Procedure IMPR_RESU with the formats `RESULT` and `ASTER`

1 Goal

To write the grid or the results of a calculation in a file with the format `RESULT` or grid with the format `ASTER`.

Currently this procedure makes it possible to write with the choice:

- a grid,
- fields with the nodes (of displacements, temperatures, clean modes, static modes,…),
- fields by elements with the nodes or the points of GAUSS (of constraints, generalized efforts, internal variables…).

Procedure IMPR_RESU also allows to write a grid and/or results on a file in sight, in particular of their graphic visualization (cf documents [U7.05.01] (format `IDEAS`), [U7.05.21] (format `MED`) and [U7.05.32] (format `GMSH′)).

For the concepts of the type result, one can print only part of information, by selecting the fields and the sequence numbers which one wishes to exploit.

For the impressions with the format `RESULT`, it is possible to select the topological entities (groups of nodes and groups of meshs) on which one wants to print the results. One can also reduce the number of the information printed by asking for only the impression of certain components, values understood in an interval chosen by the user or of extreme values. One can combine all the possibilities, for example to obtain the maximum value of a component given, on a particular topological entity and in an interval of values chosen by the user.

The format `RESULT` does not have to be used as a basis to store or to exchange results, its format is not fixed.

2 Syntax

```
      IMPR_RESU    (
      ◊  MODEL = Mo,                                             [model]
      ◊  UNIT = links,                                           [I]
      ◊  FORMAT =  / 'RESULT',
                   / 'ASTER',
      ◊  PROC0 =   / 'YES',                                      [DEFECT]
                   / 'NOT',
      ◊  INFORMATION =    / 1,                                          [DEFECT]
                   / 2,
      /  RESU = (_F   (
         #  Syntax of procedure IMPR_RESU to the format 'RESULT'
         ♦  |  GRID = MY,                                     [grid]
            |  /  CHAM_GD      =  CH_GD,                         [cham_gd]
               /  RESULT    =  RESU,                          [result]
            # Extraction of a field of size of resu
               ◊  /     TOUT_CHAM    =  /'YES',              [DEFECT]
                                        /'NOT',
                  /     NOM_CHAM        =  L_NOMSYMB,        [l_K16]
               ◊  /     TOUT_ORDRE      =  'YES',            [DEFECT]
                  /     NUME_ORDRE      =  LORDRE,           [l_I]
                  /     LIST_ORDRE      =  LENTI,            [listis]
                  /     NUME_MODE       =  LMODE,            [l_I]
                  /     NOEUD_CMP       =  LNOECMP,          [l_K16]
                  /     NOM_CAS         =  NCAS,             [l_K16]
                  /  /  ANGLE           =  LANGL,            [l_R]
                     /  FREQ            =  LFREQ,            [l_R]
                     /  LIST_FREQ       =  LREEL,            [listr8]
                     /  INST            =  LINST,            [l_R]
                     /  LIST_INST    =  LREEL,               [listr8]
                        ◊ |  PRECISION =  / PREC,            [R]
                                             / 1.0D-3,       [DEFECT]
                          |  CRITERION    =  / 'RELATIVE',   [DEFECT]
                                             / 'ABSOLUTE',
               # Parameters
               ◊  /  TOUT_PARA    =  /'YES',
                                     /'NOT',                 [DEFECT]
                  /  NOM_PARA     =  L_NOMPARA,              [l_K16]
               ◊  FORM_TABL       =  /'YES',
                                     /'EXCEL',
                                     /'NOT',                 [DEFECT]

               ◊  PART   =  /  'REAL',
                            /  'IMAG',
                            /  'MODULE',
                            /  'PHASE',
            # Selection of the components
            ◊  /  TOUT_CMP         =  'YES',                 [DEFECT]
               /  NOM_CMP          =  L_NOMCMP,              [l_K8]
            # Selection of the topological entities
            ◊  /     ALL       =  'YES',                  [DEFECT]
               /  |  GROUP_NO     =  L_GRNO,                 [l_gr_noeud]
```

```
              |   GROUP_MA      =  L_GRMA,                  [l_gr_maille]
        # Selection on the values
        ◊      VALE_MAX         =  /‘YES’,
                                   /‘NOT’,                  [DEFECT]
        ◊      VALE_MIN         =  /‘YES’,
                                   /‘NOT’,                  [DEFECT]
        ◊      BORNE_SUP    =  VSUP,                        [R]
        ◊      BORNE_INF    =  VINF,                        [R]

        # Impression of the coordinates
        ◊      IMPR_COOR    =  /‘YES’,
                               /‘NOT’,                      [DEFECT]

        # Choice of the format of writing of the actual values
        ◊      FORMAT_R  =  /  FORMS,
                            /  ‘1PE21.14’,                  [DEFECT]

        ◊      SOUS_TITRE   =  ST,                          [KN]

    #   Syntax of procedure IMPR_RESU to the format ‘ASTER’

    ♦   |   GRID      =  MY,                         /   [grid]
                                                         /   [skeleton]
        ◊   SOUS_TITRE   =  ST,                             [KN]

    )),

)
```

3 Typography in documentation IMPR_RESU

For understanding the documentation well of IMPR_RESU, it should be noted that the term 'RESULT' gather in *Code_Aster* three rather different concepts:

- a kind of concept,
- a kind of file,
- a format of writing.

