

Procedure IMPR_RESU with the format 'MED'

1 Goal

To write the result of a calculation in a file with format MED. One describes the whole of the keywords of order IMPR_RESU [U4.91.01] concerning this format of exit only.

One can write with the choice in a file with format MED:

- a grid,
- fields with the nodes,
- fields with the elements.

At the time of the writing of the fields by elements at the points of Gauss, one also writes the localization of the elements of reference (coordinated and weight of the points of Gauss).

MED (Modeling and Data exchanges) is a neutral format of data developed by EDF R & D and the ECA for the data exchanges between computer codes. The data which one can to exchange according to this format are the grids and the fields of results to the nodes and by elements. Files MED are binary and portable files (being pressed on the library HDF5, Hierarchical Dated Format). The writing of results in a file MED allows any other reading, computer code interfaced with MED the results produced by *Code_Aster* via the order IMPR_RESU.

2 Syntax

```

IMPR_RESU      (
# Syntax of procedure IMPR_RESU to format 'MED'
◇ UNIT = links,
◇ FORMAT = / 'MED',                                [DEFECT]

◇ PROC0 = / 'YES',                                [DEFECT]
          / 'NOT',
◇ IMPR_NOM_VARI = / 'YES',                          [DEFECT]
                 / 'NOT',
◇ INFORMATION = / 1,                                [DEFECT]
                 / 2,

/ LMBOU = _F (
  ◆ | GRID = my,                                    [grid]
    | / RESULT = resu,                              [sd_resultat]

        ◇ / NOM_CHAM = l_nomsymb,                    [l_K16]

        ◇ / NUME_ORDRE = lordre,                    [l_I]
          / NUME_MODE = lmode,                       [l_I]
          / NOEUD_CMP = lnoecmp,                     [l_K16]
          / NOM_CAS = ncas,                           [l_K16]
          / ANGLE = langl,                            [l_K16]
          / / FREQ = lfreq,                           [l_R]
          / INST = linst,                             [l_R]
        ◇ | PRECISION = / prec,                      [R]
          / 1.0D-3,                                  [DEFECT]
          | CRITERION = / 'RELATIVE',                 [DEFECT]
          / 'ABSOLUTE',

        / CHAM_GD = chgd,                            [cham_gd]

        ◇ / NOM_CHAM_MED = l_nomcham,                [l_K64]
          / NOM_RESU_MED = l_nomresu,                 [K8]

        ◇ CARA_ELEM = carele,                        [cara_elem]

        ◇ PART = / 'REAL',
                 / 'IMAG',
◇ INFO_MAILLAGE = / 'YES'
                  / 'NOT'                            [DEFECT]
),

# To print some fields of "data":
/ CONCEPT = _F (
  / CHAM_MATER = chmat,                              [cham_mater]
  / CARA_ELEM = carele,                              [cara_elem]
  ◇ REPERE_LOCAL = / 'NOT',                          [DEFECT]
                  / 'ELEM',
                  / 'ELNO',

  # if REPERE_LOCAL = 'ELEM' or 'ELNO',
  ◆ MODEL = Mo                                      [model]

/ LOAD = charg,                                     [load]
)

```

3 Operands **FORMAT** and **UNIT**, **PROC0** and **INFORMATION**

3.1 Operand **FORMAT**

The operand **FORMAT** allows to specify the format of the file where to write the result. The format 'MED' mean with the procedure **IMPR_RESU** that the result must be written in a file with format MED. It is the format of writing by default.

3.2 Operand **UNIT**

Defines in which unit one writes the file med. By default, **UNIT** = 80 and corresponds to the unit by default of the type **rmed** in **astk**.

3.3 Operand **PROC0**

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.

3.4 Operand **INFORMATION**

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

4 Operand **IMPR_NOM_VARI**

This keyword is useful in the case of the internal variables. When it is used and that impression of a field **VARI_*** was asked, it is in fact a field **VARI_*_NOMME** who will be printed. This field will have components whose name will be based on the catalogue of the laws of behavior used in calculation. If two laws of behavior have common internal variables, those will be amalgamated in a single component.

5 Keyword factor **RESU**

This keyword factor makes it possible to specify the results and fields to print.

5.1 Operand **GRID**

If the result is a grid (operand **grid** [U4.91.01]), the data deferred in the file result to format MED are:

- | | |
|------------------------------------|---|
| • the list of the nodes | number, name, coordinated, |
| • the list of the meshes | number, name, type, name of the nodes, |
| • the list of the groups of nodes | number, name, many nodes, names of the nodes, |
| • the list of the groups of meshes | number, name, many meshes, names of the meshes. |

Foot-note:

In a file MED, there is partition of the nodes and the meshes according to the groups. A partition corresponds to a family MED. In a file MED, the groups are distributed within the families: families of nodes and families of elements are thus found there.

5.2 Operand **RESULT**

The operand **RESULT** allows to print in a file **MED**, fields contained in a concept **result**.

