Operator CALC_EUROPLEXUS

1 Goal

This operator allows to control the execution of a Europlexus study since a command file *Code_Aster*.

The order CALC_EUROPLEXUS allows to define and carry out a Europlexus study while remaining entirely in the environment *Code_Aster*. The resolution is done in background by Europlexus, without the user not having to worry about the input files Europlexus. More precisely, all the setting in data is done by orders *Code_Aster* (possibly since an assistant Salomé-Meca), and the resolution is done by the order *Code_Aster* taking care of the piloting of Europlexus. This one takes in arguments the concepts Aster, built the command file Europlexus, controls the execution of Europlexus, then rebuilt, on the one hand a total result Aster which can be used then with Aster in postprocessing (with Stanley for example) or in continuation of calculation, and on the other hand a series of curves generated by Europlexus (in the form of table).

The macro-order produces a structure of data of the type evol_noli and possibly a table containing the evolution of certain sizes in certain moments.

Notice

It is necessary to specify the keyword DEBUG=_F (HIST_ETAPE=' OUI') in BEGINNINGCONTINUATION to use this macro-order.

2 Syntax

```
evol = CALC_EUROPLEXUS (

        ♦ /  MODEL        =     Mo,                    [model]
          CARA_ELEM    =     carac,                   [cara_elem]
          CHAM_MATER   =     chmat,                   [cham_mater]

          /  ETAT_INIT =  _F (
              ♦  RESULT       = evolno            [evol_noli]
              ◊  CONSTRAINT       =  /  'NOT'         [DEFECT]
                                     /  ' YES '       [TXM]
              # so FORCED = 'YES':
              ◊  VARI_INT         =  /  'NOT'         [DEFECT]
                                     /  'YES'         [TXM]
              ◊  SPEED        =  /  'NOT'         [DEFECT]
                                     /  'YES'         [TXM]
              # so FORCED = 'NOT':
              ◊  NITER            =  /  1             [DEFECT]
                                     /  niter         [I]
              ◊  BALANCE          =  /  'YES'         [DEFECT]
                                     /  'NOT'         [TXM]
                    ),
        ♦  BEHAVIOR =   _F (
              ♦  RELATION         = 'ELAS'            [DEFECT]
                                  = 'GLRC_DAMAGE'     [TXM]
                                  = 'VMIS_ISOT_TRAC'  [TXM]
                                  = 'VMIS_JOHN_COOK ' [TXM]
                                  = 'BPEL_FROT'       [TXM]
              ♦  GROUP_MA         = l_grma            [l_gr_GROUP_MA]

        ◊  SOFTWARE     =
           /  '/home/rd-ap-simumeca/outils/europlexus' [DEFECT]
           /  way,                                  [TXM]

        ◊  VERSION_EUROPLEXUS     =  /  '2015_DEV'       [DEFECT]
                                     /  '2014'
                                     /  '2015'
                                     /  'DEV.'

        ◊  LAUNCHING    =      /  'YES'                [DEFECT]
                               /  'NOT'                [TXM]

        ◊  FONC_PARASOL =   _F (
              |      ♦  NFKT         = nfkt           [fonction_sdaster]
                     ♦  NFKR         = nfkr           [fonction_sdaster]
              |      ♦  NFAT         = nfat           [fonction_sdaster]
                     ◊  NFAR         = nfar           [fonction_sdaster]
              ♦  GROUP_MA     = gma                   [l_gr_GROUP_MA]
                        ),

        ♦  EXCIT        =  _F (
              ♦  LOAD      =      cho,                [char_meca]
              ♦  FONC_MULT =      fi,                 [function/formula]
                           ),

        ♦  CALCULATION      =   _F (
            ♦  /  TYPE_DISCRETISATION =  'CAR',    [DEFECT]
```

```
                    ♦  CSTAB          =  /  0.3,        [DEFECT]
