
Procedure TEST_RESU

1 Drank

To compare a value extracted from a data structure with a value of reference provided by the user.

This command makes it possible to test a numerical value: integer, reality or complex extracted from a concept already calculated. Today, one can test a component of a `cham_no` or a `cham_elem`, a component of a field extracted from one `result`, a parameter of one `result`, a "total" value extracted from a field or the contents from an unspecified object from a concept user.

The procedure writes a conventional message then:

- "OK" (if it is good),
- "NOOK" (if not).

follow-up of the found value, the value of reference and percentage of error.

Systematically, one checks a value of non regression, and when that is possible, an analytical value of reference, coming from an external source or another computation with Code_Aster.

Commands `TEST_FONCTION` [U4.92.02] and `TEST_TABLE` [U4.92.03] make it possible to test the values extracted from the functions and the arrays.

2 Syntax

```
TEST_RESU (
  ♦ | CHAM_NO= (_F ( ♦ CHAM_GD =chno , [cham_no]
                    / TYPE_TEST = "SOMM_ABS",
                              / "SOMM",
                              / "MAX",
                              / "MIN",
                    ◇ NOM_CMP = ncmp , [K8]
                    / ♦ /NOEUD =no , [node]
                      /GROUP_NO =grno , [group_no]
                    ♦NOM_CMP =nomcmp , [K8]
                    ◇ LEGENDE = legend , [K16]
                    to #Voir definition of the value of reference
                    ),)
  | CHAM_ELEM= (_F ( ♦CHAM_GD =chel , [cham_elem]
                    /TYPE_TEST = "SOMM_ABS",
                              / "SOMM",
                              / "MAX",
                              / "MIN",
                    ◇NOM_CMP = ncmp , [K8]
                    /♦MAILLE =ma , [mesh]
                    ♦ /POINT = nupoint , [I]
                    /NOEUD = No , [node]
                    /GROUP_NO = grno , [group_no]
                    ◇SOUS_POINT = nusp ,
                    ♦NOM_CMP = nomcmp , [K8]
                    ◇ LEGENDE = legend , [K16]
                    to #Voir definition of the value of reference
                    ),),
  | CARTE= (_F ( ♦CHAM_GD =chel , [cham_elem]
                ♦NOM_CMP = nomcmp , [K8]
                ♦MAILLE =ma , [mesh]
                ◇ LEGENDE = legend , [K16]
                to #Voir definition of the value of reference
                ),),
```

```
| RESU = (_F ( ♦RESULTAT =res ,
[resultat_sdaster]

♦/NUMÉRIQUE_ORDRE = nuor, [I]
/NUME_MODE = numo, [I]
/INST = inst, [R]
/FREQ = freq, [R]
/NOEUD_CMP = (node, cmp), [l_Kn]
/NOM_CAS = nocas, [kN]
/ANGLE =  $\alpha$ , [R]

♦/PARA = para, [K16]
/NOM_CHAM = nosymb, [K16]
    ◊NOM_CMP = ncmp, [K8]
    / TYPE_TEST = "SOMM_ABS",
    / "SOMM",
    / "MAX",
    / "MIN",
    ◊ NOM_CMP = ncmp, [K8]
    / ♦ /NOEUD = No, [node]
    /GROUP_NO = grno, [group_no]
    /♦MAILLE = my, [mesh]
        ♦ /POINT = nupoint, [I]
        /NOEUD = No, [node]
        /GROUP_NO = grno, [group_no]
        ◊SOUS_POINT = nusp,
        ♦NOM_CMP = nomcmp, [K8]
to #Voir definition of the value of reference
),),
| GENE = (_F ( ♦ RESU_GENE =res ,
[VECT_ASSE_GENE]
    NUME_CMP_GENE = ncmp, [I]
    ~ RESU_GENE =res ,
[MODE_GENE]
    ♦/PARA = para, [K16]
    /NOM_CHAM = nosymb, [K16]
    NUME_CMP_GENE = ncmp, [I]
    ♦ /NUME_ORDRE = nuor, [I]
    /NUME_MODE = numo, [I]
    /FREQ = freq, [R]
    RESU_GENE =res ,
[HARM_GENE ]
    NOM_CHAM = nosymb, [K16]
    NUME_CMP_GENE = ncmp, [I]
    ♦ /NUME_ORDRE = nuor, [I]
    /FREQ = freq, [R]
    RESU_GENE =res ,
[TRAN_GENE]
    NOM_CHAM = nosymb, [K16]
    NUME_CMP_GENE = ncmp, [I]
    ♦ /NUME_ORDRE = nuor, [I]
    /INST = inst, [R]
```

