Macro command SIMU_POINT_MAT

1 Drank

Compute the mechanical evolution of a material point, into quasi-static nonlinear.

All the behaviors available in STAT_NON_LINE [U4.51.11] are it also here.

The goal of this macro-command is to simplify to the maximum the data: it is enough to provide:
1) The behavior and the material;
2) Functions defining the evolution of the components selected stresses or strains;
3) Discretization in time.

This makes it possible in particular in the case of to calculate the evolution of the tensor of the stresses imposed strains, or the reverse (current cases in identification of material parameters)

Produced a data structure of the type counts container, according to time, the evolution of all the components of the stress tensors and strains, as well as the local variables.

2 Syntax

```
tabres [array] = SIMU_POINT_MAT (

♦/COMP_INCR         =_F (see the document [U4.51.11] ),
   /COMP_ELAS       =_F (see the document [U4.51.11] ),

♦MATER=mater                  ,                              [to subdue]

◊MASSIF                 =  "ANGL_REP"                        [R]
                       /"ANGL_EULER"                   [R]
◊  ANGLE             =angz ,                           [R]

♦INCREMENT          =_F  ( see the document [U4.51.03]),

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

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

◊  SUPPORT =   /"ELEMENT"

            ◊  MODELISATION      =     /"3D"          [DEFAULT]
                                       /"C_PLAN"
                                       /"D_PLAN"

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

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

             # scalar command variables function of time

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

            ◊  AFFE_VARC      =_F    (♦ NOM_VARC = "TEMP",
                                        /"CORR",
                                        /"IRRA",
                                        /"HYDR",
                                        /"SECH",
                                        /"EPSA",
                                        /"NEUT1",
                                        /"NEUT2",
                                    ◊ VALE_REF=vref            [R]
                                   ♦ VALE_FONC   =foncvarc    [ function]

                                        /"M_ACIER",
                                   /  ◊ V1 =foncv1             [ function]
                                   /  ◊ V2 =foncv2             [ function]
                                   /  ◊ V3 =foncv3             [ function]
                                   /  ◊ V4 =foncv4             [ function]
                                   /  ◊ V5 =foncv5             [ function]
                                   /  ◊ V6 =foncv6             [ function]
                                   /  ◊ V7 =foncv7             [ function]
                                        /"M_ZIRC",
                                   /  ◊ V1 =foncv1             [ function]
                                   /  ◊ V2 =foncv2             [ function]
                                   /  ◊ V3 =foncv3             [ function]
                                   /  ◊ V4 =foncv4             [ function]
                                   ),
```

```
                ◊SIGM_IMPOSE=_F   ( ◊SIXX   = sigxx                [function]
                                    ◊SIYY   = sigyy                [function]
                                    ◊SIZZ   = sigzz                [function]
                                    ◊SIXY   = sigxy                [function]
                                    ◊SIXZ   = sigxz                [function]
                                    ◊SIYZ   = sigyz                [function]
                ),
                ◊EPSI_IMPOSE=_F   ( ◊EPXX   = epsxx                [function]
                                    ◊EPYY   = epsyy                [function]
                                    ◊EPZZ   = epszz                [function]
                                    ◊EPXY   = epsxy                [function]
                                    ◊EPXZ   = epsxz                [function]
                                    ◊EPYZ   = epsyz                [function]
                ),


◊  SUPPORT=   /"POINT"                              [DEFAULT]
              ◊  NB_VARI_TABLE = nvar  [I]
              ◊  FORMAT_TABLE  =  /"CMP_COLONNE"    [DEFAULT]
                                  /"CMP_LIGNE"
              ◊  OPER_TANGENT  =   "NON"            [DEFAULT]
                                  /"YES"
              ◊  ARCHIVAGE =_F (
                    ◊   LIST_INST = linst (see document [U4.51.03]),
                    ◊   accuracy= prec ),
              /  ◊SIGM_IMPOSE=_F   ( ◊SIXX   = sigxx            [function]
                                     ◊SIYY   = sigyy            [function]
                                     ◊SIZZ   = sigzz            [function]
                                     ◊SIXY   = sigxy            [function]
                                     ◊SIXZ   = sigxz            [function]
                                     ◊SIYZ   = sigyz            [function]),
                 ◊   EPSI_IMPOSE=_F (    ◊  EPXX = epsxx
        [function]
                                     ◊EPYY   = epsyy            [function]
                                     ◊EPZZ   = epszz            [function]
                                     ◊EPXY   = epsxy            [function]
                                     ◊EPXZ   = epsxz            [function]
                                     ◊EPYZ   = epsyz            [function]),
              /  ◊GRAD_IMPOSE=_F   ( ◊F11    = fonc              [function]
                                     ◊F12    = fonc              [function]
                                     ◊F12    = fonc              [function]
                                     ◊F13    = fonc              [function]
                                     ◊F21    = fonc              [function]
                                     ◊F22    = fonc              [function]
                                     ◊F23    = fonc              [function]
                                     ◊F31    = fonc              [function]
                                     ◊F32    = fonc              [function]
                                     ◊F33    = fonc              [function]),
              /  ◊MATR_C1=_F    ( ♦VALE   = cij                [R]
                                  ♦NUMÉRIQUE_LIGNE  = numlig        [I]
                                   ♦ NUME_COLONNEE = numcol      [I]
                                  ),
                 ◊MATR_C2=_F    ( ♦VALE   = cij                [R]
                                  ♦NUMÉRIQUE_LIGNE  = numlig        [I]
                                   ♦ NUME_COLONNEE = numcol      [I]
                                  ),
                 ◊VECT_IMPO=_F  ( ♦VALE   = cij                [R]
                                  ♦NUMÉRIQUE_LIGNE  = numlig        [I]
                                  ),
              ◊  AFFE_VARC      =_F    (♦ NOM_VARC = "TEMP",
```

```
                                        /"IRRA",
                                        /"SECH",
                                    ♦ VALE_FONC   =foncvarc     [ function]
                                    ◊ VALE_REF=vref             [R]
                                ),
                    ◊    SIGM_INIT=_F ( ◊   SIXX = sigxx                      [R]
                           ◊   SIYY = sigyy                    [R]
                           ◊   SIZZ = sigzz                    [R]
                           ◊   SIXY = sigxy                    [R]
                           ◊   SIXZ = sigxz                    [R]
                           ◊   SIYZ = sigyz                    [R] ),
                    ◊    EPSI_INIT=_F ( ◊EPXX   = epsxx                       [R]
                           ◊EPYY   = epsyy                     [R]
                           ◊EPZZ   = epszz                     [R]
                           ◊EPXY   = epsxy                     [R]
                           ◊EPXZ   = epsxz                     [R]
                           ◊EPYZ   = epsyz                     [R] ),
          ◊VARI_INIT=_F (     ◊VALE   = vari                    [R] ),
          ◊INFO   =      1,                          [DEFAULT]
                    /2 , );
```

