Data-processing description of CALC_ESSAI

1 Introduction

macro-command CALC_ESSAI gathers a set of functionalities in correlation CALCUL-test, controllable by the means of a graphic interface in Tk python:

- expansion of experimental data on a basis of numerical deformed shapes (via macro-command MACRO_EXPANS),
- structural modification a structure known by a base of experimental modes and a digital model simplified,
- identification of forces on a structure from the data of the inter-spectrum and a base of eigen modes describing its behavior,
- computation of filtered and realised spectrums (via macro-command CALC_SPEC),
- visualization of modal deformed shapes, curves and matrixes of MAC in Salome or tools associated with Aster (GMSH, Xmgrace, Tk).

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

2 Launching, interactive mode and noninteractive

macro-command `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 Recovery

2.1 of the concepts pre-declared For

the mitres of identification of the forces and of structural modification, it is necessary to declare with the call of the macro-command the name of the concepts in output of this one, by the use of key word RESU `_XXX = CO ("NAME").` The names of the pre-declared concepts are transported in all the classes of execution of computation of the macro-command in the shape of a list (RESU_MODIFSTRU) or 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 can be used in the continuation of computation only in the frame of a poursuite. For
the mitre of processing of the signal, the leaving name of the concepts is fixed into tough (see section 5.5 19Launching

2.2 in noninteractive mode It

is not very relevant of launching the macro-command in NON-interactive mode, except, possibly, in the case of the identification of forces, mode which cannot be replaced by the use of another command. Thus, for the expansion of models, it is to better use macro-command MACRO _EXPANS `directly. For` the structural modification, one can refer to the sequence of commands describes in U2.07.03 documentation. For the processing of the signal, one can use macro-command CALC `_SPEC (see` U4.32.01).
The non-interactive mode is thus used for the identification of forces, and, generally, for the validation of the benchmarks associated with CALC `_ESSAI. In`
this mode, one returns directly to the CalcEssaiTest routine. Launching

2.3 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) 6 the window of message in foot, one imports
- all the graphic classes, called InterfaceXxxxxx, one creates
- instances of each graphic class, which one displays in the window tabs (instance of TabbedWindow) higher definite, one creates
- a file of messages which will be displayed in the window mess defined in foot of instance the tabs. File

3 cata_ce.py: catalogs the file

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

contains: the CalcEssaiObjects class

- : classify containing all the Aster concepts which could be useful in computations, the class Results
- : the Aster concepts sd_resultat are instances, the ModeMeca class
- : subclass of the preceding one: instancie the Aster concepts mode_meca the DynaHarmo class

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

the classes most used in computations are described here. Description

3.1 of the class CalcEssaiObjects This class

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

3.1.1 This method

recovers all the concepts available and the arranges 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 mode_meca_X object will be `instance` of this concept. The classes ModeMeca and InterSpectre are used in the current operations. The class Interspectrum has in particular a very useful method and to transform the concept inter-spectrum Aster into a matrix python indexed by the frequencies of discretization sequence numbers I and J (or, with core, names of nodes and of the components I and J). Several methods
relating to the classes in question are called to bind concepts to the nume_ddl, the models, the matrixes, the ascending meshes… update Makes it possible

3.1.2 to add

a concept to instance of MeideeObjects without having to entirely recreate it (what can be long being given the number of initializations). Debug Routine

3.1.3 of display

of the concepts associated with a concept. Routines get_xxx

3.1.4 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 from its name. Description of

3.2 the class ModeMeca the notion of MOD

_meca gathers indifferently the dynamic modes, resulting from MODE_ITER_SIMULT `, and the modes` static. The first are indexed by their sequence number and their eigenfrequency, the seconds by their sequence number and the variable NOM_CMP, which is `the degree of freedom` associated with the static deformed shape. Classes get_xxx

3.2.1 These classes are used

to bind the concept mode_meca with the ascending concepts: get_matrices recovers

- the mass matrixes, of stiffness and depreciation associated with the concept, recovered in the .REFD. The associated attributes are .mass, .kass and .cass get_nume recovers
- the nume_ddl by same skew (.numeric extension), get_maillages
- recovers the associated mesh, either by the means of the nume_ddl, or directly in the contained information in the .REFE of each sequence number of result. get_modele recovers
- the model associated, either by the routine dismoi of Aster, or via the associated mesh Attention: certain

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

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

3.2.2 Routine allowing

to extract the modal deformed shapes in the form from a matrix numpy. The extraction is done with the operator CREA_CHAMP and routine EXTR_COMP. The encapsulation of these two commands 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 means of operator OBSERVATION to create the base of the modes, or by means of the filter of DDL in the graphic interface. Classify DynaHarmo