                                         /  cstab,      [R]

              /  TYPE_DISCRETISATION=    'UTIL',        [TXM]
                    ♦  PASFIX         =  pasfix,       [R]
           ♦  INST_INIT          =  tini,               [R]
           ♦  INST_FIN           =  tfin,               [R]
           ◊  NMAX               =  nmax,               [R]
                    ),
     ◊  DAMPING       =  _F  (
           ♦  TYPE_AMOR      =  'QUASI_STATIQUE',      [DEFECT]
           ♦  FREQUENCY      =  freq,                  [R]
           ♦  COEF_AMOR      =  amor,                  [R]
           ◊  INST_DEB_AMOR=  instd,                   [R]
           ◊  INST_FIN_AMOR=  instf,                   [R]
                              ),
     ◊  OBSERVATION      =  _F  (
              ♦  NOM_CHAM  =  /  'DEPL'                 [DEFECT]
                              /  'QUICKLY'              [TXM]
                              /  'ACCE'                 [TXM]
                              /  'SIEF_ELGA'            [TXM]
                              /  'EPSI_ELGA'            [TXM]
                              /  'VARI_ELGA'            [TXM]

              ♦  /  PAS_INST  =  pas_inst               [R]
                 /  PAS_NBRE  =  pas_nbre               [I]
                 /  INST      =  l_inst                 [listr8]
                 /  NUME_ORDRE = l_nume                 [l_I]

              ◊  /  GROUP_NO     =  grno                [l_gr_noeud]
                 /  TOUT_GROUP_NO=  'YES'               [TXM]

              ◊  /  GROUP_MA     =  grma                [l_gr_GROUP_MA]
                 /  TOUT_GROUP_MA=  'YES'               [TXM]

                              ),

     ♦  FILING =  _F  (
              ♦  /  PAS_INST  =  pinst                  [R]
                 /  PAS_NBRE  =  pnbre                  [I]
                 /  INST      =  l_inst                 [listr8]
                 /  NUME_ORDRE =  l_nume                [l_I]
                       )

     ◊  CURVE  =  _F  (
              ◊  UNITE_ALIT    =  ualit                 [I]
              ◊  NOM_CHAM      =  ncham                 [TXM]
              ◊  NOM_CMP       =  ncmp                  [TXM]
              ◊  /  GROUP_NO   =  node                  [l_gr_noeud]
                 /  GROUP_MA   =  grma                  [l_gr_maille]
                    ◊  NUM_GAUSS = N                    [I]
              ♦  /  PAS_INST_COURBE     =  picourbe     [R]
                 /  PAS_NBRE_COURBE     =  pncourbe     [I]
                 /  INST_COURBE         =  l_inst       [listr8]
                 /  NUME_ORDRE_COURBE   =  l_nume       [l_I]
              ◊  TABLE_COURBE =  CO ('table')           [tabl_*]
                    ),

     ◊  FIELDS =  _F  (
```

```
            ◊  GROUP_MA  =  dom_gma                [l_gr_maille]
            ◊  IDENTIFIER   =  dom_id              [I]
                    ),
◊  INTERFACES   =  _F (
            ◊  GROUP_MA_1   =  int_gma1         [l_gr_maille]
            ◊  GROUP_MA_2   =  int_gma2         [l_gr_maille]
            ◊  SHEET  =  sheet                  [R]
            ◊  IDENT_DOMAINE_1  =  int_dom1     [I]
            ◊  IDENT_DOMAINE_2  =  int_dom2     [I]
                    ),

◊  INFORMATION  =  /  1,                                [DEFECT]
            /  2,                               [I]
            )
```

3 Restrictions of use

CALC_EUROPLEXUS defines and carries out a Europlexus study. For that, it uses, not only concepts, but too keywords indicated in some of the preceding orders (AFFE_CARA_ELEM , AFFE_CHAR_MECA …).
For that, it is necessary to add the keyword DEBUG=_F (HIST_ETAPE=' OUI') in BEGINNING or CONTINUATION .

For the same reason, it is obligatory that these orders “relationships” are defined in the command set on the same level as CALC_EUROPLEXUS (and not in one INCLUDE for example).

Lastly, the assignments in these same orders must be made by groups of meshs (keywords GROUP_MA*) and not by meshs.