3 Operands

3.1 Operand MATER

♦ **MATER** =mater,

This key word makes it possible to inform the name of the material (to subdue) definite by DEFI_MATERIAU [U4.43.01], where are provided the parameters necessary to the selected behavior.

3.2 Key words COMP_INCR/COMP_ELAS

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

3.3 Key keys INCREMENT/FILING NEWTON/CONVERGENCE

the syntax of these key words is described in the document [U4.51.03].

Key word INCREMENT defines the intervals of time taken in the incremental method.

Key word ARCHIVAGE defines times when the results in the array tabres are stored. In the case SUPPORT=' POINT', these times only by the key word LIST_INST with the relative accuracy can be defined accuracy.

Key words NEWTON and CONVERGENCE, optional, make it possible to modify the default values of the parameters of convergence of the method of Newton.

3.4 Key word RECH_LINEAIRE

the syntax of these key words is described in the document [U4.51.03].

Key word RECH_LINEAIRE allows, in case SUPPORT=' ELEMENT', to activate the linear search to help with the convergence of the algorithm of Newton. This functionality is not available for SUPPORT=' POINT', because it does not seem necessary.

3.5 Key word MODELISATION

key word MODELISATION allows, in case SUPPORT=' ELEMENT', 3D to carry out computation on an element or an element 2D, in plane stresses or plane strains. It is not available in case SUPPORT=' POINT', because it is enough to zero impose a value on the components corresponding to the plane stresses or the plane strains to obtain the same one result.

This key word makes it possible to define the dimension of with the dealt problem: 3D (by default) or 2D: plane strain or plane stress. In the case 2D, the components of the tensors provided under key words SIGM_IMPOSE, EPSI_IMPOSE, SIGM_INIT, EPSI_INIT are 4: XX, YY, ZZ, XY.

