Code_Aster

Version default

Titre : Procédure TEST_COMPOR
Responsable : HABOUSSA David

Date : 01/06/2016 Page : 1/9
Clé : U4.92.06   Révision   :
41f9ef5226e6

Procedure `TEST_COMPOR`

1   Goal

This macro-order makes it possible to test the mechanical laws of behavior from the point of view of the robustness and reliability:

- in the mechanical case, the test implements a simulation of a way of multiaxial loading on a material point, with an aim of checking the robustness of the digital integration of the behaviors tested, of their insensitivity compared to a change of units, invariance of the result compared to a total rotation applied to the problem, the accuracy of the tangent matrix;
- in the case of variables of orders (temperature,…) this test checks the good taking into account of the variables of order whose the coefficients depend on the model, as well as terms of deformation who result from it.

Code_Aster

Titre : Procédure TEST_COMPOR
Responsable : HABOUSSA David

Version
default

Date : 01/06/2016 Page : 2/9
Clé : U4.92.06   Révision   :
41f9ef5226e6

2   Syntax

```
TEST_COMPOR  (
    ◊   OPTION   = / 'THER',                            [DEFECT]
                   / 'MECA',
    ♦   BEHAVIOR    =_F (see the document [U4.51.11] ),

    ◊   NEWTON         =_F (  to see the document [U4.51.03]),

    ◊ CONVERGENCE      = _F (
       /RESI_GLOB_RELA   = 1.E-6,                      [DEFECT]
       /|RESI_GLOB_MAXI  = resmax,                     [R]
        | RESI_GLOB_RELA     =  resrel,                     [R]
        ITER_GLOB_MAXI    = /10,                       [DEFECT]
                           /maglob,                    [I]
                         ),
    if OPTION = 'THER'

    ♦   MATER        =  to subdue,                           [to
subdue]
    ♦   LIST_MATER   =  to subdue,
  [l_mater]
    ◊   ALPHA        =  alpha,                      [function]
    ◊   YOUNG        =  Young,                       [function]
    ◊   TEMP_INIT    =  temp_init,                   [R]
    ◊   TEMP_FIN     =  temp_fin,                    [R]
    ◊   NB_VARI      =  nb_vari,                     [I]
    ◊   SUPPORT       = /'NOT'                  [DEFECT]
                       /'ELEMENT'
    ◊   INST_FIN     =  temp_fin,                    [R]
    ◊   VARI_TEST    =  vari_test,                   [KN]
    ◊   D_SIGM_EPSI  =  d_sigm_epsi,                 [function]
    ◊   C_PRAG       =  c_prag,                      [function]

    if OPTION = 'MECA'
    ♦   LIST_MATER   =  to subdue,
    ♦   FISH      =  fish,                           [R]
    ◊   YOUNG        =  Young,                        [R]
    ◊   LIST_NPAS    =  list_npas,                    [l_I]
    ◊   LIST_TOLE    =  list_tole,                    [l_R]
    ◊   PREC_ZERO    =  prec_zero,                    [l_R]
    ◊   VARI_TEST    =  vari_test,                    [KN]
    ◊   SUPPORT       =  /'NOT'                       [DEFECT]
                        /'ELEMENT'
    ◊   MODELING  =  /'3D'                        [DEFECT]
                      /'C_PLAN'
    ◊   SOLID MASS             = /'ANGL_REP'                   [R]
                            /'ANGL_EULER'               [R]
    ◊   ANGLE       = angz,                           [R]
    ◊   VERI_MATR_OPTION =_F (
        ◊ VALE_PERT_RELA   =                          [R]
        ◊ PRECISION        =                          [R]
        ◊ PREC_ZERO        =                          [R]
          ),

    ◊   INFORMATION =    / 1,
    [DEFECT]
```

Titre : Procédure TEST_COMPOR
Responsable : HABOUSSA David

Code_Aster

Version default

Date : 01/06/2016 Page : 3/9
Clé : U4.92.06 Révision :
41f9ef5226e6

/   2,

)

Code_Aster

Titre : Procédure TEST_COMPOR
Responsable : HABOUSSA David

Version default

Date : 01/06/2016 Page : 4/9
Clé : U4.92.06   Révision :
41f9ef5226e6

3   Operands

3.1   Words-keys COMPORTEMENT/NEWTON

The syntax of these keywords is described in the document [U4.51.03] and [U4.51.11].

