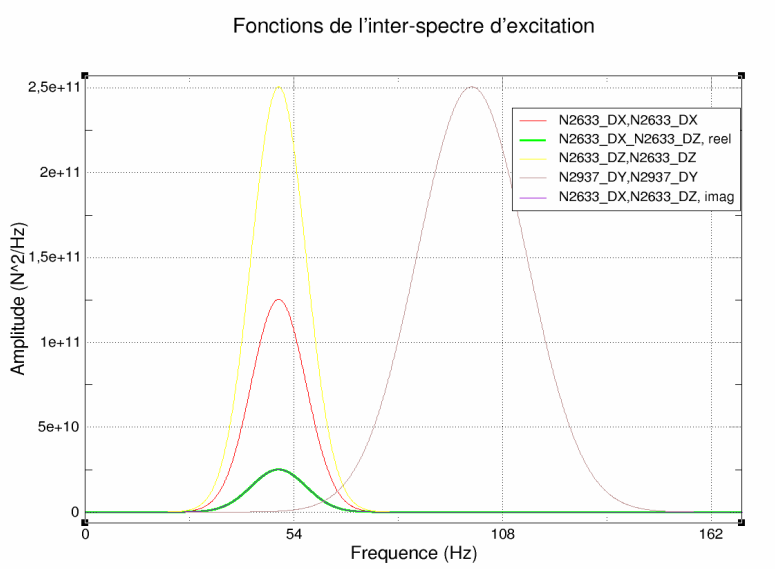
affich_FRF : recovery of the function, and recovery of the error message if the selected component and/or the node do not exist; posting is done via the class param_visu.

5.2 Identification of efforts

5.2.1 General presentation: CAS-test sdlv125a

The objective of this CAS-test is to simulate a measurement on an excited cylinder in three points. The effort is defined by a inter-spectrum of size 3, whose functions are the Gaussian ones (rustled pink). Efforts on the DDL N2633_DX and N2633_DZ are correlated, with a dephasing of $\pi/2$. The functions are represented on the graph below:

Figure 5.2.1-a : inter-spectrum of excitation.



Calculation having allowed the simulation of the data is carried out with DYNA_ALEA_MODAL, on modal basis.
A base of mode restricted with the whole of the sensors (the "DDL sensors") is calculated with the operator OBSERVATION, on which one restores the physical result with REST_SPEC_PHYS, to obtain a simulated measurement. The objective of the CAS-test is to identify the efforts applied starting from the data of the inter-spectrum SPECTPH1 and of the base of modes projected MODEIDE1. The identification is done on a reduced number of DDL chosen a priori and called "DDL orders".

5.2.2 Validation of the CAS-test in interactive

To launch the CAS-test in interactive, to modify the command file sdlv125a.comm while writing interactif=1 before the order CALC_ESSAI. The IHM of CALC_ESSAI launches out. To place in the mitre "identification of loading".
In the order BEGINNING, to remove the keyword CODE. In this manner, if the code plants on an Aster error, this one is caught up with in Exception, and the message is displayed in the IHM of CALC_ESSAI.

1) Choice of the base of modes:
   - to choose MODE_NUM : one uses in this base only the modal characteristics: modal masses, frequencies and depreciation

2) Base deformations observable:

- experimental model: MODELEXP : contains only 31 nodes,
- base modes: MODE_NUM : will be projected with OBSERVATION on MODELEXP,
- group of nodes GPCYLRED : to notch the DDL DX and DZ, to choose a change of reference mark 'NORMAL', with the option VECT_Y = (0.0, 1.0, 0.0),
- group of nodes GPSUPRED : to notch the DDL DY and to choose a change of reference mark 'CYLINDRICAL' with AXE_Z = (0.0, 1.0, 0.0),
- To click on "Validating".

3) Base commandability:
   - experimental model: MODELACT : contains only 3 nodes,
   - base modes: MODE_NUM : will be projected with OBSERVATION on MODELACT,
   - group of nodes N1 : to notch the DDL DX and DZ,
   - group of nodes N2 : to notch the DDL DY,
   - To click on "Validating".

4) Inter-spectrum under operation:
   - to choose SPECTPH1, and to leave Standard field to 'DEPL'.