3.6 Operand ANGLE

This key word 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 mesh, and the examination. This especially makes it possible to check the reliability of the integration of the behavior, as in tests COMP001, COMP002.
By default, rotation is identically null.

In the case of materials having an intrinsic directional sense (orthotropy, behaviors crystalline), it is advisable to also use key word MASSIF, with a first value of angle identical to that provided under ANGLE.

3.7 Key word MASSIF

3.7.1 Operands ANGL_EULER/ANGL_REP

These key word make it possible to lay down an intrinsic directional sense in the material (orthotropy, behaviors crystalline), and make it possible to appeal to the key word in macro-command MASSIF of AFFE_CARA_ELEM [U4.42.01].
By default, the directional sense is null, and one does not call on AFFE_CARA_ELEM.

3.8 Keywords SIGM_INIT/EPSI_INIT/VARI_INIT

These keywords make it possible to define an initial state by the data:
1) of the components initial stresses (all the components are not necessary, by default one takes value 0),
2) of the components initial strains (if key word EPSI_INIT is present, it is necessary to provide all the components of the initial strains: 4 in 2D, and 6 in 3D)
3) the group of initial intern variables for the behavior used.

This functionality is illustrated in test SSNV160E.

3.9 Key keys SIGM_IMPOSE/EPSI_IMPOSE

3.9.1 Operands SIXX, SIYY, SIZZ, SIXY, SIXZ, SIYZ

These key word make it possible to define stress tensor of the components imposed on the material point, via functions of time. These functions can be defined using DEFI_FONCTION [U4.31.02] or FORMULA [U4.31.05].

By defaults, the nonaffected components are identically null.

3.9.2 Operands EPXX, EPYY, EPZZ, EPXY, EPXZ, EPYZ

These key word make it possible to define strain tensor of the components imposed on the material point, via functions of time. These functions can be defined using DEFI_FONCTION [U4.31.02] or FORMULA [U4.31.05].

By defaults, the nonaffected components are left without value (not imposed strain).

3.10 Key words GRAD_IM POSES

3.10.1 Operands F11, F12, F13, F21, F22, F23, F31, F32, F33

These key word make it possible to define all the components of the tensor gradient of transformation imposed, in large deformations (DEFORMATION=' SIMO_MIEHE ") cf test ssnd113).

3.11 Key keys MATR_C1/MATR_C2/VECT_IMPO

These key word allow, in case SUPPORT=' POINT", to directly define the coefficients of the matrixes $C1$, $C2$ and the vector g described with the §4.24.2 : that thus makes it possible to define linear conditions on the unknowns (forced and strains of the material point) more general than the components imposed by keywords SIGM_IMPOSE/EPSI_IMPOSE. All the terms of the matrixes $C1$ and $C2$ not specified are null. For an example of use, to see test WTNV134B [V7.31.134].

3.12 Operand AFFE_VARC

This key word makes it possible to specify a command variable (cf [U4.43.03]) whose name is defined under key word NOM_VARC ; the function defining the temporal evolution of this command variable is provided via key word VALE_FONC. The possible value of reference vref is given by VALE_REF.

In the case SUPPORT=' ELEMENT', all the command variables are authorized. Moreover, for M_ZIRC (resp. M_ACIER), it is necessary to provide the evolutions of the 4 (resp. 7) metallurgical phases according to time.

In case SUPPORT=' POINT', only command variables " TEMP ", " SECH " and " IRRA " are authorized.

3.13 Key word NB_VARI_TABLE

key word NB_VARI_TABLE allows, in case SUPPORT=' POINT', to limit the number of local variables written in the array. Indeed for the polycrystalline mediums, this one can reach several thousands. One then limits the number of columns of the array to nvar. On the other hand computations are of course carried out with the totality of the local variables: those are truncated only in the array as a result.