3.3 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 computation which carries out only Aster computation. It is thus not necessary to extract the data with the format python from them. CaraElem classes,

3.4 ChampMateriau No python

processing of these concept was carried out. These classes are thus almost empty. They just make it possible to encapsulate the Aster concept corresponding and to be able to point on this one with its name. Classify Interspectrum

3.5 Classifies allowing

to treat the concepts of type inter-spectrum. The inter-spectrums are arrays containing of the subscripted functions of frequential variable by: sequence numbers

- (indices NUME_ORDRE_I and NUME_ORDRE_J) , or nodes and
- components (NOEUD_I, NOEUD_J, NOM_CMP_I , NOM_CMP_J). With the initialization

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

3.5.1 _spec With the image of operator

LIRE_INTE_SPEC, this routine an inter-spectrum manufactures Aster from functions defined in the form of list. If the concept
is connected to a model (with set_model), through a result concept (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. Routine set_model

3.5.2 Makes it possible to associate

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

3.5.3 _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 the nodes and components 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
- makes it possible to recover the sequence numbers in a list comprising only one occurrence of each one; it can be useful to dimension the nonsquare inter-spectrums, for example, functions FRF and of coherences created in CALC_SPEC; this functionality is however not used currently by the mitre of processing 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 matrixes; one thus adds temporarily a checking on the equality between many lines and of columns, association to
- a result concept and extraction of Active dof. (self.nume_phy), checking of
- coherence between the size of the nume_phy and cuts it inter-spectrum, fabrication of a list
- of couple of string for indicer in python the functions of the inter-spectrum: NOEUDS_CMP_X or NUME_ORDRE_X , according to the way in which the inter-spectrum in Aster (coupl_ddl) was manufactured, 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 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 component associated, one checks coherence with the self.nume_phy; if it 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. Classify Tempo the purpose

3.6 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 class Interspectrum. 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. Method extr_tempo

3.6.1 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 key words NUME_ORDRE_I, NUME_MES and while returning names FONCTION associated. Classify Model This class

3.7 allows to encapsulate

the models Aster. The associated methods make it possible to extract the ascending mesh and all from the downward nume_ddl which depend on this model (it can have several of them there. This class has little interest in itself. Description of the generic

4 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 computation: in a specific file python: for example, the file ce_ihm_expansion.py contains the graphic class related to the mitre of expansion of data, whereas the file ce_calcul_expansion.py contains associated computations (and can be tested in noninteractive mode without calling on the Tk moduli 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 nodes group in a mesh, choice of the parameters for an operation Aster, parameters for a change of reference. Classify TabbedWindow With the call of

4.1 macro-command

CALC_ESSAI in interactive mode, the routine create_interactive_window launches : a virgin window TabbedWindow

- (in outils.py) in which are displayed the specific contents of each mitre of work., a window of MessageBoxInteractif message
- , in which will be displayed the interactive message displayed in the course of computation. Note: Interior of window

virgin

1. 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), the routine set_current_tab is used
2. to display the interface corresponding to each change of mitre; the method setup associated with the graphic class of the mitre is called to refresh the environment (Aster concepts create); the method setup must thus exist **in the graphic classes of all the mitres, the initial size of the windows principal**
3. and of message respectively in the routines __init of definition of the aforesaid windows are given . Classify MessageBoxInteractif Fenestrates

4.2 vacuum in which the messages

of information are printed during computations. As specified in the preceding section, the window is created with the call of the macro CALC_ESSAI (instance 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 GUI, which contains TabbedWindow, objects: instance of CalcEssaiObjects
- which contains the existing Aster concepts, macro: macro-command, obsolete mess
- : instance of the window of messages
- param_visu: instance of the class
- GUI InterfaceParametres , which allows to choose the parameters of visualization of the deformed shapes and the curves (Xmgrace + GMSH or Salome). Display of the error messages resulting