5) Parameters of calculation:
   - Not to modify them (not regularization carried out),
   - To click on "Launching"

   To check, when one "forgets" one of the preceding operations, that a message appears in the window of messages. It is necessary to avoid the posting of the alignment python in the window comforts, which is not explicit for the user.
   Calculation takes less than one minute. Most of the time of calculation is used for the extraction of the inter-spectrum in the form of matrix python. Calculation in itself is almost instantaneous. When calculation is finished, one sees the whole of the operations carried out, the last being "Calculation of Syy_S: Modal synthesis of displacements".

6) Posting of curves: pace of the identified effort
   - In the left-hand column of visualization of the results, to choose option EFF Phy (for "physical efforts"), and to click on "Validating"; to select one or more functions of the inter-spectrum and to visualize the curves by modifying the scales of X-coordinates and ordinates (LIN/LOG),
   - To export the inter-spectrum: one can give him a title in the box located below the list of the functions. The name of the table is that pre-declared with the call of the macro-order ('EFFORTS').

7) Posting of the curves: comparisons measured/synthesized displacements:
   - In the left-hand column, to choose Depl Phy (for measured physical displacements), and in that of right-hand side, Depl synt (for synthesized displacements), and to click on "Validating"
   - To select in each column one or more functions to be compared, and click on "Displaying curve"
   - To export the two inter-spectra. The names are those pre-declared with the call of the macro-order: 'DEPL_PHY', and 'DEPL_SYN'. The creation of these inter-sectres with the format Aster is relatively long.

5.2.3 Tree of calculation

One mentions here the principal routines used for the realization of preceding calculation.

Class InterfaceIdentification (file ce_ihm_identification.py) :
It is a subclass of the class Frame (Tk) in which is manufactured the IHM of identification, called with the launching of CALC_ESSAI. Remarks on the behavior of this class:

- all the calculated inter-spectra are attributes of this class (inititialisation with a worthless size, the size is given by the class of calculation):
  - coil. Syy : measurement
  - coil. Syy_R : measurement recomputed starting from modal displacements,
  - coil. Sqq : modal displacements,
  - self.SQQ : modal efforts,
  - self.SQQ_R : recomputed modal efforts starting from the physical efforts,

- coil. Sff : physical efforts of dimension nb_act (many actuators)
- coil. Syy_S : measurement Re-synthesized starting from the physical efforts

- Method setup : all the graphic classes contain a method setup, called for cooling when mitre is changed, or when a series of calculations is finished; this one updates the menus unrolling with the new concepts aster created (for example, a base of modes created by expansion in the mitre corresponding).
- Method _create_opt_data : defines certain carateristic inter-spectra calculated (like the variables of access, nume_ddl or nume_mode who are associated for them), and the functions used to extract the functions from them,
- MethodS _definit_observability and _definit_commandabilite : definition of the interfaces allowing of paramétriser the operator OBSERVATION who creates the concepts self.obs_co and self.com_co.
  - these concepts are calculated while clicking on “Validating” (calculate_xxxx)
  - the windows of color are widgets of the class SelectionNoeud (see below): this class allows, inter alia, to recover the DDL associated with each node with the model, and the groups of node of the grid
- Methods _calculate_observabilite and _calculate_commandabilite : launching of OBSERVATION : the keywords factors for the changes of reference mark and the filtering of the DDL are factories in get_filtres starting from the result of the routine get_selected, described below.
- Method calculations : successive launchings of calculations:
  - calcturb is an authority of the class CalcEssaiIdentification ,
  - calculations fact links between the concepts of the graphic class and the concepts equivalent of the class of calculation,
  - after calculation, recovers the new calculated concepts and associates them with attributes of the graphic class (example: coil. Syy = self.calcturb.Syy) ; the existence of the new concepts is given by a Boolean variable (example: self.calcturb.is_Syy = 1 , variable put at 0 at the beginning of calculation)
- Methods for posting of the curves:
  - One chooses the type of curve to be displayed; example, one chooses 'EFF Phy' to display the physical efforts
  - one clicks on “Validating”: the routine get_list returns the list of the functions of the corresponding inter-spectrum; example: [['N1_DX', 'N1_DX'], ['N1_DX', 'N2_DX'], ['N2_DX', 'N2_DX']] , lists manufactured by the functions calcul_xxx_xxx , while basing itself on classification associated with the inter-spectrum (physical or generalized),
  - plot_curve : posting of the curves; use the function _get_graphe_data , which recovers the X-coordinates and ordinates of the function in the inter-spectral matrix, and method visu_courbe (InterfaceParametres class) for the sending of posting by Xmgrace or Salomé.