3.14 Key word FORMAT_TABLE

◊ FORMAT_TABLE = /"CMP_COLONNE" [DEFAULT]
/"CMP_LIGNE"

key word FORMAT_TABLE allows, in case SUPPORT=' POINT', result to define the mode of storage of the quantities in the array (test SSNV194C illustrates these two formats). If the number of local variables exceeds the maximum of columns authorized for an array (9999, cf D4,02,05), the format rocks automatically in: FORMAT_TABLE = /"CMP_LIGNE".

```
FORMAT_TABLE = /"CMP_LIGNE" :
…  V845          V846          V847          V848          NB_ITER
…  1.16186E-17   1.32359E-17   1.11751E-17   1.00000E+00   1.00000E+00
…  1.29473E-16   1.47341E-16   1.24474E-16   1.00000E+00   1.00000E+00
…  8.90739E-16   1.00875E-15   8.55093E-16   1.00000E+00   1.00000E+00
…  4.40109E-15   4.92817E-15   4.21938E-15   1.00000E+00   1.00000E+00
…  1.70332E-14   1.87022E-14   1.63484E-14   1.00000E+00   1.00000E+00
…  5.44940E-14   5.80870E-14   5.25904E-14   1.00000E+00   1.00000E+00

FORMAT_TABLE = /"CMP_COLONNE" :
…  INST          QUANTITY CMP      VALEUR
….
  4.97867E-03 VARI     V845     -1.43828E+01
  4.97867E-03 VARI     V846     -2.63548E+01
  4.97867E-03 VARI     V847      2.80907E+01
  4.97867E-03 VARI     V848      1.00000E+00
  5.00000E-03 EPSI     EPXX     -2.20535E-03
  5.00000E-03 EPSI     EPYY     -1.96506E-03
  5.00000E-03 EPSI     EPZZ      5.00000E-03
  5.00000E-03 EPSI     EPXY     -1.98892E-04
  5.00000E-03 EPSI     EPXZ     -2.11427E-04
  5.00000E-03 EPSI     EPYZ     -3.00870E-04
  5.00000E-03 V7.32.119SIGM     SIXX      1.67146E-04
  5.00000E-03 SIGM     SIYY      2.78713E-05
  5.00000E-03 SIGM     SIZZ      3.01140E+02
  5.00000E-03 SIGM     SIXY      9.15194E-05
  5.00000E-03 SIGM     SIXZ      1.62000E-04
  5.00000E-03 SIGM     SIYZ      6.86376E-05
  5.00000E-03 SIEQ     VMIS      3.01140E+02
```

```
   5.00000E-03 SIEQ     TRACES     3.01140E+02
   5.00000E-03 VARI     V1        -1.58316E-03
   5.00000E-03 VARI     V2        -1.34287E-03
   5.00000E-03 VARI     V3         2.92604E-03
   5.00000E-03 VARI     V4        -2.81276E-04
….
```

3.15 Key word OPER_TANGENT

◊ OPER_TANGENT = "NON" [DEFAULT]
/"YES"

key word OPER_TANGENT allows, in case SUPPORT=' POINT', result to add to the array the 36 values of the tangent operator resulting from the behavior.

3.16 Operand INFO

◊ **INFO =inf**

Makes it possible to carry out in the message file various intermediate printings.

4 The purpose of operation of the macro_commande

This macro_commande is restricting with bare essential the relative data for a simulation on a material point for a model of incremental behavior.

The inner working thus reduces the command file of the user, by carrying out repetitive operations for this kind of situations.

4.1 Case SUPPORT=' ELEMENT'

operation is:

1. creation of a mesh of only one element at only one Gauss point (a tetrahedron with four nodes in 3D, a triangle with three nodes in 2D) (see for example [V6.04.176]).
2. assignment of a model 3D or C_PLAN or D_PLAN
3. assignment of the material on this mesh;
4. assignment of the loadings:
5. with regard to the imposed strains, for each component affected via one of the key words of EPSI_IMPOSE , creation of a unit loading in strain which will be multiplied by the function of time provided for this component by the user;
6. with regard to the imposed stresses, for each component affected via one of the key words of SIGM_IMPOSE , creation of a unit loading in stresses which will be multiplied by the function of time provided for this component by the user;
7. Call to STAT_NON_LINE . All the key words having default values are used, except if they are overloaded by the utilisor (NEWTON, CONVERGENCE, SUIVI_DDL, ARCHIVAGE, RECH_LINEAIRE) and of the initial state.

All the results (six or four stress components and of strains, local variables) are stored in an array (tabres). For each component (column of the array) the evolution according to time appears.

4.2 Case SUPPORT=' POINT'

In this case, rather than to use a finite element (even single) to carry out computation, SIMU_POIN_MAT calls on a dedicated command, CALC_POINT_MAT, which 3D includes in FORTRAN the direct call with the routine of integration of the behaviors, NMCOMP. This is available only in the case of the small strains.