Indeed, on the one hand, one indicates by result, the quantities produced by an operator of calculation *Aster* producing several fields (evolutionary calculation, calculation of clean modes, etc…).

These data (results) are contained in a structure of data *Aster* of generic type said result. This generic type is declined under types, for example mode_meca, evol_elas, etc, of which the mode of storage differs somewhat. A concept of the type result only one under type has. In documentation *Aster* the names of the types of concept are in small letters Courier.

In addition, results intended for the impression, in particular those resulting from IMPR_RESU, are versed in a file indicated by file RESULT (associate with unit 8, type resu in astk). The format of writing associated with this file intended for the impression is also baptized RESULT.

In short :

result	data resulting from the resolution of the system,
result	generic designation of the type of concept (of the structure of data) which contains the result,
evol_ther mode_meca	name of under type of the result of the type result,
FILE	(without quote) designation of the file which contains the results, independently of the format of its contents,
'RESULT'	(with quotes) designation of the format (*Aster*) file which contains the results.

4 Operands FORMAT, PROC0, UNIT and INFORMATION

Procedure IMPR_RESU allows to write a grid and/or values of results in various formats in sight, in particular, of a graphic visualization [U7.05.01], [U7.05.11], [U7.05.21] and [U7.05.31].

The operand FORMAT allows to specify the format of impression of a result.
The format by default is 'MED' (*cf.* [U7.05.21]). It makes it possible to print the grid and/or the results in form listing.

The operand PROC0 whose value by default is 'YES', allows to restrict the impression on the processor of row 0. If one affects the value to him 'NOT', the impressions will be carried out on all the processors.

One specifies in which file the data will be written via the keyword UNIT (cf orders DEFI_FICHIER). By default, the logical file of unit is used 8 with format 'RESULT' and 26 with the format 'ASTER'.

The keyword INFORMATION when it is equal to 2 makes it possible to obtain information on the impressions carried out by the order.

5 Keyword RESU

This keyword factor makes it possible to specify the results to print and the format according to which one wants to print them.

6 Operands CHAM_GD and RESULT

The values of the computed fields are stored in structures of data called fields of size. These fields of size can be directly accessible (concept cham_gd) or to be in a structure of data gathering several fields of size (concept result).

- a field of size is a structure of data which makes it possible to store fields defined in the nodes (cham_no_*) or of the fields defined by elements (cham_elem_*). For the fields by elements, one distinguishes the fields defined in the nodes from the elements and the fields defined in the points of GAUSS of the elements,
- a concept result is composed of one or more fields of size. For example, with each step of calculation one stores in the structure of data result, the field of size displacement. This structure is thus indexed by a matric structure of order 2, whose index is, for example, the list of the moments of calculation and the other the whole of the computed fields (displacements, constraints, deformations,…).
  One reaches in this case a field of size by specifying the value of a variable of access (sequence number, moment, frequency, number of mode…) and a field name (‘DEPL’, ‘SIGM_ELNO’,…). There exist several types of concept result : evol_elas, evol_noli, mode_meca,…. With each one corresponds a list of fields and a list of licit variables of access.
  In addition, with a concept result is also associated a certain number of complementary results (for example the generalized mass or the modal factors of participation in a modal calculation (concept result of type mode_meca)), different for each value from the variable of access. These complementary results are called parameters.
  Taking into account the structure of data result, one understands easily that the possibilities of impression which one lays out are those of the fields of size, supplemented by specific possibilities:
  - information on the structure of data (for example: list of the variables of access, list of the actually calculated fields,…),
  - direct access to a field of size (for example, impression of the field of displacements at the moment $t=15.$),
  - impression of the values of parameters.

  The operand RESULT allows to print the fields contained in a concept result. One can for example choose to print only certain fields (confer the following keyword: NOM_CHAM) on certain components of certain nodes (cf. NOEUD_CMP).

7 Extraction of a field of size

7.1 Operands TOUT_CHAM/NOM_CHAM

Cf document [U4.71.00].

7.1 Operands TOUT_ORDRE/NUME_ORDRE/LIST_ORDRE/NUME_MODE/INST/LIST_INST/FREQ/LIST_FREQ/NOEUD_CMP/NOM_CAS/ANGLE/PRECISION/CRITERION

Cf document [U4.71.00].