4.2.1 from exceptions When one launches an Aster command

in the macro one through an interactive command, and that computation 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 command: try: MACRO_EXPANS (MODELE_CALCUL =
mcfact

```
    _calcul, MODELE_MESURE = mcfact_mesure, NUMERICAL
                  _DDL = numerical, RESU_ET = res_ET,
                  RESU_RD = res_RD,
                  RESOLUTION = parameters
                  , ** arguments) except
                  aster.error, err: message
                  = "ERREUR
ASTER: " + mess.GetText
     ("I", err.id_message, err.valk, err.vali, err.valr) self.mess.disp_mess
(message) UTMESS
                                       ("A", "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. The display in interactive window was removed, and replaced by a simple "Error in Aster computation". Classify generic ScrollList Frame allowing to display a table

4.3 with bar of

run. The attribute been worth is a list filled out by set_values corresponding 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"]] ModeList Classes and ModeFreqList Subclass

`of ScrollList, it`

4.3.1 has two buttons below

making it possible at a stretch to select or die-select all the lines of the list. It is used to display sequence numbers and an associated value (standard eigenfrequency or NOEU_CMP). get_selection turns 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 eigenfrequencies or the NOEU_CMP. Classify StudyList Subclass of ScrollList, it allows `to display`

4.3.2 the list of the studies

open Salomé. It has a method to validate which stores the study current Salomé for the display of `the deformed shapes` , MAC and curves. Classify VisuSpectre This class creates a frame made up of one or more

4.4 tables

for the display of the coordinates of the inter-spectrum to visualize. This class is used for the mitre of identification of forces and for the processing 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 display towards the selected software of visualization. Currently, these features are written in the corresponding classes of GUI of the mitres. Classes ParamModeIterSimult, ParamModeIterInv and ParamProjMesuModal

4.5 These classes allow to display in the principal window or

in of TopLevel of the frame allowing to control the parameters of computation for the use of MODE_ITER_SIMULT, MODE_ITER_INV or PROJ_MESU_MODAL. Note: by preoccupation with a simplification , it would be useful to remove
the possibility of using MODE_ITER_INV. Classify Observation Window This class generates Frame to apply

4.6 to a result concept

of the mode_meca type macro-command OBSERVATION. It is used in the mitre of identification of forces and in the display window of the results. This Frame is made up of two frames of colors green and blue
, in which one the model displays the nodes groups contained in experimental selected, and the DDL available in each group. A button allowing for choice of the change of associated reference is used. Classify DispObs This class opens an external window TopLevel

4.6.1 when one clicks

on the button “Observations” in the mitre “Parameters and visualization”. It is noted that in the method set_resu, one checks if the result concept
chosen upstream were created by macro-command OBSERVATION. If it is the case, then one and the preselects the DDL selected parameters of change of reference (see the method set_selected in the GroupNoWidget class below). Classes SelectionNoeuds , SelectionMailles and _SelectionBase

4.6.2 the classes SelectionNoeuds and SelectionMailles display the lines

of nodes groups and meshes, and are subclasses of _SelectionBase. This class allows to establish the link between the DDL selected by the user and the construction of the key word in the macro OBSERVATION. They are based on the class below. Classify GroupNoWidget This class displays the lists of nodes groups

4.6.3 and meshes

, with the DDL associated (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 key words 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: the method set_selected is not optimal, and it

would be advisable to modify it : one added at the end of the routine the command “return Nothing”,

- which generates an error; it was however noticed that if one turns over really nothing (command “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 result initial, 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. Validation of the graphic tools and shaft of call In this section

5 , one proposes, for each mitre: a procedure

of recettage of the graphic tools, the description

- of the shaft 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: display on XMGrace and GMSH, display on Salome: to open

- a transfer of Salomé, and to check
- in the mitre "Parameters of visualization" that the option "Salome" is well selected. Display window of the deformed shapes and MAC When CALC_ESSAI

5.1 is opened, one arrives in first on a window

making it possible to parameterize the visualization of the computation results. Two options are possible: Gmsh/Xmgrace: the deformed shapes 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 with Gauss points. Test of visualization: benchmark sdls112a the validation of

5.1.1 the features of this mitre is validated in

section 5.3, at the same time as the features of expansion of data 5.3 . Shaft of computation Classifies InterfaceParametres (file ce_ihm_parametres.py

5.1.2): It is

a subclass of the class Frame (Tk) in which is manufactured