Classes SelectionNoeud and SelectionMaille (file cata_ce.py) :
Allows to create a widget displaying the groups of nodes associated with a grid, and the DDL carried by the nodes of this group (widgets green and purple in the interface). Use:

- choice of an experimental model: call to _observability_changed ,
- recovery of the model associated with the name (get_modele),
- method set_resultat in SelectionNoeud: call to set_modele_maillage ,
- method set_modele_maillage : recovery of the PRNM of the model which carries the DDL carried by each node, and of the groups of nodes,
- method set_modele_maillage : for each group, recovery of the “true components” associated with the PRNM (see wiki Aster for the description of this structure of data) with the function find_composantes .
- Method get_selected : turn over the data of changment of reference mark and filtering of the DDL for the use of OBSERVATION .

CalcEssaiIdentification classes and CalculInverse (file ce_calcul_identification.py) :
Classes of calculation to carry out calculation reverses identification of efforts.

Note:

- in the other mitres, most calculations are realized by orders Aster. Here, once the data input are extracted, one uses only functions python (operations on the matrices, decomposition in singular values…),
- the name of variable xxxm1 indicates the reverse or theopposite one (by SVD) of xxx.

Description of the principal methods:

- Method `calculate_force` : principal method of the class managing the calculation of the efforts;

  - errors in the class `CalculInverse` are recovered and a generic error message is displayed; in each elementary method of calculating, a more precise message (specifying dimensions of the matrices concerned) is given,
  - each result is an authority of the class `InterSpectre` (see `cata_ce.py`) ; one can associate to him a classification of DDL physical or generalized, according to the type of calculated inter-spectrum; for that, one associates the inter-spectrum with an existing base; nume_phy returns a list of characters of the form `['N1_DX', 'N1_DY', 'N2_DX'…]` , and nume_ddl_gene returns a list of the form `['MO1', 'MO2', 'MO4'…]` (classification based on the variable of access `NUME_MODE`).
- Method `Calc_Z` : calculation of the matrix of impédence $Z = diag(-\omega^2 + \omega_j^2 + 2\xi\omega\omega_j)$ and of its reverse $Zm1$; according to the type of data (accelerations, speeds or displacements, one can have to integrate while dividing by $\omega^{\exp}$, where exp is worth 2.1 or 0 respectively.
- methods `Calc_SQQ` , `calc_Sff` : the pseudo-opposite of the matrices calculate respectively $C\Phi$ and $\Phi^T B$ (see U4.90.01, section 6.2.1 for the significance of these matrices); too low singular values (below a criterion ϵ chosen by the user) are put at zero; after regularization of Tikhonov, the reverse of the singular value s_j is worth $\frac{s_j}{s_j^2+\alpha^2}$, the parameter α being able to be different according to the frequency where one is (adjusted by the power m) : thus, the regularization is weak when one is before or close to the frequency of the mode, and it becomes stronger when one moves away from there. The matrices thus calculated are put in memory in authorities of the class `InterSpectre` .
- Method `verif_Syy` : reclacule physical displacements `Syy_R` starting from modal displacements to compare them with `Syy` (to estimate the quality of the estimate reverses),
- Method `verif_SQQ` : recompute the modal efforts `SQQ_R` from `Sff` to compare them with `SQQ` ,
- Method `synthes_Syy` : recompute displacements `Syy_S` starting from the identified efforts `Sff` to consider the quality general of the inversion,
- Method `choix_alpha` : the parameter α regularization of Tikhonov can be modified according to the frequency to which one is compared to the Eigen frequencies of the base.

5.3 Expansion of data

5.3.1 General presentation of the CAS-test sdls112a

The objective of this CAS-test, based on an international benchmark (benchmark of the gartor), aims to extend the FRF and experimental modes identified on the model, i.e. to find the best combination linear of a base of modes (dynamic, static) sticking to the experiment. For more precise details, to see U4.90.1 (Doc. of `CALC_ESSAI`), section 4.2.1, and V3.03.112 (reference material of the CAS-test).