Let us recall that NMCOMP is the general routine of integration of the constitutive laws, called by all the finite elements 3D and 2D. It allows computation in a point (this point being the point of integration for a finite element) of the stresses and local variables current time, knowing the stresses and local variables at previous time; and the current increase in strain. (confer [D5.04.01] and [R5.03.14]). More precisely, at time t_i, and the iteration n the tensor of the stresses σ_i^n, in a point is calculated from $(\sigma_{i-1}, \alpha_{i-1})$ and of the increment of strain $\Delta\varepsilon_i^n$.

When all the components of the tensor of the strains are provided, the algorithm is immediate: it is about a simple loop in time, containing for each temporal increment the data of the mechanical state of the preceding increment and the tensor (symmetric) correpondant with the known increase in strain.

But in the contrary case, either that one provides only n component history of the strains $n<6$, or that one provides n component history of the stresses, the algorithm is the following:

- by default, any component not specified corresponds to a component of subjugated stress to remain null (condition of Neumann)
- the equations to be solved are (by means of the notation in vectorial form of the symmetric tensors of order 2):
- $\boldsymbol{\sigma}_\mathbf{i} = F(\Delta\boldsymbol{\varepsilon}_\mathbf{i}; \sigma_{i-1}, \alpha_{i-1})$ where F represents result integration of the behavior by NMCOMP
- for J varying from 1 to 6:
  - either $(\sigma_i)_j = g_j(t_i)$
  - or $(\varepsilon_i)_j = g_j(t_i)$
    - where $\bar{\sigma}_j(t)$ and $\bar{\varepsilon}_j(t)$ are given by SIGM_IMPOSE/EPSI_IMPOSE.

This can be still written:

for each time t_i, to solve:

$$R(Y_i)=0 \text{ with } Y_i=Y(t_i)=\begin{Bmatrix}[\boldsymbol{\sigma}_\mathbf{i}]\\ [\boldsymbol{\varepsilon}_\mathbf{i}]\end{Bmatrix} \text{ and } R(Y_i)=\begin{Bmatrix}[\boldsymbol{\sigma}_\mathbf{i} - F(\Delta\boldsymbol{\varepsilon}_\mathbf{i};\ \boldsymbol{\sigma}_\mathbf{i-1}, \alpha_{i-1})]\\ [C_1]\boldsymbol{\sigma}_\mathbf{i} + [C_2]\boldsymbol{\varepsilon}_\mathbf{i} - \mathbf{g}(t_i)\end{Bmatrix}$$

which is a nonlinear system of order 12.

The last relation translates the condtitions of stresses or imposed strains: the matrixes C_1 and C_2 contain only terms on the diagonal, being worth 1 if the corresponding component is imposed, knowing that one cannot have at the same time forced and imposed strain.

For example, if the strain ε_{yy} is imposed, the last relation is written:

$$[C_1]\boldsymbol{\sigma}_\mathbf{i} + [C_2]\boldsymbol{\varepsilon}_\mathbf{i} - \mathbf{g}(t_i) = \begin{bmatrix}1&0&0&0&0&0\\0&0&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}\begin{Bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{zz}\\ \sigma_{xy}\\ \sigma_{xz}\\ \sigma_{yz}\end{Bmatrix} + \begin{bmatrix}0&0&0&0&0&0\\0&1&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{bmatrix}\begin{Bmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{zz}\\ \varepsilon_{xy}\\ \varepsilon_{xz}\\ \varepsilon_{yz}\end{Bmatrix} = \begin{Bmatrix}0\\ \varepsilon_{yy}(t_i)\\ 0\\ 0\\ 0\\ 0\end{Bmatrix}$$

If the user specified (via key keys MATR_C1, MATR_C2 and VECT_IMPO) of the additional relations, those are taken into account directly in the matrixes C_1 and C_2.

the nonlinear resolution of this system of equations is carried out by a method of Newton:

-initialization: $Y_i^0 = Y_{i-1} + [K_i^0]^{-1}\begin{bmatrix}0\\ -[C_1]\boldsymbol{\sigma}_\mathbf{i-1} - [C_2]\boldsymbol{\varepsilon}_\mathbf{i-1} + \mathbf{g}(t_i)\end{bmatrix}$

