Architecture of the crystalline behaviors

Summary:

This document describes Lhas structure of integration of the crystalline behaviors (confer R5.03.11) and the actions to undertake to add a new crystalline behavior, with an aim of carrying out the calculations of aggregates, or the calculations homogenized using STAT_NON_LINE and SIMU_POINT_MAT.

Contents

1 General description of the crystalline behaviors

A crystalline behavior uses, besides DEFI_MATERIAU, DEFI_COMPOR.

The name of the behavior in BEHAVIOR is generic: MONOCRYSTAL or POLYCRYSTAL.

The algorithm of resolution is with the choice:
- for MONOCRYSTAL: NEWTON, NEWTON_RELI, NEWTON_PERT and RUNGE_KUTTA
- for POLYCRYSTAL: RUNGE_KUTTA

One describes in this document:
- the architecture of DEFI_COMPOR
- the architecture of the resolution clarifies with error controlled by Runge-Kutta (cf [R5.03.14])
- the architecture of the implicit resolution by Newton (environment PLASTI cf [R5.03.14])
- internal objects useful for the resolution.

To apprehend this document, the reading of R5.03.11 is highly advised.

2 Architecture of DEFI_COMPOR

2.1 Illustration on an example: test SSNV194

Let us start from an example (test ssnv194).

2.1.1 Modeling a: a small aggregate comprising 10 grains:

```
ACIER=DEFI_MATERIAU (ELAS=_F (E=145200.0, NU=0.3,),
                     MONO_VISC1=_F (N=10.0, K=40.0, C=1.0,),
                     MONO_ISOT1=_F (R_0=75.5, Q=9.77, B=19.34),
                     MONO_CINE1=_F (D=36.68,),);

MONO1 =DEFI_COMPOR (MONOCRISTAL= (_F (  MATER=ACIER, ELAS=' ELAS',
                                    ECOULEMENT=' MONO_VISC1',
                                    ECRO_ISOT=' MONO_ISOT1',
                                    ECRO_CINE=' MONO_CINE1',
                                    FAMI_SYST_GLIS=' BCC24',),),);

ORIEN=AFFE_CARA_ELEM (MODELE=TROISD, MASSIF= (
_F (GROUP_MA=' GM1', ANGL_EULER= (- 150,646, 33,864, 55,646,),),
_F (GROUP_MA=' GM2', ANGL_EULER= (- 137,138, 41.5917, 142,138,),),
_F (GROUP_MA=' GM3', ANGL_EULER= (- 166,271, 35.46958, 171,271,),),
_F (GROUP_MA=' GM4', ANGL_EULER= (- 77,676, 15.61819, 154,676,),),
_F (GROUP_MA=' GM5', ANGL_EULER= (- 78.6463, 33,864, 155,646,),),
_F (GROUP_MA=' GM6', ANGL_EULER= (- 65.1378, 41.5917, 142,138,),),
_F (GROUP_MA=' GM7', ANGL_EULER= (- 94.2711, 35.46958, 71,271,),),
_F (GROUP_MA=' GM8', ANGL_EULER= (- 5.67599, 15.61819, 154,676,),),
_F (GROUP_MA=' GM9', ANGL_EULER= (- 6.64634, 33,864, 155,646,),),
_F (GROUP_MA=' GM10', ANGL_EULER= (6.86224, 41.5917, 142,138,),),
                                ),);

SOLNL=STAT_NON_LINE (MODELE=…, CHAM_MATER=…, EXCIT=_…,
                     CARA_ELEM=ORIEN,
                     COMPORTEMENT=_F (RELATION=' MONOCRISTAL',
                                 COMPOR=MONO1,
                                 ),)
```

2.1.2 Modelings B and C: polycrystal comprising 10 grains:

A material point, 10 grains of identical voluminal fractions (0.1), and of orientations similar to those of modeling A (what makes it possible to find the same average solution):

```
MONO1=DEFI_COMPOR (MONOCRISTAL= (… identical to modeling A)

COMPORP=DEFI_COMPOR (POLYCRISTAL= (
_F  (MONOCRISTAL=COMPORT,  FRAC_VOL=0.1,  ANGL_EULER=  (-  150,646,  33,864,
55,646,),),
_F  (MONOCRISTAL=MONO1,  FRAC_VOL=0.1,  ANGL_EULER=  (-  137,138,  41.5917,
142,138,),),
_F  (MONOCRISTAL=MONO1,  FRAC_VOL=0.1,  ANGL_EULER=  (-  166,271,  35.46958,
171,271,),),
_F  (MONOCRISTAL=MONO1,  FRAC_VOL=0.1,  ANGL_EULER=  (-  77,676,  15.61819,
154,676,),),
_F  (MONOCRISTAL=MONO1,  FRAC_VOL=0.1,  ANGL_EULER=  (-  78.6463,  33,864,
155,646,),),
_F  (MONOCRISTAL=MONO1,  FRAC_VOL=0.1,  ANGL_EULER=  (-  65.1378,  41.5917,
142,138,),),
_F  (MONOCRISTAL=MONO1,  FRAC_VOL=0.1,  ANGL_EULER=  (-  94.2711,  35.46958,
71,271,),),
_F  (MONOCRISTAL=MONO1,  FRAC_VOL=0.1,  ANGL_EULER=  (-  5.67599,  15.61819,
154,676,),),
_F  (MONOCRISTAL=MONO1,  FRAC_VOL=0.1,  ANGL_EULER=  (-  6.64634,  33,864,
155,646,),),
_F  (MONOCRISTAL=MONO1,  FRAC_VOL=0.1,  ANGL_EULER=  (6.86224,  41.5917,
142,138,),),
),

LOCALIZATION  = ' BETA', DL=0., DA=0., MU_LOCA=145200. /2.6,);
```

Modeling b: (grid comprising only one element)

…

```
SOLNL=STAT_NON_LINE (MODELE=TROISD,
                     CHAM_MATER=MAT2,
                     EXCIT= (_F (CHARGE=TRAC,
                                 FONC_MULT=COEF,
                                 TYPE_CHARGE=' FIXE_CSTE',),),
                       INCREMENT= (_F (LIST_INST=LINST,),),
                    COMPORTEMENT= (_F (RELATION=' POLYCRISTAL',
                                  COMPOR=COMPORP2,
                                  DEFORMATION=' PETIT',
                                  ALGO_INTE=' RUNGE_KUTTA',
                                  TOUT=' OUI',
                                  RESI_INTE_RELA=1.E-6
                                  ),),
                     NEWTON= (_F (PREDICTION=' EXTRAPOLE',
                                  MATRICE=' ELASTIQUE',
                                  REAC_ITER=0,),),
                     CONVERGENCE= (_F (ITER_GLOB_MAXI=50,
                                    RESI_GLOB_RELA=1.E-4
                                    ),),);
```