the GUI, called with the launching of CALC_ESSAI. It is made up of two principal frames: a frame 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 frame making it possible to visualize the modal deformed shapes, 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 one selects one result, the routine check_state checks

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 computation of the matrix of MAC is carried out in a file of computation: ce_calcul_expansion.py (routine calc_mac_mode). The routine view_frf makes it possible to visualize the FRF in an additional window
(DispFRFDialogue class , in outils_ihm.py), that one describes below. When one chooses a set of software for visualization, one creates one

instance associated with the choice carried out: classify CalcEssaiGmsh, and CalcEssaiXmgrace for the display of the curves,

- classifies CalcEssaiSalome; and CalcEssaiSalomeCourbes for the display 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. The method called (method of the CalcEssaiSalome class), 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): Classify hat for

the control of the external software . In particular allows

to define the files where the results are printed, and to make the IMPR_RESU necessary. Classes CalcEssaiGmsh (file ce_ihm_parametres.py): Classify for the display of

the deformed shapes in Gmsh. A to note , for the display 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): Classify for the display of the deformed shapes

in Salomé. The routines studylist and Show return

towards the scripts Salome defined for STANLEY, via operator EXEC _ LOGICIEL . For the visualization of MAC, one manufactures in make _mac_salome (in this _calcul_expansion.py) a square mesh, 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 deformed shape, by means of a script
- Salome called salomeScriptMac

. Notice important: a bug was announced in Salomé (file 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): Classify allowing to display a TopLevel

window (news fenestrates depend on the principal

Tk window), to display FRF resulting from a concept dyna_harmo, or result to calculate one harmonic from a base of modes and an excitation "hammer". all the concepts are initialized in the shape of a list with two components, for each

- of the two columns, when certain parameters are modified, one updates all the GUI; thus, if the selected
- concept is a mode_meca, one reveals Frame to choose the assumptions design of result harmonic associated; if the concept is already result harmonic, then one does not need to display this window some attributes of the class: self.dyna: contains the dyna_harmo, with calculating, or already calculated
- ; third is used
    - by the mitre of structural modification, to compute: internal displacement given as starter (definite on a super-mesh self.sumail), self.ddls: for each column, the list of the d.o.f. 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 result selected is a dyna
    - _harmo or a _meca mode, and thus to display, or not, the window of computation of result harmonic, self.param _calc: allows to gather in a dictionary the variables entered the GUI by the user

- for the computation of result harmonic: 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 result harmonic, on physical base: computation is false for the experimental

- results (and does not have, in addition, interest) because the experimental physical matrixes are not known choix_champ: look in the .TACH of data structure if the fields corresponding to displacements, velocities
- 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 with the format
- python; the components of this field available are stored in the .comp of data structure result, affich_FRF: recovery of the function, and recovery of the error message if the selected component and/or the node

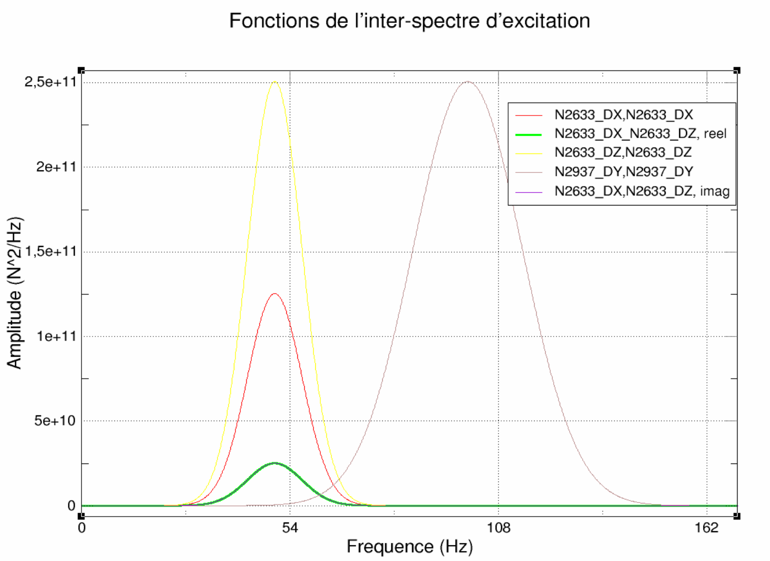
do not exist ; the display is done via the class param_visu. Identification of forces general Presentation: sdlv125a benchmark the purpose of this benchmark is to simulate

5.2 a measurement on an excited cylinder

5.2.1 in three points. The force is defined by

an inter-spectrum size 3, whose functions are the gaussian ones (rustled pink). The forces on DDL N2633_DX and N2633_DZ are correlated, with a phase shift of. The functions are represented on the graph below : Figure 5.2 .1-a: inter-spectrum of excitation. The computation $\pi/2$ having allowed the simulation of the data is carried out with

DYNA_ALEA 5.2.1-a , on modal base. A base of



mode restricted with all the sensors (the "DDL sensors") is calculated with the operator OBSERVATION

, on whom one restores the result physical one with REST_SPEC_PHYS, to obtain a simulated measurement. The purpose of the benchmark is to identify the forces applied from the data of inter-spectrum SPECTPH1 and the projected base of modes MODEIDE1. The identification is done on a reduced number of DDL chosen a priori and called "DDL orders". Validation of the benchmark in interactive to launch the benchmark in interactive, to modify the command file sdlv125

5.2.2 a.comm by writing interactif=1 before

command CALC_ESSAI. The GUI of CALC_ESSAI launches out. To place in the mitre "identification of loading ". In the command debut, to remove key word CODE. Of this way, if the code plants on an Aster error, this one
is caught up with in Exception, and the message is displayed in the GUI of CALC_ESSAI. Choice of the base of modes: to choose MODE_NUM: one uses in this base only the modal characteristics : masses

1) , frequencies and modal dampings
   - Bases deformed shapes observable: model experimental: MODELEXP: contains only 31 nodes, bases modes: MODE_