3.2   Keyword CONVERGENCE

◊   CONVERGENCE =_F ()

If none of the two operands following is present, then all occurs like if: RESI_GLOB_RELA = 1.E-6.

3.2.1   Operand RESI_GLOB_RELA/RESI_GLOB_MAXI

◊ |RESI_GLOB_RELA = resrel ,   [R]

The algorithm continues the total iterations as long as:

$$\max_{i=1,\dots,nbddl} |F_i^n| > \text{resrel} . \max |L|$$

where F^n is the residue of the iteration n and L the vector of the imposed loading and the reactions of supports (cf [R5.03.01] for more details).

When the loading and the reactions of support become worthless, i.e. when L is null (for example in the case of a total discharge), one tries to pass from the relative convergence criteria RESI_GLOB_RELA with the absolute convergence criteria RESI_GLOB_MAXI. This operation is transparent for the user (message of alarm emitted in the file .mess). When the vector L becomes again different from zero, one passes by again automatically with the relative convergence criteria RESI_GLOB_RELA.
However, this mechanism of swing cannot function with the first step of time. Indeed, to find a value of RESI_GLOB_MAXI reasonable in an automatic way (since the user did not inform it), one needs to have had at least a step converged on a mode RESI_GLOB_RELA . Consequently, if the loading is null as of the first moment, calculation stops. The user must already then check that the null loading is normal from the point of view of the modeling which it carries out, and if such is the case, to find another convergence criteria (RESI_GLOB_MAXI for example).

If this operand is absent, the test is carried out with the value by default, except if RESI_GLOB_MAXI is present.

◊ |RESI_GLOB_MAXI = resmax ,   [R]

The algorithm continues the total iterations as long as:

$$\max_{i=1,\dots,nbddl} |F_i^n| > \text{resmax}$$

where F^n is the residue of the iteration n (Cf [R5.03.01] for more details). If this operand is absent, the test is not carried out.

If RESI_GLOB_RELA and RESI_GLOB_MAXI both are present, the two tests are carried out.

3.2.2   Operand ITER_GLOB_MAXI

◊   ITER_GLOB_MAXI =   /10   [DEFECT]
                       /maglob

Maximum iteration count carried out to solve the total problem at every moment (10 by defaults).

3.3   OPTION=' THER'

Code_Aster

Titre : Procédure TEST_COMPOR
Responsable : HABOUSSA David

Version default

Date : 01/06/2016 Page : 5/9
Clé : U4.92.06   Révision   :
41f9ef5226e6

Thermomechanical test allowing of to validate the taking into account of the temperature variation in the laws of behavior (cf. *V6.07.108)*. These tests make it possible to check the two following points:

- Thermal dilation is well calculated (with taking into account of the variation of thermal dilation with the temperature)
- The variation of the coefficients material with the temperature is correct, in particular in the incremental resolution of the behavior.

It is about a double simulation, the first in thermomechanics, the second in pure mechanics. The first will be validated in comparison with the second, by supposing of course that the behavior tested provides a correct solution in pure mechanics.

The first simulation (solution which one seeks to validate) consists in applying a temperature variation to a material point, by blocking for example the deformations according to x : $\varepsilon_{xx} = 0$. The imposed temperature is increasing linearly according to time.

The second simulation (which must be equivalent to the first) consists in applying to the same material point a deformation imposed according to x : $\varepsilon_{xx} = -\varepsilon^{th} = -\alpha(T)(T - T_{ref})$, in pure mechanics. Indeed, for any behavior (while supposing the additive decomposition of the deformations):

$$\sigma_{xx} = E(T)(\varepsilon_{xx} - \varepsilon^{th} - \varepsilon_{xx}^{p})$$

in the first case, $\sigma_{xx} = E(T)(0 - \varepsilon^{th} - \varepsilon_{xx}^{p})$, and in the second: $\sigma_{xx} = E(T)(\varepsilon - \varepsilon_{xx}^{p})$.