One writes in the file 'MESSAGE' following information:

- operand 'RESULT',
- operand 'NAME_CHAM',
- operand 'NUME_ORDRE',
- name of the field stored in file MED: concatenation of the three preceding operands.

If `INFO_MALLAGE = 'YES'`, more detailed information is printed in the file 'MESSAGE' at the time of the writing of the grid MED. One will be able for example to obtain the types of printed meshes, the names of the families MED which are created, etc.

5.3 Operand CARA_ELEM

The operand `CARA_ELEM` is used for the impression of the fields for under-points. When `CARA_ELEM` is provided, the fields under-points are printed by adding information in file MED allowing to position the under-points by taking account of the contained information in `sd_cara_elem` (thickness of a hull, angle of gimlet of a multifibre beam,...).

Note:

It is currently not possible to visualize fields at the under-points on elements PIPE in the module ParaVis of Salomé-Meca. One will be able for the moment to thus use operator IMPR_RESU_SP [U7.05.41] with this intention.

5.4 Operand CHAM_GD

The operand `CHAM_GD` allows to print in the file a structure of data of the type `cham_gd`. Concretely, one can thus print with this keyword a map, a field by elements or a field with the nodes.

5.5 Operand NOM_CHAM_MED

The operand `NOM_CHAM_MED` allows to define the name of field MED. It is a chain of 64 characters. This can be useful in particular when one wishes to print certain components of the field like several fields in same file MED (for example for the visualization of `SIRO_ELEM`).

5.6 Operand NOM_RESU_MED

The operand `NOM_RESU_MED` is an alternative to `NOM_CHAM_MED` concerning the terminology of fields MED. Its use will make it possible not to name fields MED explicitly more, which means that all the fields contained in the result will be printed. Each field name MED will be built to leave:

- character string provided to `NOM_RESU_MED` (chain of with more the 8 characters),
- reference symbol of the field Aster.

For example:

```
IMPR_RESU = (  
  FORMAT = 'MED',  
  RESU = _F ( RESULT = U,  
             NOM_RESU_MED = 'U_HAUT',  
             GROUP_MA = 'HIGH',  
             NUME_ORDRE = 1, )  
)
```

If the result U contains the fields `DEPL` and `SIEF_ELGA`, then the order above will produce fields MED:

- 'U_HAUT__DEPL',
- 'U_HAUT__SIEF_ELGA',

This can be useful in particular when one wishes to print in same file MED the same field on different groups of meshes.

5.7 Operand PART

It is not possible to write complex fields. This is why it is necessary to choose between the real part (PARTIE=' REEL ') and the part complexes (PARTIE=' IMAG ').

5.8 Operands NOM_CHAM / NUME_ORDRE / NUME_MODE / NOEUD_CMP / NOM_CAS / ANGLE / FREQ / INST / PRECISION / CRITERION / FILE

Cf document [U4.91.01].

6 Keyword factor CONCEPT

This keyword factor makes it possible to print in a displayable form the quantities affected by the user with the orders AFFE_MATERIAU , AFFE_CARA_ELEM and AFFE_CHAR_MECA .

That makes it possible to check that the rules of overload of the code lead to the assignments wished.

For example:

```
IMPR_RESU (FORMAT=' MED',  
          CONCEPT= (  
            _F (CHAM_MATER = CHAMPMAT),  
            _F (CARA_ELEM = CARA_ELE),  
            _F (LOAD = CHARG1),  
          ))
```

The fields contained in these structures of data are printed in two very different forms:

- The form “with” (easiest to interpret graphically): each component is separately printed like a real number. For example, the thickness of the elements of hull is displayable like a scalar field (scalar map in Salomé).
- The form “B”: one assigns to each mesh a whole code: 1,2,3,... N. The meshes affected by the same code have ALL then their identical components. The “definition” of the codes, i.e. the values of these components is printed in the file message. One can visualize the “codes” like a scalar field, which makes it possible “to see” the zones where “all is constant”.

Form “A” is used systematically for the format “MED”, except for the material field because this field contains the name of affected material on the meshes and this name is not a number.

Form “A” is not programmed with the format “RESULT”.

The form “B” is systematically used for the format “RESULT” and for the material field.

For the form “B”, the correspondence between the affected quantities and the code is given in the file .mess.

For example, for the material field:

```
IMPRESSION OF A FIELD OF CONCEPT: MATERIAL field  
NAME OF THE FIELD: CHAMPMAT_CHAMP_MAT  
CORRESPONDENCE VALUE <-> CONTAINED:  
VALUE = 1.  
X1 = MAT_1  
VALUE = 2.  
X1 = MAT_2  
VALUE = 3.  
X1 = MAT_3
```

6.1.1 Operand REPERE_LOCAL

If REPERE_LOCAL the value has 'ELEM', the 3 vectors constituting the local reference mark of each element are printed, to be able to be visualized with L' helps of ParaVis.