2) NUM: will be projected with OBSERVATION
   - on MODELEXP, nodes group GPCYLRED: to notch
   - the DDL DX and DZ, to choose a change of reference" NORM", with the option VECT_Y
   - = (0.0, 1.0, 0.0), nodes group GPSUPRED : to notch the DDL DY and to choose a change of reference "CYLINDRIQUE " with AXE_Z = (0.0,
   - 1.0,0.0), To click on "Validating ". Base commandability: model experimental : MODELACT : contains only 3 nodes, bases
   - modes: MODE_NUM:

1) will be projected with OBSERVATION
   - on MODELACT, N1 nodes group: to notch the DDL
   - DX and DZ, nodes group N2: to notch the DDL DY, To click on " Validating"
   - . Inter-spectrum under operation : to choose SPECTPH
   - 1, and to leave Standard field to "DEPL ".
   - Parameters of computation:

1) Not to modify them (not regularization
   - carried out), To click on "Launching" To check ,

1) when one "forgets"
   - one of the preceding operations, that a message appears
   - in the window of messages

. It is necessary to avoid the display of the alignment python in the window comforts, which is not explicit for the user. The computation takes less than one minute. Most of the time of computation is used for the extraction of the inter-spectrum in the form of
matrix python. The computation in itself is almost instantaneous. When computation is finished, one sees all the operations carried out, the last being "Computation of Syy_S: Modal synthesis of displacements". Display of curves: pace of the force identified In the left-hand column of visualization of the results, to choose

1) 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 Y-coordinates (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 array is that pre-declared with the call of the macro-command ("FORCES"). Display of the curves: comparisons measured/synthesized displacements: In the left-hand column, to choose Depl Phy

1) (for measured physical displacements), and in that of right, 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-spectrums. The names are those pre-declared with the call of the macro-command: "DEPL_PHY", and
   - "DEPL_SYN". The creation of these inter-sectres with the Aster format is relatively long. Shaft of computation One mentions here the principal routines used for the realization of preceding computation. Classify

5.2.3 InterfaceIdentification

(file ce_ihm_identification.py): It is a subclass of the class Frame (Tk) in

which is manufactured the GUI of identification , called with the launching
of CALC_ESSAI. Remarks on the behavior of this class: all the calculated inter-spectrums are attributes of this class (inititialisation with a size null, the size

- is given by the class of computation): - coil. Syy: measure - coil. Syy_R: measure recomputed starting from modal displacements, - coil. Sqq: modal displacements , - coil .SQQ: modal forces , - self.SQQ_R: modal forces recomputed starting from the physical efforts , - coil. Sff: physical efforts of dimension Nb _act (many actuators) - coil. Syy_S : measure Re-synthesized 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 computations 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 calculated inter-spectrums (like the variables of access
- , nume_ d.o.f. or nume_mode which is associated for them), and the functions used to extract the functions from them, Methods _ definit_observabilite 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 SelectionNoeud class (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 mesh Methods _calculate_observabilite and _calculate_commandabilite: launching of OBSERVATION: the key words factors for
- the changes of reference and the filtering of the DDL are factories in get_filtres from result of the routine get_selected, described below. Method computations: successive launchings of computations : - calcturb is instance of the CalcEssaiIdentification class, -
- computations and the establishes the links between the concepts of
  the graphic class equivalent concepts of the class of computation, - after
  computation , recovers the new concepts calculated and the associates 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 computation) Methods for the display 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 : display of the curves; use the function _get _graphe_data, which recover the X-coordinates and Y-coordinates of the function in the inter-spectral matrix
  , and the method visu_courbe (InterfaceParametres class) for the sending of the display by Xmgrace or Salome. Classes SelectionNoeud and SelectionMaille (file cata_ce.py): Allows to create a widget displaying the nodes groups associated with a mesh, and the DDL carried

by the nodes of this group (widgets green and purple in

l'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 nodes groups, method set_modele_maillage
- : for each group, recovery of the "true components" associated with the PRNM (see wiki Aster for the description of this
- data structure) with the function find_composantes. Method get_selected: turn over the data of changment of reference and filtering of the DDL for the use of OBSERVATION. Classes CalcEssaiIdentification
- and CalculInverse (file ce_calcul_identification.py): Classes of computation to carry out computation reverses identification of forces

. Note: in the other mitres, most computations are realized by Aster commands
. Here, once the data input are extracted, one uses

only

- functions python (operations on the matrixes, 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 computation of the forces; - the errors in the CalculInverse class

are recovered and a generic