It is thus enough, at every moment to apply, for mechanical calculation, $\varepsilon_{xx} = -\varepsilon^{th} = -\alpha(T)(T - T_{ref})$.

Moreover, to get the same results in both cases, it is necessary, with each step of time of the second simulation, to carry out pure mechanical calculation with coefficients whose values are interpolated according to the temperature at the moment running (operand list_mater).

3.3.1 Operand **MATER**

♦   MATER = to subdue,

This keyword makes it possible to inform the name of material (to subdue) defined by DEFI_MATERIAU [U4.43.01], where are provided the parameters necessary to the behavior chosen, functions of the temperature.

3.3.2 Operand **LIST_MATER**

♦   **LIST**_MATER =list_mater,

This keyword makes it possible to inform a material list (list_mater), defined by DEFI_MATERIAU [U4.43.01], whose constant parameters correspond to those of subduing, interpolated according to the temperature.

3.3.3 Operands **ALPHA/YOUNG**

◆   ALPHA   = alpha,   [function]
◆   YOUNG   = Young,   [function]

These keyword make it possible to inform the Young modulus and the thermal dilation coefficient functions of the temperature, in order to calculate the thermal strains and the corresponding stresses.

3.3.4 Operands **TEMP_INIT / TEMP_FIN / INST_FIN**

Code_Aster

Titre : Procédure TEST_COMPOR
Responsable : HABOUSSA David

Version
default

Date : 01/06/2016 Page : 6/9
Clé : U4.92.06   Révision   :
41f9ef5226e6

```
    ◆   TEMP_INIT    =  temp_init,                      [R]
    ◆   TEMP_FIN     =  temp_fin,                       [R]
    ◊   INST_FIN     =  temp_fin,                       [R]
```

These keyword make it possible to inform the temperatures initial and final, as well as intant final transient (correspondent with `temp_fin`), being worth $1.$ by default.

3.3.5 Operands NB_VARI/VARI_TEST

```
    ◆   NB_VARI      =  nb_vari,                        [I]
    ◊   VARI_TEST    =  vari_test,                      [KN]
```

These keyword make it possible to inform the number of internal variables of the behavior chosen, as well as the internal variables to test (by defaults, all the internal variables are tested).

3.3.6 Operands D_SIGM_EPSI / C_PRAG

```
    ◊   D_SIGM_EPSI  =  d_sigm_epsi,                   [function]
    ◊   C_PRAG       =  c_prag,                        [function]
```

In the typical case of behaviours with linear kinematic work hardening, these keywords make it possible to define the slope of kinematic work hardening according to the temperature. This slope is worth:
   • `d_sigm_epsi` for the behavior `VMIS_CINE_LINE`,
   • `c_prag` for the behaviors `VMIS_ECMI_LINE`, `VMIS_ECMI_TRAC`.

3.3.7 Operand SUPPORT

```
    ◊   SUPPORT       = /'NOT'                          [DEFECT]
                        /'ELEMENT'
```

See [U4.51.12]

3.4   OPTION = 'MECA'

Pure mechanical test, which implements a simulation of a way of loading in deformations in a material point, i.e. on a model such as the stress and strain states are homogeneous at any moment. It thus makes it possible to test a certain number of models of behavior, with an aim of checking the robustness of their digital integration, their insensitivity compared to a change of units, invariance compared to a total rotation applied to the problem, the accuracy of the tangent matrix. This test proceeds, for each modeling, with an intercomparison between the reference solution (obtained with a step of very fine time), the solution with a fairly coarse discretization, the solution with effect of the temperature (or another variable of order), the solution by changing the system of units (Pa in MPa), and that obtained after rotation or symmetry (see the document [v6.07.101]).

3.4.1 Operand LIST_MATER

```
    ♦    LIST_MATER =list_mater,
```

This keyword makes it possible to inform a list of 2 materials (`list_mater`), defined by `DEFI_MATERIAU` [U4.43.01], whose constant parameters are evaluated either in Pa , that is to say in Mpa .