Note:

In not-interactive, the order `CALC_ESSAI` is completely useless here, it is enough to use the macro-order `MACRO_EXPANS`, or to connect the orders `PROJ_MESU_MODAL` and

REST_GENE_PHYS. For the validation in interactive of CALC_ESSAI, one proposes to reproduce the expansions carried out in this CAS-test.

5.3.2 Validation of the CAS-test in interactive

To launch the CAS-test in interactive, to modify the command file sdls112a.comm while writing interactif=1 before the order CALC_ESSAI. In the order BEGINNING, to remove the keyword CODE. In this manner, if the code plants on an Aster error, this one is caught up with in Exception, and the message is displayed in the IHM of CALC_ESSAI.
To increase the memory size used to 1024 Mo. To gain in speed of calculation, one can also erase or comment on the following operations: TEST_RESU, TEST_TABLE, MACRO_EXPANS and orders MAC_MODES, that one will carry out in interactive.
The IHM of CALC_ESSAI launches out. To place in the mitre “identification of loading”.

1) First test: expansion of a harmonic result on a basis of modes:
   - choice of the base of modes of expansion: to choose MODES and to select only the modes whose frequency lies between 1 and 100 Hz,
   - base FRF to be extended: to choose the concept DYNA ; not to select sequence numbers (it is not possible with MACRO_EXPANS , which uses EXTR_MODE , to extract part of the sequence numbers from a concept dyna_harmo .
   - to choose the name of the concept result: this name must have less than 5 letters; concepts XXX_ET (wide result), XXX_NX (expansion reduced to the selected sequence numbers bases) and XXX_RD (reduced wide result) will be created,
   - comparison of the results extended to the initial harmonic concept: to click on box FRF in the mitre “Parameters and visualization” of the IHM:
     - to choose the two concepts results to compare: on a side DYNA and other XXX_RD ,
     - to choose the field to be compared ('ACCE'),
     - to choose the node and the component: for DYNA , to choose the node 'N1011' , component 'D3' , and for XXX_RD , to choose the same node with the component 'DZ' ,
     - to click on “Displaying”: to check that the displayed curves are about confused.

2) Test 2 : simulation of a FRF starting from a base of modes: the objective is to compare a base of experimental FRF with the base of FRF simulated numerically on an experiment “blow of hammer”:
   - to click on box FRF in the mitre “Parameters and visualization”,
   - to choose in one of the columns the concept MODES ,
   - to choose a node of excitation (example, 'N1' , excitation 'FX'), a waveband of calculation (1.0, 50.0) and a frequential resolution (example: 1.0 Hz)
   - to click on “Calculating”
   - to display a FRF (example, according to the node ‘N1, comopsante ‘DX’); to check that the FRF is displayed correctly.

3) Test 3: expansion of a base of experimental modes on a digital basis:
   - to start again calculation (to avoid encumbering the memory unnecessarily),
   - in the mitre expansion of data, to choose the base of expansion : MODES ,
   - to choose the base of experimental modes to extend: MODMES ; the frequencies are the same ones, because the CAS-test is commonplace: the modes to be extended result from projection on the experimental model of the digital modes,
   - To choose a list of modes in the two columns,
   - To choose a name of concept result (less than 5 letters) and to click on “Calculating”
   - To compare the deformations results in the mitre “Parameters and visualization”: to compare, for example, experimental modes (MODMES) and reduced extend modes (XXX_RD), by choosing them in the small double in bottom of the mitre, and clicking on “Deformations”; to use the modes of visualization in GMSH and Salomé,
   - To display MAC result: to choose two deformations defined on the same model, and to click on the MAC button; to choose by exmple:
     - extend modes (XXX_ET) and expansion bases it,

- reduced extend modes (XXX_RD) and the experimental base (MODMES).