- error message is displayed; in each elementary method of calculating,

  a more precise message (specifying dimensions of the matrixes concerned) is given, - each result is instance of the class Interspectrum (see cata_ce.py); one can associate a classification of DDL physics to him 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). Calc_Z method: computation of the matrix of impédence and its reverse; according to the type of data (accelerations , velocities or displacements, one can have
- to integrate while dividing by, where exp is worth respectively $Z=diag(-\omega^2+\omega_j^2+2\xi\omega\omega_j)$ 2,1 or 0. Calc_SQQ $Zm1$ methods, calc_Sff: the pseudo-opposite of the matrixes calculate respectively and (see U4.90.01, section 6.2.1 for $\omega^{\exp}$ the meaning of these matrixes);
- the too low singular values (below a criterion chosen by the user) $C\Phi$ are put $\Phi^T B$ at zero; after regularization of Tikhonov, the reverse of the singular value is worth, the parameter which can be different ϵ according to the frequency where one is (adjusted by the power): thus, the regularization is weak when one is s_j $\frac{s_j}{s_j^2+\alpha^2}$ before or close to α the frequency of the mode, and it becomes stronger when one moves away from there. The matrixes thus m calculated are put in memory in instances of the class Interspectrum. 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 modal forces SQQ _R from Sff to compare them with SQQ, Method synthesizers _Syy: recompute Syy_S displacements from
- the identified forces Sff to consider the quality general of the inversion, Method _alpha choice
- : the parameter of the regularization of Tikhonov can be modified according to the frequency to which one is compared to the eigenfrequencies
- of the base. Expansion of data α general Presentation of the benchmark sdls112a the purpose of this benchmark, based on an international benchmark (benchmark of the gartor), aims

5.3 to extend the FRF

5.3.1 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 (documentation of reference of the benchmark). Note: In NON-interactive, command CALC_ESSAI is completely useless here , it is enough to use macro-command MACRO_EXPANS, or to connect commands

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 benchmark. Validation of the benchmark in interactive to launch the benchmark in interactive, to modify the command file sdls112a.comm by writing interactif=1 before command CALC

5.3.2 _ESSAI. In the command debut, to remove

key word CODE. Of this way, if the code plants on an Aster error, this one is caught up with in Exception , and the message is displayed in the GUI of CALC_ESSAI. To increase the memory size used to 1024 Mo. To gain in speed of computation, one can also erase or comment on the following operations: TEST _RESU, TEST
_TABLE, MACRO_EXPANS and the commands MAC_MODES, which one will carry out in interactive. The GUI of CALC_ESSAI launches out. To place in the mitre “identification of loading ”. First test : expansion of one result harmonic on a basis of modes: choice
of the base of modes of expansion: to choose MODES and to select only the modes of which

1) the frequency lies between 1 and 100 Hz, base FRF to be extended: to choose
   - 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 result concept: this name must have less than 5 letters ; concepts XXX_ET (result wide), XXX_NX (expansion reduced to the selected sequence numbers
   - bases) and XXX_RD (result wide tiny room) 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 GUI: 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. Test 2: simulation of a FRF from a base of modes: the purpose
     - is to compare a base of experimental FRF with the base of FRF simulated numerically on

2) an experiment “blow of hammer”: to click on box FRF in the mitre “Parameters and visualization”, to choose in one of the columns concept MODES, to choose a node of excitation (example, “N1”
   - , excitation “FX”), a bande de fréquence of computation (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. Test 3: expansion of a base of
   - experimental modes on
   - a numerical basis: to start again computation (to avoid encumbering the memory unnecessarily), in the mitre expansion

1) 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 benchmark is commonplace: the modes to be extended are resulting from projection
   - on the model experimental numerical modes , To choose a list of modes in the two columns, To choose a name of result concept (less than 5 letters) and to click on "Calculating" To compare the deformed shapes results
   - in the mitre "Parameters and visualization
   - ": to compare, for example, experimental modes (MODMES) and the reduced extend modes
   - (XXX_RD), by choosing them in the small double in bottom of the mitre, and clicking on "Deformed shapes"; to use the modes of visualization in GMSH and Salome, To display MAC result: to choose two deformed shapes defined on the same model, and to click on the MAC button; to choose by exmple: the extend modes (XXX_ET) and bases
   - it expansion, the reduced extend modes (XXX_RD) and the experimental base (MODMES). Shaft of computation Classifies InterfaceCorrelation
     - (file ce_ihm_expansion.py): It is a subclass
     - of the class Frame (Tk) in which is manufactured the GUI of expansion , called

5.3.3 with the launching

of CALC_ESSAI. Remarks on the attributes created with the initialization
: self.calculs: it is instance of the CalcEssaiExpansion class, which carries out all Aster computations inherent in the expansion of data (launching of MACRO_EXPANS mainly