8 Parameters

8.1 Operand TOUT_PARA

This keyword indicates if one wants or not to print all the values of the parameters attached to the concept of the type `result`.

Example:

```
IMPR_RESU (RESU= (_F (RESULTAT= resu,
       TOUT_PARA= 'YES', NUME_ORDRE= 3)))
```

One prints in the file `RESULT` values of all the parameters but only for the third step of calculation of a concept `result` of type `mode_meca` :

Impression of the parameters of the concept `resu` for the sequence number 3:

```
NOM_MODE         3
FREQ             1.09787E+4
OMEGA2           4.75843E+09
AMOR_REDUIT      0.00000E+00
ERROR            2.76853E-10
MASS_GENE        1.50009E-07
RIGI_GENE        7.13806E+02
........
........
NORMALIZES              SANS_CMP: LAGR
```

8.2 Operand NOM_PARA

This keyword makes it possible to choose a list of reference symbols of parameters among the whole of the possible ones.

Keywords `TOUT_PARA` and `NOM_PARA` cannot be used simultaneously.

Example:

```
IMPR_RESU (RESU= (_F ( RESULTAT= resu,
                 NOM_PARA= ('STANDARD'), NUME_ORDRE= 3)))
```

One prints in the file `RESULT` the value of the parameter of name '`NORMALIZES`'but only for the third step of calculation D' a concept `result` of type `mode_meca` : Impression of the parameters of the concept `resu` for the sequence number 3:

```
NORMALIZES            SANS_CMP: LAGR
```

8.3 Operand FORM_TABL

This keyword makes it possible to choose the format of impression of the values of the parameters. Maybe in the form of a table whose each line is limited to 80 characters (`FORM_TABL = 'YES'`), that is to say in the form of a table whose each line can reach 2000 characters (`FORM_TABL = 'EXCEL'`), that is to say in the form of a list (a parameter by line) (`FORM_TABL = 'NOT'`).

By defaults the parameters are printed in the form of a table.

Example :

```
1  FORM_TABL = 'YES'
     NUME_ORDRE   NUME_MODE        ITER_QR             ITER_BATHE
            FREQ                OMEGA2           AMOR_REDUIT
            MASS_GENE        RIGI_GENE        AMOR_GENE
            MASS_EFFE_DY     MASS_EFFE_DZ     FACT_PARTICI_DX
            FACT_PARTICI_DZ  MASS_EFFE_UN_DX  MASS_EFFE_UN_DY
             NORMALIZES                 METHOD

2  FORM_TABL = 'EXCEL'
NUME_ORDRE   NUME_MODE        ITER_QR             ITER_BATHE
```

9 Selection on the components

Another manner of reducing the volume of the impressions is to print only the values of certain components (for example that displacement following the axis `X` : component `DX`).
Note: the selection of the components does not function for the fields under-points.

9.1 Operand **TOUT_CMP**

This keyword makes it possible to indicate that one wishes to print all the components of the field.

9.2 Operand **NOM_CMP**

This keyword makes it possible to choose the list of the components of `cham_gd` or of all the fields of the concept result which one wishes to print.
Keywords `TOUT_CMP` and `NOM_CMP` cannot be used simultaneously.
These components are described in the specific documentation of the elements.

10 Selection of the topological entities

In order to reduce the volume of the impressions, it is sometimes necessary to print only part of the result. With this intention one can print a field with the nodes only in certain nodes, or a field by element that in certain elements.
Note: the selection of the topological entities does not function for the fields under-points.

10.1 Operand **ALL**

This keyword indicates that one wishes to print the field on all the structure (all the nodes for a field with the nodes, all elements for a field by element).

10.2 Operand **GROUP_NO**

This keyword makes it possible to indicate the list of the groups of nodes on which one wishes to print one `cham_no`. If this keyword is used in the case of the impression of one `cham_elem`, he is ignored, and it `cham_elem` is printed in all the meshs, specified in addition.

10.3 Operand **GROUP_MA**

This keyword makes it possible to indicate the list of the groups of meshs on which one wishes to print one `cham_elem`. For one `cham_no`, it makes it possible to indicate the list of the nodes, tops of the meshs to which one wishes to print it `cham_no`.

11 Selection on the values

It is also interesting to print only the extreme values of all or certain components, on whole or part of the structure.

11.1 Operand **VALE_MAX**

This keyword indicates that one wishes to print the maximum value of each component of the field, possibly in a given interval specified by the arguments BORNE_SUP and/or BORNE_INF.

11.2 Operand **VALE_MIN**

This keyword indicates that one wishes to print the minimal value of each component of the field, possibly in a given interval specified by the arguments BORNE_SUP and/or BORNE_INF.