to #Voir definition of the value of reference
) ,) ,

```
| OBJET = (_F (
              ◆NOM=nomobj , [K24]
              to #Voir definition of the value of reference
              ),),
| TEST_NAN = "NON" , [DEFAULT]
              / "YES" ,
)

# Definition of the value of reference

◆ /VALE_CALC = val , [R]
  /VALE_CALC_C = val , [C]
  /VALE_CALC_I = val , [I]
  /VALE_CALC_K = val , [K*]

◇ LEGENDE = legend, [K16]

◇VALE_ABS = "NON" , [DEFAULT]
            / "YES" ,

◇/ | TOLE_MACHINE=/1.0D-6 ,
[DEFAULT]

| CRITERE = /prec , [R]
            "RELATIF" ,
[DEFAULT]
            / "ABSOLU" ,

# only for RESU or GENE
/ | TOLE_MACHINE=/ (prec1, prec2), [1_R]
  | CRITERE = (crit1, crit2), [1_Kn]

# safe in TEST_FICHER
◇REFERENCE = "ANALYTIQUE",
            / "SOURCE_EXTERNE",
            / "AUTRE_ASTER",

If REFERENCE is indicated
◆ /VALE_REFE = val , [R]
  /VALE_REFE_C = val , [C]
  /VALE_REFE_I = val , [I]
◇PRECISION =/1.0D-3 ,
[DEFAULT]
            /prec , [R]
```

Notices 1

the definition of the value of reference is common to commands TEST_RESU, TEST_TABLE, TEST_FONCTION and TEST_FICHER. All the key words are not available in all the commands, to see the comments.

The whole type (_I) does not exist in TEST_FONCTION and TEST_FICHER. The type complexes (_C) does not exist in TEST_FICHER and for OBJET. The type chains (_K) exists only in TEST_FICHER.

VALE_ABS does not exist for CARD and TEST_FICHER.

Notice 2

key keys CHAM_NO, CHAM_ELEM and CARD makes it possible to test the cham_no, the cham_elem and the card. Key word RESU is reserved for the concepts of the type result.

3 General information

This command makes it possible to test a scalar numerical value recovered in a concept of the `cham_no` type, `cham_elem`, `card` or `result`, compared to a value of non regression and, when it is possible, compared to an analytical value of reference, coming from an external source or another computation with Code_Aster.

Three types of numerical values can be tested:

- a component of a field (`cham_no`, `cham_elem`, `card` or field which is part of one `result`),
- a parameter contained in a concept of `result`,
- a global value of a field [§4.44.4].

To test a component of field, it is necessary to choose a field [§4.14.1] then to choose a component [§4.34.3].

To test a parameter, it is necessary to choose a sequence number [§4.1.44.1.4] and to choose the name of the parameter.

The expected numerical value (real, complex or whole) is provided in accordance with [§4.64.6].

Notice concerning the tests of non regression

One systematically makes a test of non regression compared to a computed value previously. The tolerance associated with this test (`TOLE_MACHINE`) must be very weak and should not be higher than the value by default. In particular, the value should be the same one (on at least 8 decimals) on all the platforms. Moreover, this computed value should change only when algorithm is modified, corrected. Any other variation owes alerted the developer on the reliability of the programming.

Remarks concerning the tests in “generalized” data structures:

One can test the generalized components (displacements, velocities or accelerations of a transient in modal space). It is advisable nevertheless to be circumspect with this kind of test. Indeed the value of a generalized component depends entirely on the norm of the mode. However that - C_i is given in an arbitrary way. Thus without preliminary standardization of the norms, the value of a generalized quantity is arbitrary. Lastly, there is no possibility in Code_Aster of fixing the direction of a mode. For a multiple mode, that wants to say that, even once the normalized modes, a generalized quantity can take an unspecified value. In the case of a simple mode, it can be directed in a direction or the opposite direction. One then obtains a generalized value or his opposite.

4 Operands

4.1 Selection of a field

In order to test a field which can be an isolated field (`cham_no` or `cham_elem` or `card`), or a field extracts from one result, or will use the key word factors: `CHAM_NO`, `CHAM_ELEM`, `CARD` or `RESU`.

4.1.1 Fields “isolated”

◆`CHAM_GD` = field

Name from the `cham_no`, `cham_elem` or `card` which one wants to extract a value.

4.1.2 Operand `RESULTAT`

◆`RESULTAT` = LMBO

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Name of the treated result concept.

4.1.3 Operand NOM_CHAM

◆/NOM_CHAM = nosymb

Symbolic name of the field to be selected.