- , and MAC_MODE for postprocessing) , self.resu_num and self.resu_exp: the base of expansion (a concept mode_meca) and the result experimental one indicate respectively (a mode_meca or a dyna _harmo); with these variables
- StringVar self.var_resu_num and self.var_resu_exp are associated, 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 GUI) Method setup: all the graphic
- classes contain a method setup, called for cooling when mitre is changed, or when a series of computations 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 computation of expansion itself: checking that the data necessary to computation 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 of the ModeList class), assignment of the attributes of the class
  - of computation by calling the method self.calculs.setup, launching of computations: self.calculs.calc_proj_resu. Classify CalcEssaiExpansion (file ce_calcul_expansion.py): Launching of computations of
  - expansion, mainly macro-command 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 result principal one of the expansion: the base of modes or result harmonic the wide XXX_RD is the reprojection of result wide on the model experimental , if one

- selects whole or part of the base of numerical
  - 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

- computation, all the concepts created are added to the list of the existing objects Aster (self.ce_objects , or mdo) with the command update; in parallel
- the MyMenu menus themselves are brought up to date in the class Interfaces Correlation
- . Structural the purpose of modification general Presentation of the benchmark sdll137 This case test is validating the procedure and the computation of structural modification starting from measured information. One proposes to calculate the variations of the first two

5.4 eigenfrequencies

5.4.1 of a embed-free beam, following

the modification made on a portion of the beam. The modelizations A and D of the case test detail the course of the procedure of computation while making use only of the commands standards of Aster (without passing CALC_ESSAI by the command). The modelizations B and C utilize command CALC_ESSAI. Validation of the benchmark in interactive to launch the sdll137b benchmarks and sdll137c in interactive, to rock the variable interactive with a non-zero value (1 for example) . This variable is localised before call of the command CALC_ESSAI

5.4.2 . In the command debut, to remove

key word CODE. Of this way, if the code plants on an Aster error, this one is caught up with in Exception, and the message is displayed in the GUI of CALC_ESSAI. The GUI of CALC_ESSAI launches out . To place in the mitre " structural Modification" . First test: structural modification by means of 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

1) Models support: MODLSUP Stamps stiffness (support): KASSUP Method (expansion bases): ES Models modification

- : MODLCPL For the choice of the nodes and d.o.f. sensor, DY to select the d.o.f. and DZ of the SENSOR
  nodes group For the choice
  of the nodes and d.o.f. interfaces
  , to select all the d.o.f. of the EXTERNAL
  nodes group (the use
  of the elevator is perhaps
  necessary in order to reach this nodes group) Name of super − mesh : SUMAIL One can then click
  on Validating for the validation of the seized data. The list of the identified eigen 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/models condensed": One keeps the parameters by default for PROJ_MESU_MODAL and modal computation. One clicks then on Calculating in order to obtain the eigenfrequencies of modified structure which

- are displayed in the column Frequencies modified structure. To check
  that the first two calculated eigenfrequencies 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 Validating, 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 deformed shapes **, one clicks** on Deformed shapes in the panel "Comparison initial structure/modified structure". One then sees appearing a window GMSH or SALOME which make it possible to compare

- the initial modal deformed shapes and after modification. Second test: simulation **of** a structural modification by means of 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

2) <u>the first test, except the choice of the method for computation of the base of expansion and the name of the model modification. Method (expansion bases</u>

-): LMME Models modification: MODLX While clicking on Validating, the list of the identified eigen 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 the first eight modes of the "Base of expansion". In the panel "Coupling modification/models condensed", 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 <u>criterion IERI</u> (instead of the criterion of MAC).

- A matrix of weighting is necessary for this criterion. One chooses to balance with the stiffness matrix while clicking
  on Stiffness. While clicking on Validating, 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. Shaft of computation **the two** most important **files python** for the computation of structural modification are: ce_ihm_modifstruct.py and ce_calcul_modifstruct.py. The modulus ce_ihm_modifstruct manages the classes relating to the part graphic interface

5.4.3 and the modulus this

_calcul_modifstruct manages the classes which launch the procedure of computation of structural modification by calling the operators aster. Classify InterfaceModifStruct
This class is a class of the modulus ce_ihm_modifstruct. It is the class hat for the mitre "Modification structural". It has several attributes of which expansion and coupling. self.expansion is instance