11.3 Operand **BORNE_SUP**

This keyword indicates the upper limit of the interval on which one wishes to print the field. One prints all the values of the field lower than this upper limit.

11.4 Operand **BORNE_INF**

This keyword indicates the lower limit of the interval on which one wishes to print the field. One prints the values of the field higher on this lower terminal.

12 Impression of the coordinates: operand **IMPR_COOR**

This keyword makes it possible to specify if one wishes to print the coordinates of a node at the time of the writing of one cham_no with the format ‘RESULT’.

13 Operand **FORMAT_R**

```
◊  FORMAT_R  =  /  FORMS,
                /  ‘1PE21.14’            [DEFECT]
```

Allows to specify the format of writing of the actual values to the format ‘RESULT’.

Note: the choice of the format of writing of the actual values does not function for the fields under-points.

14 Operand **PART**

```
◊  PART   =  /  ‘REAL’,
             /  ‘IMAG’,
             /  ‘MODULE’,
             /  ‘PHASE’,
```

Allows to specify the format of impression of the compelxes. IF nothing is specified, the complex will be printed. If not one can print the real part (PARTIE=' REEL') , the part complexes (PARTIE=' IMAG'), the module (PARTIE=' MODULE') or the phase (PARTIE=' PHASE').

15 Notice concerning the fields under-points

In *Aster*, there exists `cham_elem` with under points (as well for `cham_elem` with the nodes that for `cham_elem` at the points of Gauss). With the format '`RESULT`', `IMPR_RESU` the impression allows of `cham_elem` with under points.
If same `cham_elem` have under points on certain meshs and not on others, the impression of `cham_elem` will be done in two times:

- initially impression of `cham_elem` on all the meshs on which there is not under points,
- then impression of `cham_elem` on all the meshs on which there is under points.

A specific format is used for each one of these impressions. The impression of the fields under-points is "basic": one systematically prints all the geometrical entities, all the components and all the under-points (a line not under-point). The real numbers are always printed with 10 significant figures.

A trick to print more significant figures for a field under-points consists in calling the order `POST_CHAMP` to extract the field on a particular under-point then to print the new structure of data with the keyword `FORMAT_R`.

16 Impression of the grid: operands GRID / MODEL

This argument makes it possible to print the grid with `FORMAT = 'ASTER'` or `'RESULT'`. One then finds the format used for `LIRE_MAILLAGE` [U4.21.01] and defined in [U3.01.00].
By default, one prints all the grid.
The concept of the type `model` is optional in the majority of the impressions. However, it can be used in the case of the impression of a grid to the formats `'ASTER'` or `'RESULT'`, in order to print only the part of the grid whose meshs are affected in the model.
If format is chosen `'ASTER'`, the grid is written in a file of the type `.mail` who can then be read again by `LIRE_MAILLAGE`.
If format is chosen `'RESULT'`, the grid is written, by default, in the file `.resu` and cannot be read again by `LIRE_MAILLAGE` such as it is because the file `.resu` contains titles in addition, under titles and possibly of other information.

17 Operand SOUS_TITRE

This argument makes it possible to print under title of comment, for more details to consult the document [U4.03.01].
`SOUS_TITRE` by default is:

- for one `cham_gd`
  - for one `cham_no`
    `SOUS_TITRE = ('FIELD WITH THE NODES')`
  - for one `cham_elem`
    `SOUS_TITRE = ('FIELD BY ELEMENT &LOC (cham_elem)' )`

    Example: "`FIELD BY ELEMENT AT THE POINTS OF GAUSS`"
- for one `result`
  - for one `cham_no`

```
SOUS_TITRE =
    (   'FIELD WITH the NODES OF REFERENCE SYMBOL &NOM_SYMB (cham_no
Result) &RL NUMBER D' 'ORDER &NUM_ORDRE (cham_no Result) &ACCES (cham_no
Result)')
```

Example of impression:

```
FIELD WITH THE NODES OF REFERENCE SYMBOL DEPL
SEQUENCE NUMBER 2 NUME_MODE 3
FREQ  5.52739E+00
```

  - for one `cham_elem`

```
    SOUS_TITRE =
        (  'FIELD BY ELEMENT &LOC (cham_elem Result) OF REFERENCE SYMBOL
&NOM_SYMB (cham_elem Result) &RL NUMBER D' 'ORDER &NUM_ORDRE (cham_elem
Result) &ACCES (cham_elem Result)')
```

Example of impression:

```
                 FIELD BY ELEMENT WITH THE NODES OF REFERENCE SYMBOL
EPSI_ELNO
                 SEQUENCE NUMBER 1
                 INST   0.00000E+00
```