Modeling C: (not material)

```
SOLNL=SIMU_POINT_MAT (COMPORTEMENT=_F (RELATION=' POLYCRISTAL',
                                       COMPOR=COMPORP,
                                       ALGO_INTE=' RUNGE_KUTTA',),

                      NEWTON=_F (MATRICE=' ELASTIQUE', REAC_ITER=0),
                      MATER =…, NB_VARI_TABLE=6,
                      INCREMENT=…,
                      EPSI_IMPOSE=…
                            );
```

2.2 Description of DEFI_COMPOR

2.2.1 Structure:

The purpose of routine OP0050 is to produce the structure of data described in [D4.06.24]: the objects composing this SD are different according to whether one treats a monocrystal (routine OP5901) or a polycrystal (routine OP5902).

2.2.2 Addition of a crystalline behavior to the catalogue of DEFI_MATERIAU/DEFI_COMPOR

If the new crystalline behavior uses parameters material different from those which are already available in the keywords MONO_* of DEFI_MATERIAU it is enough to introduce these new parameters of behavior, that is to say under only one keyword (case of the behaviors MONO_DD_*, that is to say by separating the coefficients relating to isotropic work hardening, kinematic work hardening, and the flow (cf. MONO_ISOT*, MONO_CINE*, MONO_VISC*,). These parameters will be exploited in integration (routines LCMMAT, LCMMAP), and the keyword corresponding factors will be used in DEFI_COMPOR.

Example: catalogue DEFI_MATERIAU

```
MONO_DD_CFC =FACT (statut=' f',
regles= ( UN_PARMI ('H', 'H1'),
          PRESENT_PRESENT ('H1', 'H2', 'H3', 'H4', 'H5'),
          PRESENT_ABSENT ('H', 'H1', 'H2', 'H3', 'H4', 'H5'),),

GAMMA0 =SIMP (statut=' f', typ=' R', defaut=0.001, units: S ** - 1"),
TAU_F =SIMP (statut=' o', typ=' R', fr= " in unit of constraints ex 20
MPa"),
With =SIMP (statut=' f', typ=' R', defaut=0.13, fr= " parameter A, without
unit"),
B =SIMP (statut=' f', typ=' R', defaut=0.005, fr= " parameter B, without
unit"),
NR =SIMP (statut=' f', typ=' R', defaut=200., fr= " parameter N, without
unit"),
There =SIMP (statut=' o', typ=' R', fr= " in unit of length ex 2.5 A"),
ALPHA=SIMP (statut=' f', typ=' R', defaut=0.35, fr= " parameter alpha"),
BETA =SIMP (statut=' o', typ=' R', fr= " parameter B, in unit of length"),
…),
```

This makes it possible to describe each coefficient, its character optional (with a possible value by default) or obligatory (for more precise details, to refer to [D5.01.01]).

The catalogue of DEFI_COMPOR is, according to the cases:

MONOCRYSTAL =FACT (statut=' f', max=5,

```
    MATER =SIMP (statut=' o', typ=mater_sdaster, max=1),
    ECOULEMENT=SIMP (statut=' o', typ=' TXM', into= ('MONO_VISC1',
    'MONO_VISC2',
                                          'MONO_DD_CFC', 'MONO_DD_CC',…),
                                    fr= " standard of viscoplastic flow"),
    ELAS =SIMP (statut=' f', typ=' TXM',),

    # case of a behavior of the type MONO_VISC*
    b_non_dd=BLOC (condition= " ECOULEMENT==' MONO_VISC1'
                            however ECOULEMENT==' MONO_VISC2',
                  ECRO_ISOT=SIMP (statut=' f', typ=' TXM', max=1,
                     fr= " To give the isotropic type of work hardening"),
                  ECRO_CINE=SIMP (statut=' f', typ=' TXM', max=1,
                     fr= " To give standard kinematic work hardening"),
                  FAMI_SYST_GLIS=SIMP (statut=' f', typ=' TXM',
                     into= ('OCTAHEDRAL', 'BCC24', 'CUBIQUE1', 'CUBIQUE2',
                            'ZIRCONIUM', 'UNIAXIAL', 'USER'),),
                  b_util =BLOC (condition= " FAMI_SYST_GLIS==' UTILISATEUR'
                  ",
                   TABL_SYST_GLIS =SIMP (statut=' f', typ=table_sdaster,),),),

    # case of a behavior of the type DD
    b_dd_cc=BLOC (condition= " ECOULEMENT==' MONO_DD_CC'",
                  FAMI_SYST_GLIS=SIMP   (statut='   f',   typ='   TXM',   into=
                  ('CUBIQUE1', 'USER',),
                  b_util=BLOC (condition= " FAMI_SYST_GLIS==' UTILISATEUR'",
                  TABL_SYST_GLIS=SIMP (statut=' f', typ=table_sdaster),),),),

MATR_INTER =SIMP (statut=' f', typ=table_sdaster, max=1,),

ROTA_RESEAU=SIMP  (statut='  f',  typ='  TXM',  max=1,  into=  ('NOT',  'POST',
'CALC'),
                  defaut='  NON',  fr=  "  rotation  of  network:  NOT,  POST,
                  CALC"),

POLYCRYSTAL =FACT (statut=' f', max=' ** ',
              regles= (UN_PARMI ('ANGL_REP', 'ANGL_EULER'),),

   MONOCRISTAL=SIMP (statut=' o', typ=compor_sdaster, max=1),
   FRAC_VOL  =SIMP (statut=' o', typ=' R', fr= " voluminal fraction "),
   ANGL_REP=SIMP (statut=' f', typ=' R', max=3, fr= " nautical angles in
   degrees"),
   ANGL_EULER=SIMP (statut=' f', typ=' R', max=3, fr= " angles of Euler in
   degrees"),

b_poly =BLOC (condition = "POLYCRYSTAL! =None",
       MU_LOCA     =SIMP (statut=' o', typ=' R', max=1),
          LOCALISATION=SIMP  (statut='  f',  typ='  TXM',  max=1,  into=  ('BZ',
'BETA',),
       fr=tr ("To give the name of the rule of localization")),
       b_beta =BLOC (condition = "LOCALISATION==' BETA'",
                 DL            =SIMP (statut=' o', typ=' R', max=1),
                 DA            =SIMP (statut=' o', typ=' R', max=1),),
          ),
```