4.1.4 Selection of a sequence number

◆/NUMÉRIQUE_ORDRE = nuor,

Sequence number of the field (or parameter) required.

```
/NUME_MODE = numo,  
/INST      = inst,  
/FREQ      = freq,  
/NOEUD_CMP = (node, cmp),  
/NOM_CAS   = nocas,  
/ANGLE     =  $\alpha$ ,
```

These keywords make it possible to identify a sequence number in one result [U4.71.00].

They are called “variables of access”.

They all are not valid for all the types of result.

When the access is not made an actual value (ANGLE, FREQ, INST) the value given should not be ambiguous (cf [§4.64.6]).

4.2 Selection of a parameter as a result

selecting a parameter in one result, it is necessary to specify the desired sequence number [§4.1.44.1.4] and to give the name of the parameter.

◆PARA = para

Name of the sought parameter. This name is attached to the type of the treated result concept.

4.3 Selection of a component of a field

the access to a quantity is done for a cham_no by:

- the name of the node which carries this quantity.

The access to a quantity is done for a cham_elem by:

- the name of the mesh which supports the element,
- something which specifies:
 - that is to say the name of a node of this mesh for the cham_elem “with the nodes” (ELNO).
 - that is to say the number of the Gauss point for the cham_elem “with Gauss points” (ELGA).

The access to a quantity is done for a card by:

- the name of the mesh,

◆MAILLE = my

Name of the mesh where one wants to test the cham_elem or the card.

◆/NOEUD = No

Name of the node which one wants to check a component.

```
/GROUP_NO = grno
```

to facilitate the use of this command, one can replace the key word by GROUP_NO NOEUD .

In this case, it is necessary that the group is reduced to only one node .

```
/POINT = nupoint
```

the integer `nupoint` specifies the number of the Gauss point which one wants to test the value (case of the `cham_elem` "to Gauss points").

```
◇SOUS_POINT = nusp
```

the integer `nusp` specifies the number of the subpoint of which one wishes to obtain the value (case of the `cham_elem` at subpoints, used by the structural elements: beam, pipes, shells).

In the case of the plates and of the multi-layer shells, the number of the subpoint corresponds to the level in all the layers. Each layer is described by a lower, average and higher skin. By convention, for N layers, this number varies between 1 and $3N$ where the first point is at the level of the skin lower of the first layer and $3N$ the ème not than the level of the higher skin of the last layer (cf [R3.07.03] and [R3.07.04] for the numeration of the layers).

In the case of the multifibre beams, this integer is the number of the fiber whose classification is described in documentation [U4.26.01] and [R3.08.08].

In the case of the pipes, it is necessary to refer to the description made in the document [R3.08.06].

◆ `NOM_CMP = ncmp`

Name of the component which one wants to test [U2.01.04].

4.4 To test a field "overall"

Once a selected field [§4.14.1], one can test a quantity calculated overall on all the field. For that, `NOM_CMP` should not be indicated to make it possible to take into account all the components of the field.

```
/TYPE_TEST = "SOMM_ABS"
```

the sum of the absolute values of the components of the field.

```
/TYPE_TEST = "SOMM"
```

the sum of the values of the components of the field.

```
/TYPE_TEST = "MAX"
```

the maximum of the values of the components of the field.

```
/TYPE_TEST = "MIN"
```

minimum of the values of the components of the field.

4.5 A JEVEUX object to test the contents of

This functionality is reserved for the developers of the Code. To use it, it is necessary to know the names of the JEVEUX objects composing the concepts of the user. It is intended to check non regression data structures other than `RESULTAT`, `FIELDS`, `ARRAY` and `FONCTION`.

4.5.1 Operand `NOM`

```
NOM = nomobj
```

Name of the `jeveux` object which one wants to test.

Note:

The purpose of this kind of test ($NOM + VALE_CALC/_I$) is to test a whole vector overall. The "sum" which is tested is unfortunately a bad "check sum" of the object: a permutation within the vector does not change this sum. A more careful test consists in printing the object in a file ($IMPR_CO$) then to test the contents of this file with a truth "check sum" ($TEST_FICHER$).

4.5.2 Operands CRITERE and PRECISION

See below [§4.64.6].

4.6 Definition of the value of non regression and reference

One systematically makes a test of non regression compared to a value previously calculated, with a very weak tolerance: key words $VALE_CALC$ and $TOLE_MACHINE$.

As often as possible, one adds a test compared to a value of reference compared to an analytical solution, a value obtained of an external source or another modelization: key words $REFERENCE$, $VALE_REFE$, accuracy.