Code_Aster

Version default

Titre : Procédure TEST_COMPOR
Responsable : HABOUSSA David

Date : 01/06/2016 Page : 7/9
Clé : U4.92.06 Révision :
41f9ef5226e6

3.4.2 Operands FISH / YOUNG

◆ FISH   =   fish,   [R]
◆ YOUNG   =   Young,   [R]

These keyword make it possible to inform the Young modulus and the Poisson's ratio.

3.4.3 Operands LIST_NPAS/LIST_TOLE

◊ LIST_NPAS   =   list_npas,   [l_I]
◊ LIST_TOLE   =   list_tole,   [l_R]

These keyword make it possible to inform the discretization in time and the tolerances corresponding.

By default, $list\_npas = [1,1,1,1\,,\,1,5,25]$ (4 "equivalent" problems with the coarsest discretization, is 1 increment by segment of loading, then variation of the discretization: 1 then 5 then 25 increments per segment).

By default, $list\_tole = 4 \times [1.E-10] + [1.E-1] + 2*[1.E-2] + [1.E-8]$. The precision necessary for the problems equivalent is voluntarily very small (if not there is a risk of bug). The following precise details are looser, since the comporements are in general sensitive to the discretization in time. The denière value is the tolerance on the tangent matrix.

3.4.4 Operand PREC_ZERO

◊ PREC_ZERO   =   prec_zero,   [l_R]

This keyword make it possible to provide one zero "digital" for each variable tested, in order to calculate a significant relative error. `prec_zero` thus the same length has as `vari_test`. By default this list is worth: $3 \times 1.E-10$.

3.4.5 Operand VARI_TEST

◊ VARI_TEST   =   vari_test,   [KN]

List of the components tested, presumedly invariant in the problem equivalents (rotation, change of unit). By default `vari_test = ('V1', VMIS', 'TRACE')`.

3.4.6 Operand SUPPORT

◊ SUPPORT   = /'NOT'   [DEFECT]
                 /'ELEMENT'

See [U4.51.12]

3.4.7 Keyword MODELING

The keyword MODELING allows, in the case SUPPORT=' ELEMENT', to carry out calculation on an element 3D or an element 2D, in plane constraints. It is not available in the case SUPPORT=' POINT', because it is enough to impose a zero value on the components corresponding to the plane constraints or the plane deformations to get the same result.

3.4.8 Keyword ANGLE

This keyword makes it possible to specify an angle (in degrees) to carry out an overall rotation around Z applied at the same time to the loading, the grid, and the examination. This especially makes it possible to check the reliability of the integration of the behavior, as in the tests **COMP001, COMP002**.

Code_Aster

Titre : Procédure TEST_COMPOR
Responsable : HABOUSSA David

Version
default

Date : 01/06/2016 Page : 8/9
Clé : U4.92.06   Révision   :
41f9ef5226e6

By default, rotation is identically worthless.

In the case of materials having an intrinsic orientation (orthotropism, behaviors crystalline), it is advisable to also use the keyword SOLID MASS, with a first value of angle identical to that provided under ANGLE.

3.4.9 Keyword SOLID MASS / ANGL_EULER/ANGL_REP

These keyword make it possible to lay down an intrinsic orientation in the material (orthotropism, behaviors crystalline), and make it possible to appeal in the macro-order the keyword **SOLID MASS** of **AFFE_CARA_ELEM** [U4.42.01].
By default, the orientation is worthless, and one does not call on **AFFE_CARA_ELEM**.

3.4.10 Keyword VERI_MATR_OPTION

This keyword is used to gather the operands which manage the behavior of TEST_COMPOR during the checking of the analytical tangent matrix with that obtained by digital disturbance.

3.4.11 Operand VALE_PERT_RELA

◊ VALE_PERT_RELA =   [R]

Allows to define the value of the digital relative disturbance which intervenes in the calculation of the disturbed matrix. For more detail to refer to [U4.51.11].

3.4.12 Operand PRECISION

◊ PRECISION   =   [R]

The operand PRECISION allows to define the value in the top of which it is considered that the analytical matrix and the disturbed matrix are different.