5.3.3 Tree of calculation

Class InterfaceCorrelation (file ce_ihm_expansion.py) :
It is a subclass of the class Frame (Tk) in which is manufactured the IHM of expansion, called with the launching of CALC_ESSAI. Remarks on the attributes created with initialization:

- self.calculs : it is an authority of the class CalcEssaiExpansion, which carries out all calculations Aster inherent in the expansion of data (launching of MACRO_EXPANS mainly, and of MAC_MODE for postprocessing),
- self.resu_num and self.resu_exp : indicate respectively the base of expansion (a concept mode_meca) and the experimental result (one mode_meca or one dyna_harmo) ; with these variables are associated StringVar self.var_resu_num and self.var_resu_exp, which indicates the G-string containing the concept selected by the user,
- self.var_resu1 and self.var_resu2 : StringVar containing the names of the concepts selected by the user in the menu of postprocessing (in bottom of the IHM)
- Method setup : all the graphic classes contain a method setup, called for cooling when mitre is changed, or when a series of calculations is finished; this one updates the menus unrolling with the new concepts aster created (for example, a base of modes created by expansion in the mitre corresponding),
- Method prepare_calculs: allows to launch the calculation of expansion itself:
  - checking that the data necessary to calculation were indeed selected by the utilsator,
  - recovery of the list of the indices of the lines selected in the list of the modes (one uses for that the method selection class ModeList),
  - assignment of the attributes of the class of calculation by calling the method self.calculs.setup,
  - launching of calculations: self.calculs.calc_proj_resu.

CalcEssaiExpansion class (file ce_calcul_expansion.py) :
Launching of calculations of expansion, mainly macro-order MACRO_EXPANS. One details the operation of the principal method here, namely calc_proj_resu :

- one creates two to four concepts results:
  - XXX_ET is the principal result of the expansion: the base of modes or the wide harmonic result
  - XXX_RD is the reprojection of the result extended on the experimental model,
  - if one selects whole or part of the base of digital modes, one creates a called extracted base XXX_NX (condition " yew self.mode_num_list "),
  - if one selects whole or part of the base of experimental modes, one creates a called extracted base XXX_EX (condition " yew self.mode_exp_list "),
- XXX is the name of the concept chosen, it does not have to exceed 5 characters,
- after calculation, the whole of the concepts created is added to the list of the existing objects Aster (self.ce_objects , or mdo) with the order update; in parallel the MyMenu menus themselves are brought up to date in the class Interfaces Correlation.

5.4 Structural modification

5.4.1 General presentation of the CAS-test sdll137

The purpose of this case test is to validate the procedure and the calculation of structural modification starting from measured information. One proposes to calculate the variations of the first two Eigen frequencies of a embed-free beam, following the modification made on a portion of the beam.
Modelings A and D of the case test detail the course of the procedure of calculation while making use only of the orders standards of Aster (without placing by the order CALC_ESSAI).
Modelings B and C utilize the order CALC_ESSAI.

5.4.2 Validation of the CAS-test in interactive

To launch the CAS-tests sdll137b and sdll137c in interactive, to rock the variable `interactive` with a nonworthless value (1 for example). This variable is localised before the call of the order `CALC_ESSAI`. In the order `BEGINNING`, to remove the keyword `CODE`. In this manner, if the code plants on an Aster error, this one is caught up with in Exception, and the message is displayed in the IHM of `CALC_ESSAI`.
The IHM of `CALC_ESSAI` launches out. To place in the mitre “structural Modification”.

1) First test: structural modification by using the method ES for the choice of the base of expansion (sdll137b.comm)
   - In the panel “Choice of the base of expansion”, to choose the following data input:
     Experimental modes: `MODEIDE`
     Model support: `MODLSUP`
     Matrix stiffness (support): `KASSUP`
     Method (expansion bases): `ES`
     Model modification: `MODLCPL`
     For the choice of the nodes and ddl sensor, to select the ddl `DY` and `DZ` group of nodes `SENSOR`
     For the choice of the nodes and ddl interface, to select all ddl of the group of nodes `EXTERNAL` (the use of the elevator is perhaps necessary in order to reach this group of nodes)
     Name of the super-mesh: `SUMAIL`

     One can then click on **To validate** for the validation of the seized data. The list of the identified clean modes and lists it vectors available for the base of expansion are then displayed in the two left-hands column. One selects, with the mouse, all the vectors of the list of the “Modes of the experimental model” and the eight vectors of the “Base of expansion”.