<u>of the InterfaceExpansion class</u>
. 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 instance of the InterfaceCouplage class. This class allows to seize and display the various parameters of computation for the estimate of the eigen modes of modified structure. This modified structure is resulting from the coupling enters
the model of condensed initial structure and the model of the modification. Classify CalcEssaiModifStruct This class is a class of the modulus ce_calcul_modifstruct. It makes it possible to connect the operators aster necessary for the computation of the eigen modes of modified structure. It understands several methods of which most important

<u>are: calc_base_es</u>
: calculate the base of expansion by means of the method ES (static expansion). This method launches operator MODE_STATIQUE. The basic vectors obtained are static deformed shapes. calc_base_lmme: calculate the base of expansion by means of
- method LMME (Local Model Modeshapes Expansion). This method carries out a modal computation with operator MODE_ITER_SIMULT with option PLUS_PETITE. The basic vectors obtained are eigen modes
- . condensation : carry out the condensation of the measured model and creates the super-mesh representing initial structure. This method connects operators PROJ_MESU_MODAL, MACR_ELEM_STAT and DEFI_MAILLAGE. modes_modele_couple: calculate the eigen modes

- `of the system` couples (initial structure + modification) with the parameters of computation provided by the user. This method also carries out the retro-projection `of the results` `on the mesh` `measures`
- `operator DEPL_INTERNE` by means of. indicateur_choix_base_expansion: compare the field of displacement with the interface obtained with the model coupled and those obtained by static expansion. One raises the field of displacement of 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 operators PROJ_MESU_MODAL, REST_GENE_PHYS, PROJ_CHAMP and MAC_MODES. Computations of inter spectrums, car spectrums and transfer functions general Presentation transfer of the zzzz241a benchmark the purpose of this benchmark is to validate the elementary functions of processing of the signal `integrated` in `operator CALC_SPEC`. One `simulates` the temporal

5.5 response of an oscillator with 4 degrees of freedom, and one calculates

5.5.1 various transfer transfer functions. For

more precise details, to see U4.32.21 documentation (Doc. of CALC_SPEC) Note: In NON-interactive, command CALC `_ESSAI` is completely useless here, it is enough to use command CALC_SPEC. For the validation in interactive of CALC_ESSAI, one proposes to reproduce the expansions carried out in this benchmark. `Validation`

of the benchmark

in interactive to launch the benchmark `in interactive` , to modify the command file zzzz241a.comm by replacing `all` the lines located under the command `tab_rep=CRÉA_TABLE` (...) (lines 196 to 307) by TEST_CE=CALC_ESSAI (INTERACTIF=' OUI'

5.5.2 ,), and not to use the file of

zzzz241a.com1 poursuite. In the command debut, to remove key word CODE. Of this way, if the code plants on an Aster error, this one `is caught up with in Exception` , and the message is displayed
`in the GUI of CALC_ESSAI. The GUI`
of CALC_ESSAI launches out. To place in the mitre "Processing
of the signal ". `First` test: Computation of `the estimators` H1 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`
"Arrays `available` ", to select "tab_rep" to put the name in intensified brightness

1) , 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 "Period", To inform a covering
- from "50", and to click on "For hundred", To select "H1" under the button "Transfer", and to compute: to click on "
- Transfer" the estimators H1.
- 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 compute: to click on
- "Transfer" 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". Second test: Computation of inter spectrums and car spectrums: While setting out again of
- the initial window of CALC_ESSAI, it is necessary: In the list "Arrays available", to select "tab_rep" to put the name in intensified brightness, To click on the button "=>"

2) <u>to fill out the lists entitled "Points of measurements"</u> 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 result, this selection, for the computation of the spectrums, 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 compute: To click on "Interspectrums" them inter spectrums. To display the results by
- selecting in the list "Interspectrums
- ". To choose the format of the data to be displayed, as well as
- the nature of the axes, and to click on "Visualizing". Shaft of computation
- the mitre "Processing of the signal" calls on a single
- class. Classify InterfaceCalcSpec (file ce_calc_spec.py): Classify allowing to manage the mitre "Processing of the signal". One there definite at the same time graphic objects and the calls to

5.5.3 the operator CALC

_SPEC, who carries out computations. The principal methods are: InterfaceCalcSpec

<u>**: method for the generation `of`** the graphic interface</u>
(lists, frames, 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 arrays. display_mes: generation of the curves before display calc_intespec
- `: call` to CALC_SPEC for the computation of inter spectrums calc_coherence: call to CALC_SPEC for
- `the computation` of the functions of coherences calc_transfert: call to CALC_SPEC for the computation of transfer functions
- `crea_tab_fonc` the transfer: creation of a concept
- `TABLE_FONCTION` from `a selection` of functions
-
-
-