

---

## Procedure IMPR\_RESU with the format 'GMSH'

---

### 1 Goal

---

To write a field of size or a concept result with the format 'GMSH'.

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

- fields with the nodes (of displacements, temperatures, clean modes, static modes,...),
- fields by elements with the nodes (of constraints, generalized efforts, internal variables...).

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.

It is possible to select the topological entities (meshes or groups of meshes) on which one wants to print the results.

## 2 Syntax

```

IMPR_RESU      (
  ♦ MODEL = Mo,                                     [model]
  ♦ UNIT = / links,
              / 37                                 [DEFECT]
  ♦ PROC0 = / 'YES',                               [DEFECT]
              / 'NOT',
  ♦ FORMAT = / 'GMSH',

  ♦ RESU = _F (
    ♦ / GRID = my,                                  / [grid]
                                              / [skeleton]

    / CHAM_GD = ch_gd,
    / RESULT = 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_K16]
              / / 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',

    # Selection of the topological entities and the components
    ♦ / MESH = l_maille,                             [l_maille]
      / GROUP_MA = l_grma,                           [l_gr_maille]
    ♦ TYPE_CHAM = / 'SCALAR',                         [DEFECT]
      ♦ NOM_CMP = lnomcmp, [l_K8]
      / / 'VECT_3D',
      / 'TENS_2D',
      / 'VECT_2D',
      / 'TENS_3D',
      ♦ NOM_CMP = lnomcmp, [l_K8]
    ),
    ♦ PART = / 'REAL',
              / 'IMAG',

    ♦ VERSION = / 1.2,                               [DEFECT]
              / 1.0,                                 [R]
  ) ;

```

## 3 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.

### 3.1 Operand GRID

This operand makes it possible to choose the grid which one wishes to print. The simultaneous impression of the grid and fields is impossible

### 3.2 Selection of the data to be printed

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 a matrix 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 a 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.

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.

#### Caution

One cannot print in the same file it `GRID` and it `RESULTAT/CHAM_GD`, `GMSH` cannot read such a file. The order thus stops if the two keywords are present.

#### 3.2.1 Operand CHAM\_GD

For `CHAM_NO`, the components of the field to the nodes are burst in a vector field gathering the components `DX`, `DY`, `DZ` and in as many scalar fields there are other components.

**Example** : In the case of a field with the nodes of size `DEPL` and whose modeling is of standard beam, one writes:

- 1 vector field (gathering the whole of the components of translation),
- 3 scalar fields (a field per degree of rotation).

For `CHAM_ELEM`, one prints only the fields by elements with the nodes (`_ELNO_`). A field by elements with the nodes is burst in as many scalar fields there are components.

## Notice

When one writes a result with format GMSH, the title of the fields, which appears in GMSH, is made up by the name of the Aster result + the name of the field + the component.

Example: RESU\_DEPL\_DX : the component 'DX' field 'DEPL' result 'RESU'

If a derived field is written, one intercalates the name of the significant parameter by report to which one derives, between the name of the result and the name of the field.

Example: RESU\_PS1\_DEPL\_DX : the derivative compared to 'PS1' component 'DX' field 'DEPL' result 'RESU'.

### 3.2.2 Operand RESULT

The operand RESULT allows to print the fields contained in a concept result. One can for example choose to print only certain fields (cf the keyword: NOM\_CHAM).

The concept result is written field of size by field of size as described in [§4.1].

## 4 Choice of the fields to be printed

### 4.1 Operands TOUT\_CHAM/NOM\_CHAM

Cf document [U4.71.00].

### 4.2 Operands

TOUT\_ORDRE/NUME\_ORDRE/LIST\_ORDRE/NUME\_MODE/INST/LIST\_I  
NST/FREQ/LIST\_FREQ/NOEUD\_CMP/NOM\_CAS/ANGLE/PRECISION/C  
RITERION

Cf document [U4.71.00].

## 5 Selection of the geometrical zones and the components

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 or a field by element only in certain elements.

This selection which one will note "selection on topological entities" is possible with the format 'GMSH'.

### 5.1 Operand MESH

This keyword makes it possible to indicate the list of the meshes 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 meshes to which one wishes to print it cham\_no.

### 5.2 Operand GROUP\_MA

This operand makes it possible to indicate the list of the groups of meshes 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 meshes to which one wishes to print it cham\_no.

### 5.3 Operand TYPE\_CHAM

GMSH because it is able to visualize vectors and tensors; it is thus possible to visualize the other things that each component independently. This operand makes it possible to choose it type of field to

be printed ('VECT\_2D', 'VECT\_3D', 'SCALAR', 'TENS\_2D' or 'TENS\_3D'). The value by default is 'SCALAR'.

## 5.4 Operand NOM\_CMP

This operand makes it possible to choose the list of the components which one wishes to print according to the type of selected field.

If one chose to print scalar fields it is not necessary to inform NOM\_CMP, in this case, each component will be printed independently from/to each other.

## 5.5 Operand PART

One cannot print complex fields with format GMSH, it is thus necessary, for such fields, to select the part REALITY or IMAGINAIRE to be printed.

## 6 Operand VERSION

Meshes accepted by GMSH (in postprocessing) are, since the version 1.35, the meshes of the type POI1, SEG2, TRIA3, QUAD4, TETRA4, HEXA8, PENTA6, PYRAM5, which corresponds to:

VERSION = '1.2' who is the value by default

If one requires to visualize the fields with an old version of GMSH or if one wishes to cut out all the meshes in POI1, SEG2, TRIA3 and TETRA4, one uses VERSION = '1.0'.

To be able to visualize the sights with GMSH, the grid is modified to have only meshes GMSH. Currently, meshes PYRAM13 are not treated.

- SEG3 is broken up into 2 SEG2,
- SEG4 is broken up into 3 SEG2,
- TRIA6 is broken up into 4 TRIA3,
- TRIA7 is broken up into 4 TRIA3,
- QUAD4 is broken up into 2 TRIA3 (cut out only if VERSION=' 1.0 '),
- QUAD8 is broken up into 6 TRIA3,
- QUAD9 is broken up into 6 TRIA3,
- TETRA10 is broken up into 8 TETRA4,
- PENTA6 is broken up into 3 TETRA4 (cut out only if VERSION=' 1.0 '),
- PENTA15 is broken up into 16 TETRA4,
- PYRAM5 is broken up into 2 TETRA4 (cut out only if VERSION=' 1.0 '),
- HEXA8 is broken up into 6 TETRA4 (cut out only if VERSION=' 1.0 '),
- HEXA20 is broken up of 24 TETRA4,
- HEXA27 is broken up into 48 TETRA4.

### Notice 1

*TRIA7 (respectively QUAD9) is cut out like TRIA6 (resp. QUAD8) because the central node does not carry a degree of freedom of displacement in modeling COQUES\_3D.*

### Notice 2

*With the recent versions of GMSH (>1.60), one could avoid cutting out the elements. The choice is not available in Code\_Aster because it is possible to read format MED in GMSH since version 2.0.*

## 7 Operand UNIT

UNIT allows to choose on which file the results will be written. By default, UNIT = 37 with the format 'GMSH'.

One can thus print results on different files with each order IMPR\_RESU.

The natural suffix of the files of post treatment GMSH is .pos and the usual logical unit (by default in astk) is 37.

## 8 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.