The architecture of the routines OP5901 and OP5902 is simple, and consists in filling the structure of data sd_compor, intended to prepare calculations. For that, several information is deduced from the data of the user:

- For the monocrystal:
  - the number of systems of slip, is by calling on the routine **LCMMSG,** who defines the families of systems of slip pre-established, that is to say by reading the table provided for each family (which itself is stored in sd_compor)
  - the number of internal variables which results amongst systems from slips total, which will be associated with the behavior MONOCRYSTAL in BEHAVIOR.

- For the polycrystal:
  - various monocrystals relative to each grain, with the voluminal fraction and the orientation
  - the rule of localization and its parameters.
  - The number of internal variables total, deduced from the monocrystals and amongst grains, which will be associated with the behavior POLYCRYSTAL in COMPORTMENT.

The addition of a crystalline behavior is thus reduced, in DEFI_COMPOR, with the modification of the catalogue for the part MONOCRYSTAL (for the syntactic checking). The addition of a family of systems of slip also does not represent a modification of the catalogue of DEFI_COMPOR, with possible blocks to manage the possibilities of association between laws of flow and families of systems of slip.

For the part POLYCRYSTAL, the addition of a crystalline behavior does not modify the catalogue of DEFI_COMPOR, One of the only modifications would consist of the addition of a rule of localization.

One can print the structure of data produced using IMPR_CO.

3 Architecture of integration

3.1 Catalogues

crystalline laws of behavior are usable in C_COMPORTEMENT.capy via RELATION = 'MONOCRYSTAL' or RELATION=' POLYCRISTAL'. Relative data with the specific crystalline laws sotn as for it defined in sd_compor, resulting from DEFI_COMPOR and provided under the keyword COMPOR in c_comportement.capy.

catapy.commun/c_comportement.capy

```
b_monox     = BLOCK (condition = "RELATION == 'MONOCRYSTAL'",
                fr=tr ("SD resulting from DEFI_COMPOR"),
          COMPOR =SIMP (statut=' o', typ=compor_sdaster, max=1),),

b_polyx     = BLOCK (condition = "RELATION == 'POLYCRYSTAL'",
                fr=tr ("SD resulting from DEFI_COMPOR"),
          COMPOR =SIMP (statut=' o', typ=compor_sdaster, max=1),),
```

The catalogues of the laws of behavior are:

bibpyt/Behavior/monocristal.py

```
from cata_comportement LoiComportement importation
law = LoiComportement (
   name            = 'MONOCRYSTAL',
Doc. = """ This model makes it possible to describe the behavior of a
monocrystal whose relations of behavior are provided via the concept compor,
resulting from DEFI_COMPOR. The number of internal variables is function of
the choices carried out in DEFI_COMPOR; for more precise details to see
[R5.03.11]. """,
   num_lc          = 32,
   nb_vari         = 0,
   nom_vari        = Nun,
   mc_mater        = Nun,
```

```
   modeling   = ('3D', 'AXIS', 'D_PLAN'),
   deformation    = ('SMALL', 'PETIT_REAC', 'SIMO_MIEHE'),
   nom_varc = ('TEMP'),
   algo_inte = ('NEWTON', 'NEWTON_RELI', 'RUNGE_KUTTA', 'NEWTON_PERT',),
   type_matr_tang = ('DISTURBANCE', 'CHECKING'),
   properties     = Nun, )
```

bibpyt/Behavior/polycristal.py

```
FRomanian cata_comportement LoiComportement importation
law = LoiComportement (
   name            = 'POLYCRYSTAL',
Doc. = """ homogenized polycrystalline Behavior, defined by DEFI_COMPOR """,
   num_lc          = 37,
   nb_vari         = 0,
   nom_vari        = Nun,
   mc_mater        = Nun,
   modeling   = ('3D', 'AXIS', 'D_PLAN'),
   deformation    = ('SMALL', 'PETIT_REAC', 'GROT_GDEP'),
   nom_varc        = ('TEMP'),
   algo_inte         = ('RUNGE_KUTTA'),
   type_matr_tang = ('DISTURBANCE', 'CHECKING'),
   properties     = Nun,)
```

3.2 Routines of integration of the laws

The integration of the crystalline laws of behavior is done as for all the laws of behavior on the level of each point of integration of the finite elements for the nonlinear options (FULL_MECA, RAPH_MECA, RIGI_MECA_TANG).

The routine LC0032 (MONOCRYSTAL) is called for each point integration by the routines following:

- if DEFORMATION=' PETIT' or = ' PETIT_REAC':
  - in 3D: TE0139/NMPL3D/NMCOMP/REDECE/LC0000/LC0032
  - in 2D: TE0100/NMPL2D/NMCOMP/REDECE/LC0000/LC0032
- if DEFORMATION=' SIMO_MIEHE':
  - in 2D or 3D, NMPL2D or NMPL3D are replaced by NMGPFI

The routine LC0032 fact call following the algorithm chosen, is with PLASTI (implicit integration by the method of Newton or an alternative) that is to say with NMVPRK (integration by Runge-Kutta).

The routine LC0037 (POLYCRYSTAL) is called for each point integration by the following routines:

- if DEFORMATION=' PETIT' or = ' PETIT_REAC':
  - in 3D: TE0139/NMPL3D/NMCOMP/REDECE/LC0000/LC0032
  - in 2D: TE0100/NMPL2D/NMCOMP/REDECE/LC0000/LC0032

The routine LC0037 fact call with NMVPRK (integration by Runge-Kutta).

3.3 Recovery of the coefficients material

Some is the selected type of integration (implicit or clarifies) the recovery of the characteristics material and behavior, resulting from DEFI_MATERIAU and DEFI_COMPOR, is done via the routine LCMMAT, called by LCMATE, for MONOCRYSTAL , and by the routine LCMMAP, also called by LCMATE, for POLYCRYSTAL.