-iteration N has:

$$\delta Y_i^{n+1} = -[K_i^n]^{-1} R(Y_i^n) = [K_i^n]^{-1} \begin{bmatrix} F(\Delta \boldsymbol{\varepsilon}_{\mathbf{i}}^{\mathbf{n}}; \boldsymbol{\sigma}_{\mathbf{i-1}}, \alpha_{i-1}) - \boldsymbol{\sigma}_{\mathbf{i}}^{\mathbf{n}} \\ -[C_1]\boldsymbol{\sigma}_{\mathbf{i}}^{\mathbf{n}} - [C_2]\boldsymbol{\varepsilon}_{\mathbf{i}}^{\mathbf{n}} + \mathbf{g}(t_i) \end{bmatrix} ;$$

$$\Delta Y_i^n = \delta Y_i^n + \Delta Y_i^{n-1} \ ; \ Y_i^n = \Delta Y_i^n + Y_{i-1}$$

with

$$[K_i^n] = \left[\frac{\partial R}{\partial Y}\right] = \begin{bmatrix} \mathbf{1} & -\left(\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}\right)_i^n \\ [C_1] & [C_2] \end{bmatrix} \text{ and } [K_i^0] = \begin{bmatrix} \mathbf{1} & -\left(\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}\right)_i^0 \\ [C_1] & [C_2] \end{bmatrix}$$

where $\left(\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}\right)_i^0$ the tangent operator of prediction represents (option RIGI_MECA_TANG , cf [R5.03.01, R5.03.02]) and $\left(\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}\right)_i^n$ represents the coherent tangent operator (option FULL_MECA) , cf [R5.03.01, R5.03.02]). These operators can be replaced by the operator elasticity according to key words PREDICTION, REAC_ITER .

The only not linearity of the problem comes from the behavior: $F(\Delta \boldsymbol{\varepsilon}_{\mathbf{i}}^{\mathbf{n}}; \boldsymbol{\sigma}_{\mathbf{i-1}}, \alpha_{i-1})$.

In the case of a linear behavior, one checks that the solution of the problem is obtained at the conclusion of phase of prediction.

The convergence of the iterations is vérfiée:

- maybe in relative value, (key word RESI_GLOB_RELA) :

$$max\left(\frac{\max_{j=1,6}\left|(R_i^n)_j\right|}{\max_{j=1,6}\left|(\sigma_i^0)_j\right|}; \frac{\max_{j=7,12}\left|(R_i^n)_j\right|}{\max_{j=7,12}\left|(R_i^0)_j\right|}\right) < \text{RESI\_GLOB\_RELA}$$

  In this case, the terms in stresses and strains are separate for the examination of the convergence criterion to avoid the problems due to the differences in orders of gandor.

- maybe in absolute value (key word RESI_GLOB_MAXI) or vavlor of the denominator close to zero in the relative criterion above:

$$\max_{j=1,12}\left|(R_i^n)_j\right| < \text{RESI\_GLOB\_MAXI}$$

The computation options of the tangent stiffness by disturbance and the automatic management of time step are also activated, as in [U4.51.03].

In the preceding resolution, the terms in stresses are adimensionnalisés, to avoid a bad conditioning of the jacobian matrix. One thus divides for the resolution all the terms into stresses by the max of the diagonal terms of the operator of elasticity; it is thus necessary to provide in DEFI_MATERIAU key word ELAS or ELAS_ORTH or ELAS_ISTR.

In the case SUPPORT=' POINT', certain keywords do not have utility:
- in the case of the linear search, this one is not programmed in the current version
- in the case of the archivage, only key word LIST_INST is taken into account

- in case CONVERGENCE/RESI_REFE_RELA , this key word without object: the residue by value of reference does not have a meaning for a material point.
- If the value of key word DEFORMATION is not PETIT , one alarms the user by specifying that the selected type of DEFORMATION is incompatible with SUPPORT=POINT , and that one thus uses SUPPORT=ELEMENT, except if one provides all the components of the gradient of transformation (key word GRAD_IMPOSE).