   - In the panel “Coupling modification/condensed model”:
     One keeps the parameters by default for `PROJ_MESU_MODAL` and modal calculation.
     One clicks then on **To calculate** in order to obtain the Eigen frequencies of the modified structure which are displayed in the column Frequencies modified structure.
     To check that the first two calculated Eigen frequencies are close to: 7.79 Hz and 32.84 Hz
     One can then check the quality of the base of expansion. One chooses MAC for the heading criterion. While clicking on **To validate**, a matrix having the results of the criterion is displayed. The base of expansion is supposed to be correct if the diagonal terms of this matrix are close to the unit.

   - To visualize the effect of the modification on the deformations, one clicks on **Deformations** in the panel “Comparison initial structure/modified structure”.
     One then sees appearing a window GMSH or SALOMÉ which makes it possible to compare the initial modal deformations and after modification.

2) Second test: simulation of a structural modification by using method LMME for the choice of the base of expansion (sdll137c.comm):

   - In the panel “Choice of the base of expansion”, one chooses the same data input as for the first test, except the choice of the method for calculation of the base of expansion and the name of the model modification.
     Method (expansion bases): `LMME`
     Model modification: `MODLX`

     While clicking on **To validate**, the list of the identified clean modes and lists it vectors available for the base of expansion are displayed in the two left-hands column. One selects, with the mouse, all the vectors of the list of the “Modes of the experimental model” and them the first eight modes “Base of expansion”.

- In the panel “Coupling modification/condensed model”, one carries out the same procedure as for the first test.
  For the checking of the quality of the base of expansion, one will use criterion IERI (instead of the criterion of MAC). A matrix of weighting is necessary for this criterion. One chooses to balance with the matrix of rigidity while clicking on **Rigidity**. While clicking on **To validate**, a matrix having the results of the criterion is displayed. With criterion IERI, the base of expansion is supposed to be correct if the diagonal terms of this matrix are close to zero.

5.4.3 Tree of calculation

The two most important files python for the calculation of structural modification are: ce_ihm_modifstruct.py and ce_calcul_modifstruct.py.
The module ce_ihm_modifstruct manage the classes relating to the graphic interface part and the module ce_calcul_modifstruct manage the classes which launch the procedure of calculation of structural modification by calling the operators aster.

InterfaceModifStruct class
This class is a class of the module ce_ihm_modifstruct. It is the class hat for the mitre “structural Modification”. It has several attributes of which expansion and coupling.
self.expansion is an authority of the class InterfaceExpansion. This class allows to seize and display the data necessary for the choice of the base of expansion necessary for obtaining of the field of displacement to the interface.
self.couplage is an authority of the class InterfaceCouplage. This class allows to seize and display the various parameters of calculation for the estimate of the clean modes of the modified structure. This modified structure is resulting from the coupling between the model of the condensed initial structure and the model of the modification.

CalcEssaiModifStruct class
This class is a class of the module ce_calcul_modifstruct. It allows to connect the operators aster necessary for the calculation of the clean modes of the modified structure. It understands several methods of which most important are:
- calc_base_es : calculate the base of expansion by using the method ES (static expansion). This method launches the operator MODE_STATIQUE. The basic vectors obtained are static deformations.
- calc_base_lmme : calculate the base of expansion by using method LMME (Local Model Modeshapes Expansion). This method carries out a modal calculation with the operator CALC_MODES with the option PLUS_PETITE. The basic vectors obtained are clean modes.
- condensation : carry out the condensation of the measured model and creates the super-mesh representing the initial structure. This method connects the operators PROJ_MESU_MODAL, MACR_ELEM_STAT and DEFI_MAILLAGE.
- modes_modele_couple : calculate the clean modes of the system couples (initial structure + modification) with the parameters of calculation provided by the user. This method carries out also the retro-projection of the results on the grid measures by using the operator DEPL_INTERNE.
- indicateur_choix_base_expansion : compare the field of displacement with the interface obtained with the coupled model and those obtained by static expansion. One raises the field of displacement of the structure modified at the point sensor, then one carries out a static expansion in order to obtain the field with the interface. This task is carried out with the assistance the operators PROJ_MESU_MODAL, REST_GENE_PHYS, PROJ_CHAMP and MAC_MODES.