These routines have several functions:

- Recovery of the values of the keyword defining the parameters, mainly using the general routine RCVALB, and storage in two tables (different only if the coefficients depend on the temperature): MATERD defining the parameters in the previous moment, i.e. the beginning of the step of time, and MATERF at the current moment, therefore at the end of the step of time). These tables make it possible to pass the parameters material to the routines of resolution;
- Reading of sd_compor (exit of DEFI_COMPOR) and information storage (families of systems of slip, matrix of interaction,…) in tables used at the time of the resolution.

Architecture of LCMMAT:

LCMMJV : reading of sd_compor exit of DEFI_COMPOR
for each family of systems:
LCMMSG provides the number of systems of slip
LCMMJS (if it is about a family “user”)
LCMAFL recover the coefficients material relating to the flow
LCMHSR+LCMHDD : specific call in this routine for MONO_DD_KR
LCMAEC coefficients material relating to kinematic work hardening,
LCMAEI coefficients material relating to isotropic work hardening,
LCMHSR: calculation or reading of the matrix of interaction
DMAT3D, D1MA3D : operator of elasticity and his reverse
CALCMM LCMMSG. : calculation and storage of the tensors of orientation of all the systems, defined in the total reference mark, to optimize the performances.

In the case of a new single-crystal behavior, it is enough a priori to intervene in the routines LCMAFL, LCMAEI and possibly LCMAEC, by adding in each one of these routines the block of instructions necessary to the recovery of the coefficients material of this behavior.

It is also advisable to allot a number to him: indeed, to optimize the performances, it is preferable to read and compare entireties rather than these character strings; in the routines of integration, rather than to test:

```
IF (NECOUL.EQ. 'MONO_VISC1') THEN.
```
one will test:
```
IF (NUCOUL.EQ.1) THEN….
```

The nomenclature of the numbers of laws of flow is defined in LCMAFL :

Name of the law of flow	Associated number
MONO_VISC1	1
MONO_VISC2	2
MONO_DD_KR	4
MONO_DD_CFC	5
MONO_DD_CFC_IRRA	8
MONO_DD_FAT	6
MONO_DD_CC	7
MONO_DD_CC_IRRA	7

Table 3.3-1

The numbers of laws of isotropic work hardening are defined in LCMAEI :

Name of the law of flow	Associated number

MONO_ISOT1	1
MONO_ISOT2	2
MONO_DD_CFC	3
MONO_DD_CFC_IRRA	8
MONO_DD_FAT	4
MONO_DD_CC	7
MONO_DD_CC_IRRA	7

Table 3.3-2

The numbers of laws of isotropic work hardening are defined in LCMAEC :

Name of the law of flow	Associated number
MONO_CINE1	1
MONO_CINE2	2

Table 3.3-3

Architecture of LCMMAP:
Reading of sd_compor of standard polycrystal

For each behavior monocrystal (5 at the most) used by the whole of the grains
for each family of systems: reading of the characteristics like LCMMAT
LCMMSG provides the number of systems of slip
LCMAFL recover the coefficients material relating to the flow
LCMAEC coefficients material relating to kinematic work hardening,
LCMAEI coefficients material relating to isotropic work hardening, and stamps interaction
DMAT3D, D1MA3D : operator of elasticity and his reverse.

Storage of the relative information to the monocrystals used for each phase in specific tables (nbcomm, coeft/materf, cpmono, described at the end of this document).

Right before the call to the resolution clarifies by Runge-Kutta (routine GERPAS), call to the specific routine CALCMS allowing to store in a single table systems of slip relative to all the grains, defined in total reference mark, to optimize the performances.

3.4 Explicit integration – diagram of RUNGE - KUTTA

3.4.1 Case of the monocrystal:

It is the way fastest (but less optimal than implicit integration, for a system of a few tens of equations) to introduce a new single-crystal behavior in small deformations: it is enough to write the derivative of the internal variables in routine LCMMON, called by RDIF01.

The routine LCMMON is to calculate the derivative of the internal variables. One must solve a system of $6+3\times n_s$ differential equations, of the type (cf [R5.03.11]): $\frac{dY}{dt}=F(Y,t)$, where Y represent the whole of the internal variables: $Y=\begin{cases}\alpha_s\\ \gamma_s\\ p_s\\ E^{vp}\end{cases}$

The system of differential equations to solve is:

- $\dot{\boldsymbol{\varepsilon}}^{vp}=\sum_s \boldsymbol{\mu}_s \dot{\gamma}_s$ 6 equations
- for each system of slip (on the whole of the families of systems) 3 relations:
  - $\dot{\gamma}_s=\dot{p}_s\left(\tau_s(\boldsymbol{\sigma}),\alpha_s,\gamma_s,R_s(p)\right)\eta(\tau_s,\alpha_s)$ where $\eta(\tau_s,\alpha_s)=\pm 1$
  - $\dot{\alpha}_s=h(\tau_s,\alpha_s,\gamma_s,p_s)$
  - $R_s(p)$

  where $\tau_s=\boldsymbol{\mu}_s:\boldsymbol{\sigma}$ where $\boldsymbol{\sigma}$ is deduced from the relation: $\boldsymbol{\sigma}=\boldsymbol{\Lambda}\left(\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}^{vp}\right)$

In practice, the resolution is carried out in the following way:

The routine LCMMON calculate the constraints by the relation of elasticity (isotropic or orthotropic)

CAL CALSIG (...) what provides the tensor SIG = $\boldsymbol{\sigma}$

Then it calculates them $6+3\times n_s$ derived resulting from the differential equations above, and stores them in a table DVIN :

- Buckle on the families of systems of slip:

C IFA=1, NBFSYS
…
Recovery amongst systems of slip
CAL LCMMSG (NOMFAM, NBSYS, 0, PGL, MS, NG, LG, 0, Q)

Buckle on the systems of slip of the family IFA:

C IS=1, NBSYS

CAL LCMMSG (.) second reading tensor of orientation MuS

CALCULATION OF THE REDUCED CISSION
TAUS= SIG (I) *MuS (I) for i=1,6 $\tau_s=\boldsymbol{\mu}_s:\boldsymbol{\sigma}$

CAL LCMMFI (=> RP) **routine of calculation of isotropic work hardening** $R_s(p)$