4 Operands

4.1 Opérandes MODEL/CHAM_MATER/CARA_ELEM

```
♦  /  MODEL      =      Mo,
      CARA_ELEM    =      carac,
      CHAM_MATER   =      chmat,
```

These keywords make it possible to inform:

- the name of the model (Mo) whose elements are the object of mechanical calculation. Only mechanical calculations are authorized, and for modelings Q4GG, BARS (section GENERAL in AFFE_CARA_ELEM), POU_D_E (rectangular section and circular in AFFE_CARA_ELEM), 3D (mesh HEXA8 and TETRA4), 3D_SI (mesh HEXA8), DIS_T and DIS_TR.
- the name of the material field (chmat) affected on the grid. Attention, all the groups of meshs of the model must be associated with a material.
- the name of the characteristics (carac) elements of hull, beam, pipe, bars, cable, and discrete elements affected on the model Mo . This operand is not obligatory if no element of the model needs some (ex: 3D).

4.2 Keyword ETAT_INIT

```
♦  /  ETAT_INIT     = _F (
      ♦  RESULT        =      resu,
      ◊  CONSTRAINT
      ◊  VARI_INT
      ◊  SPEED
      ◊  NITER
      ◊  BALANCE
```

This keyword makes it possible Europlexus to begin calculation starting from an initial state resulting from a concept result of Code_Aster. The fields provided to Europlexus are those corresponding to the last moment of calculation of the concept result given.

The field of displacement is transmitted in all the case. field speed can be transmitted in the case of a dynamic initial state (SPEED = 'YES'), but even in this case, the field of acceleration is not transmitted because it is useless with EPX which calculates it automatically.

Note:

One recovers the model, the field of material and the elementary characteristics of the concept result.

4.2.1 Operand RESULT

♦ RESULT

Concept result providing the grid and the fields of displacements and constraints which will be used as initial state in Europlexus.

4.2.2 Operand CONSTRAINT

◊ CONSTRAINT

If CONSTRAINT = ' OUI', the stress field SIEF_ELGA fact part of the initial state, if not only the field of displacement is given, the initial constraints are then calculated starting from displacements by Europlexus.

Only modelings BAR, Q4GG , 3D and 3D_SI are compatible with CONSTRAINT = ' OUI'.
Contrary to the other elements, for 3D_SI , the Code_Aster models and EPX not being completely the same ones, an initial state balanced in Code_Aster will not be it in EPX. The functionality is all the same to activate but it is necessary in this case to activate the keyword EQUI .

4.2.3 Operand VARI_INT

◊ VARI_INT

If VARI_INT = ' OUI', a field of internal variables (VARI_ELGA) fact part of the initial state. This field is not null solely on the affected meshs of a behavior for which a transformation of the internal variables of Code_Aster towards EPX is developed. Messages of alarms prevent when internal variables put at zero are sent to EPX and when the field of internal variables is null on all the meshs of the model (this last specifying whereas the use of the operand is without effect).
This operand is available only if CONSTRAINT = 'YES'.

4.2.4 Operand VITESSE

◊ VITESSE

If VITESSE = ' OUI', field of vitess (VITE) fact part of the initial state who is thus dynamic.
This operand is available only if CONSTRAINT = 'YES'.

4.2.5 Operand NITER

If CONSTRAINT = ' NON', this operand indicates to Europlexus in how much stages (not of time) to recompute the constraints starting from displacement. During this calculation initial displacement given is regarded as an imposed displacement. At the stage $i=1,\ldots,niter$, imposed displacement is multiplied by $\frac{i}{niter}$.

4.2.6 Operand BALANCE

◊ BALANCE

When a result is transferred from a computer code to another, it often happens that a state balanced in the first code is not it completely any more in the second. To avoid that, Europlexus lays out of a functionality to balance perfectly an initial state by adding what it is necessary for the external forces. This functionality will be activated if one gives EQUILIBRE=' OUI' (value by default).

4.3 Keyword BEHAVIOR

```
♦ BEHAVIOR =   _F (
                  ♦ RELATION        =  'ELAS'              [DEFECT]
                                    =  'GLRC_DAMAGE'       [TMX]
                                    =  'VMIS_ISOT_TRAC'    [TMX]
                                    =  'VMIS_JOHN_COOK'    [TMX]
                                    =  'BPEL_FROT'         [TMX]
                  ♦  GROUP_MA       =  l_grma              [l_gr_GROUP_MA]
```

On the model of the operators STAT_NON_LINE and DYNA_NON_LINE, the keyword BEHAVIOR allows to assign a behavior to the groups of meshs modelled in calculation.