If REPERE_LOCAL the value has 'ELNO', the 3 vectors constituting the local reference mark of each element are recorded in the form of field with the nodes, to be able to be then used in the computer of ParaVis in combination with D' other fields with the nodes. In this case no other information coming from the concept CARA_ELEM is not recorded in the file.

When REPERE_LOCAL=' ELNO' it is thus possible to combine the local vectors with the components of the internal fields of efforts. That makes it possible to visualize the vectors efforts in 3D, like carrying out an animation of their evolution during moments of calculation. To carry out this action several elementary operations are to be realized in ParaVis:

- To open the file MED containing the reference marks:
on REPE -> Filter "ELNO Points" -> Filter "Merge blocks"
- To open the file MED containing the fields:
on EFGE_ELNO -> Filter "ELNO Points" -> Filter "Merge blocks"
- selection of the 2 "Merge blocks" then "Suspend Attributes"

In "Calculator" one has access to the vectors of REPE and with the components of the fields.

The vector effort (NR, Vy, Vz) is calculated in the following way:

$$Fint = CAREL_._REPLC_1 * xxxxxxxx EFGE_ELNO_N + \\ CAREL_._REPLC_2 * xxxxxxxx EFGE_ELNO_VY + \\ CAREL_._REPLC_3 * xxxxxxxx EFGE_ELNO_VZ$$

where 'xxxxxxx' indicate the name of the concept result, produced by Code_Aster.

The vector Moment (MT, My, Mz) is calculated in the following way:

$$Mint = CAREL_._REPLC_1 * xxxxxxxx EFGE_ELNO_MT + \\ CAREL_._REPLC_2 * xxxxxxxx EFGE_ELNO_MFY + \\ CAREL_._REPLC_3 * xxxxxxxx EFGE_ELNO_MFZ$$

where 'xxxxxxx' indicate the name of the concept result, produced by Code_Aster.

6.2 Fields being able to be visualized

CHAM_MATER:

Material field

CARA_ELEM:

General characteristics of the bars
Characteristics géom. bars
General characteristics of the beams
Characteristics géom. beams
Wire specifications
Characteristics of the curved beams
Characteristics of the "fluid" beams
Characteristics of the discrete elements K_*
Characteristics of the discrete elements M_*
Characteristics of the discrete elements A_*
Characteristics géom. hulls
Orientation of the elements 2D and 3D
Orientation of the hulls and the beams

LOAD :

Loading of GRAVITY
Loading of ROTATION
Loading of PRES_REP
Loading of voluminal forces in 3D
Loading of surface forces in 3D
Loading of linear forces in 3D
Loading of surface forces in 2D

Loading of linear forces in 2D
Loading of forces distributed for the hulls
Loading of PRE_EPSI
Loading of FORCE_ELEC
Loading of FLUX_THM_REP
Loading of IMPE_FACE
Loading of FONDE_FLUI

7 Example

```
IMPR_RESU = (  
  FORMAT = 'MED',  
  RESU = _F ( RESULT = REMEZERO,  
             NOM_CHAM = 'ERME_ELEM',  
             NUME_ORDRE = 3,)  
)
```

Execution of the order IMPR_RESU following posting in the file will cause 'MESSAGE' :

```
RESULT          : REMEZERO  
FIELD           : ERME_ELEM  
NUME_ORDRE      : 3  
==> NAME MED    : REMEZEROERME_ELEM
```

Example of use of NOM_CHAM_MED for the impression of SIRO_ELEM :

```
IMPR_RESU (FORMAT=' MED',  
          RESU= (  
    _F (RESULTAT=RESUNL,  
        NOM_CHAM= ('SIRO_ELEM',),  
        NOM_CHAM_MED= ('RESUNL_SIRO_ELEM_NORMAL'),  
        NOM_CMP= ('SIG_NX', 'SIG_NY', 'SIG_NZ', 'SIG_N'),  
        GROUP_MA=' PRES',),  
    _F (RESULTAT=RESUNL,  
        NOM_CHAM= ('SIRO_ELEM',),  
        NOM_CHAM_MED= ('RESUNL_SIRO_ELEM_TANGENT'),  
        NOM_CMP= ('SIG_TX', 'SIG_TY', 'SIG_TZ'),  
        GROUP_MA=' PRES',),  
    _F (RESULTAT=RESUNL,  
        NOM_CHAM= ('SIRO_ELEM',),  
        NOM_CHAM_MED= ('RESUNL_SIRO_ELEM_T1'),  
        NOM_CMP= ('SIG_T1X', 'SIG_T1Y', 'SIG_T1Z',  
'SIG_T1'),),  
        GROUP_MA=' PRES',),  
    _F (RESULTAT=RESUNL,  
        NOM_CHAM= ('SIRO_ELEM',),  
        NOM_CHAM_MED= ('RESUNL_SIRO_ELEM_T2'),  
        NOM_CMP= ('SIG_T2X', 'SIG_T2Y', 'SIG_T2Z',  
'SIG_T2'),),  
        GROUP_MA=' PRES',),  
    ),),
```