5.5 Calculations of inter spectra, car spectra and transfer transfer functions

5.5.1 General presentation of the CAS-test zzzz241a

The objective of this CAS-test is to validate the basic functions of treatment of the signal integrated in the operator CALC_SPEC. One simulates the temporal response of an oscillator to 4 degrees of

freedom, and one calculates various transfer transfer functions. For more precise details, to see U4.32.21 documentation (Doc. of CALC_SPEC)

Note:

In not-interactive, the order CALC_ESSAI is completely useless here, it is enough to use the order CALC_SPEC. For the validation in interactive of CALC_ESSAI, one proposes to reproduce the expansions carried out in this CAS-test.

5.5.2 Validation of the CAS-test in interactive

To launch the CAS-test in interactive, to modify the command file zzzz241a.comm by replacing all the lines located under the order tab_rep=CRÉA_TABLE (...) (lines 196 to 307) by
TEST_CE=CALC_ESSAI (INTERACTIF=' OUI',),
and not to use the file of zzzz241a.com1 continuation.
In the order BEGINNING, to remove the keyword CODE. In this manner, if the code plants on an Aster error, this one is caught up with in Exception, and the message is displayed in the IHM of CALC_ESSAI.
The IHM of CALC_ESSAI launches out. To place in the mitre “Treatment of the signal”.

1) First test: Calculation of the H1 estimators and H2 of the transfer transfer functions, with measurement 1 in reference: On the basis of the initial window of CALC_ESSAI, it is necessary:
   - In the list “Tables available”, to select “tab_rep” to put the name in intensified brightness,
   - To click on the button “=>” to fill out the lists entitled “Points of measurements” and “Points of reference”. One must obtain 5 points of measurements, numbered from 1 to 5, and 5 points of reference, also numbered from 1 to 5, corresponding to same measurements.
   - To choose measurements associated with items 2 to 5 in the list “Points with measurements”, and measurement associated with item 1 like reference,
   - To choose the window “Hanning”,
   - To inform a length from “12”, and to click over “Duration”,
   - To inform a covering from “50”, and to click on “For hundred”,
   - To select “H1” under the button “Transfer”, and to click on “Transfer” to calculate the H1 estimators.
   - To display the results by selecting in the list “Transfer (H1)”. To choose the format of the data to be displayed, as well as the nature of the axes, and to click on “Visualizing”.
   - To select “H2” under the button “Transfer”, and to click on “Transfer” to calculate the estimators H2.
   - To display the results by selecting in the list “Transfer (H2)”. To choose the format of the data to be displayed, as well as the nature of the axes, and to click on “Visualizing”.

2) Second test: Calculation of inter spectra and car spectra: While setting out again of the initial window of CALC_ESSAI , it is necessary:
   - In the list “Tables available”, to select “tab_rep” to put the name in intensified brightness,
   - To click on the button “=>” to fill out the lists entitled “Points of measurements” and “Points of reference”. One must obtain 5 points of measurements, numbered from 1 to 5, and 5 points of reference, also numbered from 1 to 5, corresponding to same measurements.
   - To choose measurements associated with items 1 to 5 in the list “Points with measurements”. The selection or not of measurements of reference does not change the result, this selection, for the calculation of the spectra, not being taken into account.
   - To choose the window “Hanning”,
   - To inform a length from “12”, and to click on “Points”,
   - To inform a covering from “50”, and to click on “For hundred”,
   - To click on “Interspectres” to calculate them inter spectra.
   - To display the results while selecting in the list “Interspectres”. To choose the format of the data to be displayed, as well as the nature of the axes, and to click on “Visualizing”.

5.5.3 Tree of calculation

The mitre “Treatment of the signal” calls on a single class.

Class InterfaceCalcSpec (file ce_calc_spec.py) :

Class allowing to manage the mitre “Treatment of the signal”. One there definite at the same time graphic objects and calls to the operator CALC_SPEC, which carries out calculations. The principal methods are:

- InterfaceCalcSpec : method for the generation of the graphic interface (lists, frameworks, buttons, etc)
- frame_visu : method for cooling/initialization of the visualization part of the mitre
- visu_tempo : update of the list of visualization by adding the imported functions of the tables.
- display_mes : generation of the curves before posting
- calc_intespec : call to CALC_SPEC for the calculation of inter spectra
- calc_coherence : call to CALC_SPEC for the calculation of the functions of coherences
- calc_transfert : call to CALC_SPEC for the calculation of the transfer transfer functions
- crea_tab_fonc: creation of a concept TABLE_FONCTION starting from a selection of functions