The only behaviors available are 'ELAS', 'GLRC_DAMAGE', 'VMIS_ISOT_TRAC', 'VMIS_JOHN_COOK' and 'BPEL_FROT'. They are indicated by the keyword RELATION.

4.4 Keyword SOFTWARE

◊ SOFTWARE

Way towards the site of the Europlexus program. Allows to specify the way towards the script of Europlexus launching.

It is possible to position the variable of environment ASTER_EUROPLEXUS to overload the value by default fixed at /home/rd-ap-simumeca/outils/europlexus.

4.5 Keyword VERSION_EUROPLEXUS

◊ VERSION_EUROPLEXUS

Keyword allowing for choice of the version of Europlexus among those installed on the centralized waiters (ASTER5 and ATHOSDEV). By default it is the version 2015_DEV, stabilized state of the version of development, who is used. The chain DEV. allows to point on the version of development.

4.6 Keyword LAUNCHING

◊ LAUNCHING

Allows to stop (LANCEMENT=' NON') after the generation of the data files of Europlexus (command file and grid). All the Europlexus files are recoverable in a repertoire defined by REPE in mode result (R) in profile ASTK.

If calculation EPX fails, a generic message is transmitted by Code_Aster inviting to go to seek the information given by EPX in the file ".mess" right before the error message.

4.7 Keyword FONC_PARASOL

♦ FONC_PARASOL

This keyword makes it possible to define the functions of stiffness, and damping, translation and rotation of the elastic supports definite in the keywords factors RIGI_PARASOL and DISCRETE order AFFE_CARA_ELEM.

It is possible to combine at the same time a carpet of springs of ground with a carpet of shock absorbers (standard dashpots). Within this framework, one can mix descriptions: for example to couple springs K_TR_D_N with 6 components, of which stiffnesses of rotation, with shock absorbers of the type A_T_D_N . In Europlexus, the damping coefficients in rotation are worth then implicitly 0.

Of course, homogeneous descriptions on the level of the degrees of freedom, type K_TR_D_N with A_TR_D_N or K_T_D_N with A_T_D_N are also authorized. In all the cases, the arguments which follow (NFKT , NFKR , NFAT and NFAR) must be specified in coherence with the degrees of freedom of stiffness and damping.

4.7.1 Operand NFKT

♦ NFKT

Allows to define the function of translation of the stiffnesses following the total axes.

4.7.2 Operand NFKR

♦ NFKR

Allows to define the function of rotation of the stiffnesses following the total axes.

4.7.3 Operand NFAT

♦ NFAT

Allows to define the function of translation of following depreciation the total axes.

4.7.4 Operand NFAR

♦ NFAR

Allows to define the function of rotation of following depreciation the total axes.

4.7.5 Operand GROUP_MA

♦ GROUP_MA

Inform the groups of meshs to which the various functions declared in the occurrence of the keyword factor will be affected.

Note:

- So elementary characteristics are declared on a group of meshs by the keyword DISCRETE of AFFE_CARA_ELEM then this group of meshs must necessarily appear in this operand for one of the occurrences of FONC_PARASOL.
- So springs of ground are declared on a group of meshs by the operand GROUP_MA_POI1 keyword RIGI_PARASOL of AFFE_CARA_ELEM then this group of meshs must necessarily appear in this operand for one of the occurrences of FONC_PARASOL.

4.8 Keyword EXCIT

♦ EXCIT

This keyword factor makes it possible to define a load in each occurrence. These loads result from the operator AFFE_CHAR_MECA.

4.8.1 Operand LOAD

♦ LOAD

Types of loads takings into account by CALC_EUROPLEXUS are the following ones:

- DDL_IMPO : it makes it possible to declare displacements imposed. If they are blockings, it is preferable not to associate a multiplying function with this loading. Thus information will be translated by the connection "BLOQ" EPX, which makes it possible to treat all the ddls in the same occurrence. If a multiplying function is then associated the loading will be represented by the connection "DEPL" EPX, it can then have one keyword by occurrence there among DX, DY, DZ, DRX, DRY MARTINI, DRZ. To represent a displacement imposed other that a blocking, it is obligatory to associate a multiplying function with the loading.
- FORCE_COQUE/PRES : pressure on a hull, one must obligatorily associate with this kind of load with a multiplying coefficient (keyword FONC_MULT). Other keywords of FORCE_COQUE are not authorized.
- RELA_CINE_BP : relations kinematics resulting from the macro-order DEFI_CABLE_BP, defines connections between degrees of freedom of concrete and cables.
- PRES_REP/PRES : pressure on faces of elements 3D, one must obligatorily associate with this kind of load with a multiplying coefficient (keyword FONC_MULT). Other keywords of PRES_REP are not authorized. **Attention, in Code_Aster this loading is authorized on the hulls but this use in CALC_EUROPLEXUS will lead to an error of calculation EPX.**
- FORCE_NODALE : allows to declare forces nodal in the same way as for a calculation with the other operators of Code_Aster. As for DDL_IMPO, one can declare one keyword among FX, FY, FZ, MX, MY and MZ. One must obligatorily associate with this kind of load with a multiplying coefficient (keyword FONC_MULT).

4.8.2 Operand FONC_MULT

◊ FONC_MULT

Multiplying function of the time of the loading defined by the current occurrence of the keyword factor EXCIT.

4.9 Keyword CALCULATION

♦ CALCULATION

This keyword factor makes it possible to choose the parameters of calculation to be used.

4.9.1 Operand TYPE_DISCRETISATION

♦ TYPE_DISCRETISATION = 'CAR', [DEFECT]
'UTIL',

Allows to choose between an automatic discretization in time ('CAR', it is then necessary to specify CSTAB) and a discretization imposed by the user ('UTIL', it is then necessary to specify PASFIX). INST_INI and INST_FIN specify initial and final time calculation.

4.9.2 Operand CSTAB

♦ CSTAB = / 0.3, [DEFECT]
/ cstab,

Coefficient safety taken during the step of time of stability estimated (i.e. critical) for each element. The value by default is of $0,3$.

4.9.3 Operand PASFIX

♦ PASFIX

This parameter is a short cut making it possible to assign a step of fixed time user. Must be used in conjunction of TYPE_DISCRETISATION = UTIL.

4.9.4 Operand INST_INIT

♦ INST_INIT

Initial moment of calculation. In the case of a calculation of recovery, this parameter is ignored (it can however be left), because the new value of initial time is read in the file of recovery.

4.9.5 Operand INST_FIN

♦ INST_FIN

Final moment of calculation.

4.9.6 Operand NMAX

◊ NMAX
Maximum number of steps of calculation. The value not defect is: 1000000 .

4.10 Keyword DAMPING

♦ DAMPING

This keyword factor makes it possible to add damping to calculation EPX.

4.10.1 Operand TYPE_AMOR

♦ TYPE_AMOR = 'QUASI_STATIQUE', [DEFECT]

Allows to choose the type of desired damping. For the moment only the value 'QUASI_STATIQUE' is available.

4.10.2 Operand FREQUENCY

♦ FREQUENCY

The frequency of damping defines.

4.10.3 Operand COEF_AMOR

♦ COEF_AMOR

Defines the damping coefficient.

4.10.4 OperandS INST_DEB_AMOR and INST_FIN_AMOR

◊ INST_DEB_AMOR
◊ INST_FIN_AMOR

These two keywords make it possible to give the moment from which damping must be activated and that from which it must cease. These two keywords must be well informed unit, if they are not to it damping will be active throughout calculation.

4.11 Keyword OBSERVATION

Activate the impression in the output file “listing” of EPX of the fields desired on the entities (nodes or elements) asked by the user.

4.11.1 Operand NOM_CHAM

◊ NOM_CHAM

Inform the name of the fields to be written in the listing.

4.11.2 Operands PAS_INST/PAS_NBRE/INST/NUME_ORDRE

◊ / PAS_INST
/ PAS_NBRE
/ INST
/ NUME_ORDRE

Determine the list of the moments for which one wishes posting:
- by frequency defined by the time interval: PAS_INST ;
- by many steps of time: PAS_NBRE ;
- by a list of moments: INST ;
- by a list of sequence number: NUME_ORDRE .

The four keywords can be simultaneously indicated. The list of the selected moments will be then the union of the moments defined by the various keywords.

4.11.3 Operands GROUP_NO/TOUT_GROUP_NO

◊ / GROUP_NO
/ TOUT_GROUP_NO

Determine the nodes for which one wishes to visualize information:
- With certain nodes, through the list defined in a group: GROUP_NO ;
- For all the groups of nodes of the grid: TOUT_GROUP_NO.