3.4.13 Operand PREC_ZERO

◊ PREC_ZERO   =   [R]

Below PREC_ZERO, one does not compare the values of the terms of the tangent matrix. That makes it possible to manage the situations where the terms of the disturbed tangent matrix are very close to zero.

3.5 Operand INFORMATION

Specify the detail of the information printed in the file message.

In mode INFO=2, one prints all the tables produced by SIMU_POINT_MAT.

4   Example

4.1   OPTION=' MECA'

See tests COMP001 [V6,07,101] , COMP002 [V6,07,102]

```
Pa #unities
STEEL [0] =DEFI_MATERIAU (ELAS=_F (E=YOUNG_Pa,
                         NU=POISSON,
                         ALPHA=11.8e-6),
                 ECRO_LINE=_F (D_SIGM_EPSI=pente_Pa,
```

Code_Aster

Titre : Procédure TEST_COMPOR
Responsable : HABOUSSA David

Version
default

Date : 01/06/2016 *Page : 9/9*
Clé : U4.92.06 *Révision* :
41f9ef5226e6

```
                                        SY=SY_Pa,),);
        #unities in MPa
        STEEL [1] =DEFI_MATERIAU (ELAS=_F (E=YOUNG,
                                          NU=POISSON,
                                          ALPHA=11.8e-6),
                         ECRO_LINE=_F (D_SIGM_EPSI=pente,
                                          SY=SY,),)
        compor=' VMIS_ISOT_LINE'
        tabresu=TEST_COMPOR (OPTION=' MECA',
                    COMPORTEMENT=_F (RELATION=compor,),
                    NEWTON=_F (REAC_ITER=1),
                    LIST_MATER=ACIER,
                    VARI_TEST= ('V1', 'VMIS', 'TRACE'),
                    YOUNG=YOUNG, POISSON=POISSON,
                    )
```

4.2   OPTION=' THER'

```
        See tests COMP008*
        TREF = 0.
        Tmax = 500.
        YOUN=DEFI_FONCTION (NOM_PARA=' TEMP', VALE= (TREF, 200000. ,
                                              Tmax, 100000. ,
                                                ),);
        ALPH=DEFI_FONCTION (NOM_PARA=' TEMP', VALE= (TREF, 1.E-5,
                                             Tmax, 2.E-5,
                                                ),);
        SIGY=DEFI_FONCTION (NOM_PARA=' TEMP', VALE= (TREF, 100. ,
                                             Tmax, 50. ,
                                                ),);
        DSDE=DEFI_FONCTION (NOM_PARA=' TEMP', VALE= (TREF, 10000. ,
                                             Tmax, 5000. ,
                                                ),);
        MATERI=DEFI_MATERIAU (ELAS_FO=_F (E=YOUN, NU=ZERO,
                                         TEMP_DEF_ALPHA=TREF,
                                         ALPHA=ALPH,),
                            ECRO_LINE_FO=_F (D_SIGM_EPSI=DSDE,
                                                   SY=SIGY,),
                     );
        LMAT2  = [Nun] * (NCAL)
        time=0.
        for I in arranges (NCAL):
            timem = time
            time = timem + tfin/NCAL
            Ti = T0 + time /tfin * (Tmax - T0)
            LMAT2 [I] =DEFI_MATERIAU (ELAS =_F (E=YOUN (Ti),
                                               NU=0.,
                                               ALPHA=0.,),
                              ECRO_LINE=_F (D_SIGM_EPSI=DSDE (Ti),
                                                  SY=SIGY (Ti),),)
        compor=' VMIS_ISOT_LINE'

        tabresu=TEST_COMPOR (
                MATER=MATERI, COMPORTEMENT=_F (RELATION=compor),
                LIST_MATER=LMAT2, ALPHA=ALPH, YOUNG=YOUN, TEMP_INIT=TREF,
        TEMP_FIN=Tmax,
                NEWTON=_F (REAC_ITER=1),
                NB_VARI=2,  VARI_TEST= ('V1', 'V2'),
                    )
```