5 Example of use

This example is resulting from test SSNV160E :

```
# TITER CAS TEST HYDROSTATICS CAM_CLAY IN 3D AVEC SIMU_POINT_MAT

#  CHARACTERISTIC OF MATERIAU
MATER=DEFI_MATERIAU (ELAS=_F (E=7.74E6, NU=0.285),
                     CAM_CLAY=_F (MU  = 6.E6,
                            PORO=0.66,
                            LAMBDA=0.25,
                            KAPA=0.05,
                            M=0.9,
                            PRES_CRIT=3.E5),);

#  CHARGEMENT
PRESS2=DEFI_FONCTION (NOM_PARA=' INST', NOM_RESU=' PRESSION',
            VALE= (0.0, 0.0,
           100.0, - 100000.0,
           600.0, - 320000.0,
           1000.0, - 350000.0,
           5000.0, - 500000.0,
           8000.0, - 800000.0),
           PROL_DROITE=' CONSTANT');

#  LISTE OF TIMES OF CALCUL
LI1=DEFI_LISTE_REEL (DEBUT=0.0,
       INTERVALLE= (_F (JUSQU_A=1000.0, NOMBRE=10,),
                    _F (JUSQU_A=1.E4, NOMBRE=60,),),);

SXXINI= -7.99000E+05
EXXINI= -1.82689E-02

RESU3=SIMU_POINT_MAT (
                COMP_INCR=_F   (RELATION='   CAM_CLAY',   ITER_INTE_MAXI=100,
ITER_INTE_PAS=-10,),
          NEWTON=_F (MATRICE=' TANGENTE', REAC_ITER=1,),
          CONVERGENCE=_F (ITER_GLOB_MAXI=20,),
          MATER      = MATER,
          INCREMENT=_F (LIST_INST=LI1, INST_INIT= 7990. , INST_FIN = 8000.),
          SIGM_INIT=_F (SIXX=SXXINI, SIYY=SXXINI, SIZZ=SXXINI,),
          EPSI_INIT=_F (   EPXX=EXXINI, EPYY=EXXINI, EPZZ=EXXINI,
                        EPXY=0., EPYZ=0., EPXZ=0.,),
          VARI_INIT=_F (   VALE= (   3.99500E+05,1.0,
7.99000E+05,4.63066E-10,
                                1.94773E-02,2.99086E-17,1.79821E+00),),
          SIGM_IMPOSE=_F ( SIXX=PRESS2, SIYY=PRESS2, SIZZ=PRESS2,),
          );
IMPR_TABLE (TABLE=RESU3)

the array result contains:
```

```
#-------------------------------------------------------------------------------
```

```
#
#CALC_POINT_MAT
 INST          EPXX          EPYY          … SIXX          SIYY          … TRACE        … V1           V2           .
  0.00000E+00  0.00000E+00  0.00000E+00     -1.00000E+05 -1.00000E+05      0.00000E+00      0.00000E+00  0.00000E+00
  2.00000E+02 -2.06631E-03 -2.06631E-03     -1.44000E+05 -1.44000E+05     -4.32000E+05      3.00000E+05  0.00000E+00
  3.00000E+02 -3.57721E-03 -3.57721E-03     -1.88000E+05 -1.88000E+05     -5.64000E+05      3.00000E+05  0.00000E+00
  4.00000E+02 -4.76888E-03 -4.76888E-03     -2.32000E+05 -2.32000E+05     -6.96000E+05      3.00000E+05  0.00000E+00
  5.00000E+02 -5.75297E-03 -5.75297E-03     -2.76000E+05 -2.76000E+05     -8.28000E+05      3.00000E+05  0.00000E+00
  6.00000E+02 -6.59119E-03 -6.59119E-03     -3.20000E+05 -3.20000E+05     -9.60000E+05      3.00000E+05  0.00000E+00
  7.00000E+02 -6.72247E-03 -6.72247E-03     -3.27500E+05 -3.27500E+05     -9.82500E+05      3.00000E+05  0.00000E+00
  8.00000E+02 -6.85078E-03 -6.85078E-03     -3.35000E+05 -3.35000E+05     -1.00500E+06      3.00000E+05  0.00000E+00
  9.00000E+02 -6.97624E-03 -6.97624E-03     -3.42500E+05 -3.42500E+05     -1.02750E+06      3.00000E+05  0.00000E+00
  1.00000E+03 -7.09899E-03 -7.09899E-03     -3.50000E+05 -3.50000E+05     -1.05000E+06      3.00000E+05  0.00000E+00
  1.40000E+03 -7.33679E-03 -7.33679E-03     -3.65000E+05 -3.65000E+05     -1.09500E+06      3.00000E+05  0.00000E+00
  1.80000E+03 -7.56501E-03 -7.56501E-03     -3.80000E+05 -3.80000E+05     -1.14000E+06      3.00000E+05  0.00000E+00
 .......
```