CAL LCMMFE (=> DGAMMA, DP) **routine of calculation of the flow** $\dot{\gamma}_s, \dot{p}_s$

CAL LCMMEC (=> DALPHA) **routine of calculation of kinematic work hardening** $\dot{\alpha}_s$

Calculation of the total viscoplastic deformation
C ITENS=1,6

```
            DEVI (ITENS) =DEVI (ITENS) +MUS (ITENS) *DGAMMA   ε^vp = Σ_s μ_s γ̇_s
          ENDDO

          storage of the derivative of the internal variables for the system of slip IS
          DVIN (NUVI-2) =DALPHA
          DVIN (NUVI-1) =DGAMMA
          DVIN (NUVI  ) =DP

        ENDDO

    ENDDO

    storage of the tensor derived from the viscoplastic deformation.
    C ITENS=1,6
        DVIN (ITENS) = DEVI (ITENS)
    ENDDO
```

The algorithm of Runge-Kutta then manages the integration of these differential equations, by controlling the error, and by refining the step of time until obtaining an error lower than the required precision (RESI_INTE_RELA) [R5.03.14].

A priori, the structure of the routine LCMMON, does not have to evolve at the time of the addition of a new single-crystal behavior; only routines LCMMFI, LCMMFE and LCMMEC are to be modified.

They are built in the following way:

```
LCMMFI
C----------------------------------------------------------------------
C     FOR A NEW TYPE OF ISOTROPIC WORK HARDENING, TO ADD A BLOCK IF
C----------------------------------------------------------------------
C     IF (NECRIS.EQ. 'MONO_ISOT1') THEN
      IF (NUEISO.EQ.1) THEN
       ......
            RP=…
C     ELSEIF (NECRIS.EQ. 'MONO_ISOT2') THEN
      ELSEIF (NUEISO.EQ.2) THEN
      ............
         RP=…

C     ELSEIF (NECRIS.EQ. 'MONO_DD_CFC') THEN
      ELSEIF (NUEISO.EQ.3) THEN
         ..........
         RP=MU*SQRT (RP) *CEFF
      ENDIF
```

Notice : one uses here the numbers associated with each type of behavior rather than the names, to optimize the performances.

In the same way the structure of the routine LCMMFC is :

```
C----------------------------------------------------------------------
C   FOR A NEW TYPE OF KINEMATIC WORK HARDENING, TO ADD A BLOCK IF
C----------------------------------------------------------------------
C     IF (NECRCI.EQ. 'MONO_CINE1') THEN
      IF (NUECIN.EQ.1) THEN
          DALPHA=…
C     ELSEIF (NECRCI.EQ. 'MONO_CINE2') THEN
      ELSEIF (NUECIN.EQ.2) THEN
```

```
      ENDIF
```

And in a similar way, the structure of the routine LCMMFC is :

```
C-----------------------------------------------------------
C     FOR A NEW TYPE OF FLOW, TO CREATE A BLOCK IF
C-----------------------------------------------------------
C     IF (NECOUL.EQ. 'MONO_VISC1') THEN
      IF (NUECOU.EQ.1) THEN
          DP=…
          DGAMMA=…
      ELSEIF (NUECOU.EQ.2) THEN
      ENDIF
```

Notice : Routines LCMMFE, LCMMFI, LCMMFC are also used by the explicit integration of the polycrystal and implicit integration. This simplifies the implementation of a new crystalline behavior.

3.4.2 Case of the polycrystal:

The integration of the polycrystal rests largely on that of the monocrystal: there one calculates still the derivative of the internal variables for each monocrystal of each grain g, in routine LCMMOP, called by RDIF01.

One must solve a system of $6+n_g\left(6+3\times n_s(g)\right)$ differential equations, of the type:

- for each grain defined by an orientation and a proportion f_g, a relation of localization of the constraints, general form:

$$\boldsymbol{\sigma_g}=L\left(\boldsymbol{\Sigma},\boldsymbol{E^{vp}},\varepsilon_g^{vp},\beta_g\right) \quad \text{with,} \quad \boldsymbol{\Sigma}=\boldsymbol{\Lambda}(\boldsymbol{\Lambda}^{-1})\boldsymbol{\Sigma}^{-}+\boldsymbol{\Lambda}(\Delta\boldsymbol{E}-\Delta\boldsymbol{E^{th}}-\Delta\boldsymbol{E^{vp}})$$

- for each one of $n_s(g)$ systems of slip of each grain g, 3 relations:
  - $\dot{\gamma}_s=\dot{p}_s\left(\tau_s(\boldsymbol{\sigma}),\alpha_s,\gamma_s,R_s(p)\right)\eta(\tau_s,\alpha_s)$ where $\eta(\tau_s,\alpha_s)=\pm1$
  - $\dot{\alpha}_s=h\left(\tau_s,\alpha_s,\gamma_s,p_s\right)$
  - $R_s(p)$
- at the level of grain, lcalcul of the plastic deformation: $\dot{\boldsymbol{\varepsilon}}_{\boldsymbol{g}}^{\boldsymbol{vp}}=\sum_s \boldsymbol{\mu}_s\dot{\gamma}_s$
- calculation of the macroscopic plastic deformation: $\dot{E}^{vp}=\sum_g f_g\dot{\varepsilon}_g^{vp}$

In practice, the resolution is carried out in the following way:

The routine LCMMOP calculate the constraints by the relation of elasticity (isotropic or orthotropic)
CAL CALSIG (...) what provides the tensor SIG ($\boldsymbol{\Sigma}$)

Then it calculates them $6+n_g\left(6+3\times n_s(g)\right)$ derived resulting from the differential equations above, and stores them in a table DVIN:

- Buckle on the grains:

C IGRAIN=1, NGRAIN
CAL LCLOCA () relation of localization allowing to calculate SIGG $(\boldsymbol{\sigma_g})$

- Buckle on the families of systems of slip:

C IFA=1, NBFSYS
…
Recovery amongst systems of slip

```
    CAL LCMMSG (...)

    Buckle on the systems of slip of the family IFA:
    C IS=1, NBSYS

        CAL LCMMSG (.)    second reading of the tensor of orientation MuS (IGRAIN,
IS)

        CALCULATION OF THE REDUCED CISSION
        TAUS= SIGG (I) *MuS (I) for i=1,6

         CAL LCMMFI (=> RP)       routine of calculation of isotropic work
hardening

         CAL LCMMFE (=> DGAMMA, DP)          routine of calculation of the flow

         CAL LCMMEC (=> DALPHA) routine of calculation of kinematic work
hardening

         Calculation of the total viscoplastic deformation
        C ITENS=1,6
C          DEVG (ITENS) =DEVG (ITENS) +MUS (ITENS) *DGAMMA
(DEVG=ε_g^vp)
        ENDDO

        storage of the derivative of the internal variables for the system of slip IS
        DVIN (NUVI-2) =DALPHA
        DVIN (NUVI-1) =DGAMMA
        DVIN (NUVI  ) =DP

    ENDDO

ENDDO
homogenisation of the viscoplastic deformations
C I=1,6
   DEVI (I) =DEVI (I) +FV*DEVG (I)
ENDDO
storage of the tensor derived from the viscoplastic deformation.
C ITENS=1,6
   DVIN (ITENS) = DEVI (ITENS)
ENDDO

The seventh internal variable contains the cumulated equivalent viscoplastic deformation
DVIN (7) = DVINEQ
```

The algorithm of Runge-Kutta then manages the integration of these differential equations, by controlling the error, and by refining the step of time until obtaining an error lower than the required precision (RESI_INTE_RELA) [R5.03.14].

The structure of the routine LCMMOP, does not have to evolve at the time of the addition of a new single-crystal behavior; only routines LCMMFI, LCMMFE and LCMMEC are to be modified (what is already made in theory for the integration of the monocrystal). An additional modification is to be carried out in the routine LCLOCA at the time of the addition of a new rule of localization. Lastly, for certain postprocessings specific to the level of the point of integration, it is necessary to intervene in the routine LCDPEQ.

3.5 Implicit integration of the monocrystal by Newton in PLASTI

One integrates this time the single-crystal behavior by a method of Newton. This method is programmed in PLASTI [R5.03.14]. It is thus necessary to provide to this algorithm to write the system of equations to be solved in purely implicit form, in the following way: $\boldsymbol{R}(\boldsymbol{Y})=0$

$$R(Y)=\begin{cases} \boldsymbol{\Lambda}^{-1}\boldsymbol{\Sigma}-(\boldsymbol{\Lambda}_{-}^{-1})\boldsymbol{\Sigma}^{-}-(\Delta\boldsymbol{E}-\Delta\boldsymbol{E}^{\mathrm{th}}-\Delta\boldsymbol{E}^{\mathrm{vp}}) \\ \Delta\boldsymbol{E}^{\mathrm{vp}}-\sum_s \boldsymbol{\mu}_s\Delta\gamma_s \\ n_s\begin{cases}\Delta\alpha_s-h(\tau_s^+,\alpha_s^+,\gamma_s^+,p_s^+)\\ \Delta\gamma_s-g(\tau_s^+,\alpha_s^+,\gamma_s^+,p_s^+)\\ \Delta p_s-f(\tau_s^+,\alpha_s^+,\gamma_s^+,p_s^+)\end{cases} \end{cases} =\mathbf{0}$$

It is a system of $6+6+3n_s$ nonlinear equations (of the same size than that which is integrated by the method of Runge-Kutta into explicit). But it is not the actually solved system: in order to optimize the performances, one solves in fact a system of equations reduced, of size $6+n_s$, built in the following way [R5.03.11]:

- In the expression of the 6 components of the tensor of the constraints, $\Delta\boldsymbol{E}^{vp}$ can be expressed according to $\sum_s \boldsymbol{\mu}_s\Delta\gamma_s$ thus 6 equations can be eliminated from the total system to solve.
- Like $\Delta p_s=|\Delta\gamma_s|$, the equation in Δp_s can be eliminated
- for all the crystalline behaviors currently considered, is $\Delta\alpha_s$ express yourself directly according to $\Delta\gamma_s$, in the case of kinematic work hardening, that is to say $\Delta\gamma_s$ express yourself according to $\Delta\alpha_s$ who then represents the density of dislocation (except for a factor) for the behaviors of the type MONO_DD_*. There is thus only one unknown factor per system of slip: $\Delta\beta_s$, which represents is $\Delta\gamma_s$, that is to say $\Delta\alpha_s$

The following system is thus obtained:

- **Small deformations:**

$$\begin{aligned}&\boldsymbol{R_1}(\boldsymbol{\sigma},\Delta\beta)=\boldsymbol{\Lambda}^{-1}.\Delta\boldsymbol{\sigma}-\Delta\boldsymbol{\varepsilon}+\Delta\boldsymbol{\varepsilon}^{th}+\sum_s \Delta\gamma_s\boldsymbol{\mu}_s=0\\ &\boldsymbol{R_2}(\boldsymbol{\sigma},\Delta\beta)=\Delta\beta_s-k_s(\tau_s(\boldsymbol{\sigma}),\Delta\beta)=0\end{aligned}$$ where the unknown factor is: $\boldsymbol{Y}=\begin{bmatrix}\boldsymbol{\sigma}\\ \Delta\beta\end{bmatrix}$

with $\tau_s(\boldsymbol{\sigma})=\boldsymbol{\sigma}:\boldsymbol{\mu}_s$

- **Great deformations**

$$\begin{aligned}&\boldsymbol{R_1}(\boldsymbol{S},\Delta\beta)=\boldsymbol{\Lambda}^{-1}.\boldsymbol{S}-\frac{1}{2}\left(\boldsymbol{F}^{\mathrm{eT}}\boldsymbol{F}^{\mathrm{e}}-\boldsymbol{I}_d\right)=0\\ &\boldsymbol{R_2}(\boldsymbol{S},\Delta\beta)=\Delta\beta_s-k_s(\tau_s(\boldsymbol{S}),\Delta\beta)=0\end{aligned}$$ where the unknown factor is: $\boldsymbol{Y}=\begin{bmatrix}\boldsymbol{S}\\ \Delta\beta\end{bmatrix}$

with $\tau_s(\boldsymbol{S})=\left[\left(2\boldsymbol{\Lambda}^{-1}\boldsymbol{S}+\boldsymbol{I}_d\right)\boldsymbol{S}\right]:\boldsymbol{m}_s\otimes\boldsymbol{n}_s$ and $\boldsymbol{F}_{n+1}^{e}=\Delta\boldsymbol{F}\,\boldsymbol{F}_n^{e}\left(\Delta\boldsymbol{F}^{p}(\Delta\gamma_s)\right)^{-1}$