4.11.4 Operands GROUP_MA/TOUT_GROUP_MA

◊ / GROUP_MA
/ TOUT_GROUP_MAILLE

Determine the meshs for which one wishes to visualize information:
- On certain meshs, through the list defined in a group: GROUP_MA ;
- For all the groups of meshs of the grid: TOUT_GROUP_MA.

4.12 Keyword FILING

Allows to select the moments for which one wishes that the results be filed in the file with format MED written by EPX. It is starting from this file that is rebuilt the Aster result. The recovery of this information is rather expensive, it is advised to file only the moments strictly necessary for the postprocessing or the continuation of calculation.

4.12.1 Operands PAS_INST/PAS_NBRE/INST/NUME_ORDRE

See § 4.11.2 .

4.13 Keyword CURVE

Directive EPX " EXIT GRAP " allows, thanks to its keyword " CURVE ", to store in the form of table the values in the course of the time of various sizes, typically a component of a field for a node or a point of Gauss.

This keyword makes it possible to put this information in the shape of an object sd_table of Code_Aster. This object is defined by the keyword TABLE_COURBE.

4.13.1 Keyword NOM_CHAM/NOM_CMP

◊ NOM_CHAM
◊ NOM_CMP

Selected field and component.

4.13.2 Operand GROUP_NO

◊ GROUP_NO

Node which one wishes to store the component of interest. The group of nodes must contain that only one node.

4.13.3 Operands GROUP_MA and NUM_GAUSS

◊ GROUP_MA
◊ NUM_GAUSS

Mesh which one wishes to store the component of interest. The group of mesh should contain only one nets.

NUM_GAUSS indicate the number of the point of Gauss of the element post-to be treated. If the element has less points of Gauss than the number required then EPX will fail this stage.

4.13.4 Operand NOM_COURBE

Name of the column concerning the present occurrence of CURVE in the table created. The name should not make more than 16 characters without what it will be truncated.

4.13.5 Operands PAS_INST_COURBE/PAS_NBRE_COURBE/INST_COURBE/NUME_ORDRE_COURBE

♦ / PAS_INST_COURBE
/ PAS_NBRE_COURBE
/ INST_COURBE
/ NUME_ORDRE_COURBE

These keywords define the moments of filings for the keyword CURVE.
Same logic qu is followed ‘in the paragraph 4.11.2.

Note:
All values filed under the keyword CURVE divide the same list of moments of filing.

4.13.6 Keyword TABLE_COURBE

◊ TABLE_COURBE

The table where the values resulting from the keyword are stored CURVE.

4.14 Keyword FIELDS

Defines under fields for the studies multi-fields.

4.14.1 Keyword GROUP_MA

◊ GROUP_MA

For each under field it is necessary to have a group of mesh GROUP_MA.

4.14.2 Keyword IDENTIFIER

◊ IDENTIFIER

Each under field defines by its group of meshs, must also have one identifier.

4.15 Keyword INTERFACES

◊ INTERFACES

Defines the interfaces between under fields previously definite.

4.15.1 Keyword GROUPE_MA_1/GROUPE_MA_2

◊ GROUPE_MA_1
◊ GROUPE_MA_2

These two groups of meshs define the edges of the two pennies fields in contact between them.

4.15.2 Keyword SHEET

◊ SHEET

Specify the tolerance used to pair the nodes of the edges GROUP_MA_1 and GROUP_MA_2.

4.15.3 Keyword IDENT_DOMAINE_1/IDENT_DOMAINE_2

◊ IDENT_DOMAINE_1
◊ IDENT_DOMAINE_2

These two words refer to the respective identifiers of under fields, already specified.

4.16 Keyword INFORMATION

◊ INFORMATION =/ 1, [DEFECT]
/ 2,

Allows to control the level of message of the macro-order.

Note:

- *If the keyword IMPR_MACRO=' OUI' is present in the order BEGINNING, then all orders of the macro-order CALC_EUROPLEXUS will be printed in the file of messages.*
- *The keyword INFORMATION is transmitted to all the orders used in the macro-order. INFORMATION = 2 can thus print much information.*
- *In all the cases, the command file Europlexus is printed in the file of messages.*