◆ $/VALE_CALC$ = val

Actual value of non regression. It is the computed value by Code_Aster.

$/VALE_CALC_C$ = val

Value complexes the non regression one.

$/VALE_CALC_I$ = val

whole Value of non regression.

$/VALE_CALC_K$ = val

Character string of non regression. Only by $TEST_FICHER$.

◇ $TOLE_MACHINE$

Accuracy requested (by default 1.D-6) to accept the computed value compared to the value of non regression ($VALE_CALC$).

◇ $VALE_ABS$

= "NON" the value of reference and the computed value by Aster are compared such as they are.

= "YES" the value of reference and the computed value by Aster are compared in absolute values.

◇ $CRITERE$ =

Standard of test to be carried out. Applies to the test of non regression and the test compared to a value of reference if necessary.

If v are the extracted value, the test will carry for:

- "RELATIF" on: $|val - v| \leq prec \cdot |val|$
- "ABSOLU" on: $|val - v| \leq prec$

◇ $REFERENCE$ =

$/$ "ANALYTIQUE" : the provided value of reference is "analytical"

$/$ "SOURCE_EXTERNE" : the provided value of reference comes from a program other than Code_Aster , one bibliographical reference, a measurement, etc

$/$ "AUTRE_ASTER" : the provided value of reference is that obtained by another computation with Code_Aster (another command, another

modelization, computation option,...)

It is the presence of the key word REFERENCE which indicates that one has an external reference and conditions the presence of key words VALE_REFE [_I/_C] and PRECISE DETAILS.

◆VALE_REFE , VALE_REFE_C, VALE_REFE_I

Similar to key words VALE_CALC above. They define the value obtained by the external source.

◇PRECISION =

Accuracy requested (by default 1.D-3) to accept the computed value compared to the value of reference (VALE_REFE).

Note:

When the definition of the sequence number of a RESULTAT is done by a real variable of access (FREQ, INST, ANGLE), it is not necessary that there is ambiguity on this sequence number. For that the user defines a small interval around the value requested thanks to key keys CRITERE and TOLE_MACHINE.

In this case ("real" access) key keys CRITERE and TOLE_MACHINE will thus expect 2 values each one: (crit1, crit2) and (pre1, prec2).

crit1 and pre1 relate to the value of non regression.

crit2 and prec2 make it possible to choose the interval of search of the sequence number.

The default values of crit1 and pre1 are "RELATIF" and 1.D-6.

The default values of crit2 and prec2 are "RELATIF" and 1.D-3.

One cannot explicitly define crit2 and prec2 without defining crit1 and pre1.

4.7 Addition of one specificity to the value tested

◇ LEGENDE =

Character string of to more the 16 characters describing the test carried out.
The user thus has the possibility of commenting on his test.

4.8 Key word TEST_NAN

| TEST_NAN = "NON" , [DEFAULT]
/ "YES" ,

This key word is used to validate the operation of NaN (Not-a-Number) of Code_Aster. This key word is to be used only in optics to carry out tests. Its use must cause a fatal error in FPE (Floating Not Exception) in debug mode in TEST_RESU.

5 Examples

```
TEST_RESU
  ( CHAM_NO = _F      ( CHAM_GD = F,
                      NOEUD   = "N2",
                      NOM_CMP  = "DX",
                      REFERENCE = "ANALYTIQUE",
                      VALE_CALC = 1.1999996845E-5,
                      VALE_REFE = 1.2E-5,
                      accuracy = 1D-4 ,),

  CARD =_F ( CHAM_GD = CART1,
            MESH   = "m3",
            NOM_CMP = "X4",
            REFERENCE = "AUTRE_ASTER",
            VALE_CALC_I = 3,
            VALE_REFE_I = 3,),

  CHAM_ELEM= (_F      ( CHAM_GD = SIGGA,
                      MESH   = "m3",
                      POINT  = 3,
                      NOM_CMP = "SIXX",
                      VALE_CALC = 3.4E6,      ),

  _F ( CHAM_GD = SIGNO,
      MESH   = "M5",
      NOEUD  = "N1",
      NOM_CMP = "SIXX",
      VALE_CALC = 3.4E6 , , ),),)

TEST_RESU
  ( RESU = (_F      ( RESULTAT = evolth,
                    NOM_CHAM = "FLUX_ELGA",
                    TYPE_TEST = "MAX",
                    REFERENCE = "ANALYTIQUE",
                    VALE_CALC = 154.35000201404,
                    VALE_REFE = 154.35),

  OBJET =_F ( NOM = "CH1.CHME.LIGREL.LIEL ", VALE_CALC_I =
            1278484, ), )
```