And, according to the behavior considered,

- $\Delta\gamma_s=\Delta p_s(\tau_s,\Delta\beta_s)\xi_s$ and $\xi_s=\frac{\tau_s}{|\tau_s|}$ *ou* $\frac{\tau_s-f(\alpha)}{|\tau_s-f(\alpha)|}$ corresponds with sign of the flow

- $\Delta\beta_s$ represent is the plastic increment of slip $\Delta\gamma_s$, for the laws MONO_VISC* , that is to say the variation of density of dislocations $\Delta\omega_s$ for the laws MONO_DD_ *

Notice : the extraction of the unknown factors of the system starting from the internal variables (which remain 3 per system of slip) is done in the routine LCAFYD. Conversely, after resolution by NEWTON, the calculation of the 3 internal variables per system of slip according to the basic variable used in the resolution is done in the routine LCPLNF.

The general shape of the algorithm solved by Newton is

$$Y_{k+1}=Y_k-(\frac{dR}{dY_k})^{-1}R(Y_k)$$

Thus should be defined the initial values ΔY_0 (0 by default, in the routine LCMMIN) , and to calculate the residue Y_k , like matrix jacobienne of the system: $\frac{dR}{dY_k}$

3.5.1 General architecture of PLASTI :

The algorithm of Newton used in PLASTI is described in [R5.03.14].

Reading of the coefficients material and storage in the objects NBCOMM, MATERF/MATERD, CPMONO

```
CAL LCMATE => LCMMAT, identical to RUNGE_KUTTA
```

Elastic prediction
```
   CAL LCELAS
```
Calculation of THRESHOLD
```
   CAL LCCNVX => routine LCMMVX evaluation of the threshold for MONOCRYSTAL.
```

Calculation of the solution élasto-visco-plastic by the method of Newton:
```
IF (THRESHOLD .GE. 0.D0) THEN
   CAL LCPLAS/CAL LCPLNL
ENDIF
```

Calculation of the tangent operator:
```
IF (OPT .EQ. 'RIGI_MECA_TANG' .OR. OPT .EQ. 'FULL_MECA') THEN
   CAL LCJPLC => CAL LCMMJP
ENDIF
```

Notice : The tangent operator is calculated automatically according to the matrix jacobienne local system of equations [R5.03.11]. This is carried out into small and great deformations in the routine LCMMJP. There is thus a priori nothing to modify for this calculation at the time of the addition a new crystalline behavior.

The routine LCCNVX allows to detect if the threshold is crossed for at least a system of slip. It calls in the case of the monocrystal the routine LCMMVX. Its structure is the following one:
```
SEUIL=0.D0
C IFA=1, NBFSYS
   C IS=1, NBSYS
      CAL LCMMFI
C     VISCOPLASTIC FLOW
      CAL LCMMFE => DP  calculated starting from the elastic prediction
      IF (DP.GT.0.D0) SEUIL=1.D0
   ENDDO
ENDDO
```

The same routines are thus used LCMMFE and LCMMFI that for explicit and implicit integration.

The routine LCPLNL realize **the loop of Newton**. Its structure (generic with the unit of the laws of behavior under PLASTI) is the following one:

LCPLNL
- LCAFYD (extraction of the internal variables useful for the reduced system)
- LCINIT => LCMMIN : initialization of ΔY_0 , 0 by default
- LCRESI => LCMMRE : calculation of the residue
  - That is to say LCJACB => LCMMJA : calculation of the matrix jacobienne
  - That is to say (if ALGO_INTE=NEWTON_PERT) LCJACP calculation of the matrix jacobienne by disturbance, which calls LCRESI
- MGAUSS resolution
- LCRELI linear research (if ALGO_INTE=NEWTON_RELI), which calls LCRESI…
- LCCONV => LCMMCV convergence criteria
- LCPLNF => LCDPEC calculation of all the internal variables.

3.5.2 Calculation of the residue

The routine LCMMRE calculate the residue. Its structure is the following one:

```
C IFA=1, NBFSYS
   C IS=1, NBSYS
      CALTAU calculation of τ_s (small or great deformations)
      LCMMLC calculate quantities relating to the system of slip:
         CAL LCMMFI (=> RP)        routine of calculation of isotropic work hardening
         CAL LCMMFE (=> DGAMMA, DP)        routine of calculation of the flow
         CAL LCMMEC (=> DALPHA) routine of calculation of kinematic work hardening
      calculation of k_s and storage in R_2(σ,Δβ)=Δβ_s−k_s(τ_s(σ),Δβ)

      In small deformations: total viscoplastic deformation
      C ITENS=1,6
         DEVI (ITENS) =DEVI (ITENS) +MUS (ITENS) *DGAMMA
      ENDDO
      In great deformations, calculation of the terms necessary to
ΔF^p(Δγ_s)
   ENDDO
ENDDO
```

- in small deformations, calculation of $\boldsymbol{R_1}(\boldsymbol{\sigma},\Delta\beta)=\boldsymbol{\Lambda^{-1}}.\Delta\boldsymbol{\sigma}-\Delta\boldsymbol{\varepsilon}+\Delta\boldsymbol{\varepsilon}^{th}+\sum_s \Delta\gamma_s\boldsymbol{\mu}_s=0$
- in great deformations

  CALCFE calculation of $\boldsymbol{F}^e_{n+1}=\Delta\boldsymbol{F}\,\boldsymbol{F}^e_n\left(\Delta\boldsymbol{F}^p(\Delta\gamma_s)\right)^{-1}$

  LCGRLA calculation of $\boldsymbol{E}^e_{GL}=\frac{1}{2}(\boldsymbol{F}^{eT}\boldsymbol{F}^e-\boldsymbol{I}_d)\quad \boldsymbol{S}=\boldsymbol{\Lambda}:\boldsymbol{E}^e_{GL}$

  and $\boldsymbol{R_1}(\boldsymbol{S},\Delta\beta)=\boldsymbol{\Lambda^{-1}}.\boldsymbol{S}-\frac{1}{2}\left(\boldsymbol{F}^{eT}\boldsymbol{F}^e-\boldsymbol{I}_d\right)$

It is thus noted there that one uses still same the routines as for explicit integration: LCMMFI, LCMMFE, LCMMEC are a priori the only routines to be modified at the time of the addition of a behavior, that it is into small or great deformations, for the calculation of the residue, during implicit integration.

3.5.3 Calculation of the matrix jacobienne

The routine LCMMJA calculate the matrix jacobienne (except if ALGO_INTE= 'NEWTON_PERT').

Its structure is the following one:

```
C IFA=1, NBFSYS
   C IS=1, NBSYS
      LCMMJB : calculation of the derived terms
         LCMMJ2 : calculation of the derived terms for MONO_DD_KR
            LCMMJD : calculation of the derived terms for MONO_DD_CFC, MONO_DD_CC (more
_IRRA)
         LCMMJ1 : calculation of the derived terms for MONO_VISC1, MONO_VISC2
   ENDDO
ENDDO
```

The derivative of these equations for the calculation of the matrix jacobienne can be written in a general way:

HP		GDEF	
$J_{11}=\frac{\partial \boldsymbol{R_1}(\sigma,\Delta\beta)_i}{\partial \boldsymbol{\sigma_j}}$	$J_{12}=\frac{\partial \boldsymbol{R_1}(\sigma,\Delta\beta)_i}{\partial \beta_s}$	$J_{11}=\frac{\partial \boldsymbol{R_1}(\boldsymbol{S},\Delta\beta)_i}{\partial \boldsymbol{S_j}}$	$J_{12}=\frac{\partial \boldsymbol{R_1}(\boldsymbol{S},\Delta\beta)_i}{\partial \beta_s}$
$J_{21}=\frac{\partial \boldsymbol{R_2}(\sigma,\Delta\beta)_s}{\partial \boldsymbol{\sigma_i}}$	$J_{22}=\frac{\partial \boldsymbol{R_2}(\sigma,\Delta\beta)_s}{\partial \beta_r}$	$J_{21}=\frac{\partial \boldsymbol{R_2}(\boldsymbol{S},\Delta\beta)_s}{\partial \boldsymbol{S_i}}$	$J_{22}=\frac{\partial \boldsymbol{R_2}(\boldsymbol{S},\Delta\beta)_s}{\partial \beta_r}$

Table 3.5.3-1

Terms intervening in each submatrix narrower terms to each behavior have in common, which are calculated in the routines LCMMJ* [R5,03,11 appendix 5]:

- $\frac{\partial \Delta\gamma_s}{\partial \tau_s}=\frac{\partial \Delta p_s}{\partial \tau_s}\xi_s$,
- $\frac{\partial \Delta\gamma_r}{\partial \Delta\beta_s}=\frac{\partial \Delta p_r}{\partial \Delta\beta_s}\xi_r$,
- $\frac{\partial k_s}{\partial \tau_s}$
- $\frac{\partial k_r}{\partial \Delta\beta_s}$

In the case of a new behavior, it is thus necessary is to add the calculation of these terms derived in a routine LCMMJ* existing, that is to say to add a news of it.

Notice : for a first test, one can do without the calculation of the matrix jacobienne, by using the automatic construction of the matrix jacobienne (by disturbance, ALGO_INTE=' NEWTON_PERT'). So it is very fast to introduce a new crystalline behavior into the environment PLASTI: it is enough to calculate the residue in the routine LCMMRE, called by LCRESI. On the other hand, time calculation will be optimized only with one programmed matrix jacobienne.

3.5.4 Convergence criteria and postprocessings

The convergence criteria are generic a priori, and does not depend on the behavior, but can be possibly modified:

- LCCONV => LCMMCV convergence criteria

The last routine to be modified (possibly, if one wants to calculate and add to the internal variables at exit of the values useful for postprocessing) is:

- LCPLNF => LCDPEC who recomputes all the internal variables starting from the solution of the reduced system. She once again calls on the routine of behavior LCMMLC.

4 O bjets internal for the information storage

L reading of the coefficients material has is carried out only once by point of integration in the routines LCMMAT/LCMMAP . The quantities read (or calculated, in the case of the tensors of orientation) must be transmitted to the routines of integration. For that one uses specific tables: nbcomm, coeft/materf, cpmono, toutms…)

4.1 In the case of the monocrystal:

Storage of the relative information to the monocrystal in specific tables (nbcomm, coeft/materf, materd, cpmono,).

```
MATERF:  coefficients material has t+dt
          to subdue (*, 1) = characteristic   rubber bands
          to subdue (*, 2) = characteristic   plastics

MATERD:  coefficients material with T

NBCOMM (*, 3): POSITION OF THE COEFF FOR EACH  SYSTEM
table of entireties
             Column 1      Column 2      Colonne3
            _____________________________________________
 Line 1     1             Nb var.int.   Nb monocrystals
 for each family of systems of slip:
  family 1  ind coeff       ind coeff      ind coeff
             flow     ecr Iso       ecr cin
    .....
  ind means the index in MATERF/MATERD (*, 2)
  last line:          Nb of families

CPMONO: table of, copy character strings of the sd_compor.CPRK object
         by family of systems of slip
            Surname
            Name of material
            Name of the law of flow
            Name of the isotropic law of work hardening
            Name of the kinematic law of work hardening
```

TOUTMS real table of dimension (Many families, many systems, 6) containing all the tensors mus=sym (ms*ns) in HP, and all vectors ms and NS in great deformations.

4.2 In the case of the polycrystal:

STRUCTURE OF THE OBJECTS CREATE

```
COEFT (*): table of realities
             Many monocrystals
             index (in COEFT) of the beginning of the first monocrystal
```

```
              index beginning second monocrystal
              ….
              index beginning last monocrystal
              index of the parameters of localization
              Fv and 3 angles per phase
for each monocrystal different
      by family of system of slip
         Nb coeff flow
            numéo of the law of flow + coeff,
         Nb coeff nut isot + num_loi + coeff,
         Nb coeff ecou movies + num_loi + coeff
             then parameters of the law of localization

CPMONO (*): table of character strings
      name of the method of localization
      then, for each different material
      name of the monocrystal, many families SG, and,
         by family of systems of slip
            Surname
            Name of material
            Name of the law of éécoulement
            Name of the isotropic law of work hardening
            Name of the kinematic law of work hardening
            Name of the law of elasticity (ELAS or ELAS_ORTH)

NBCOMM (*, 3): table of entireties
              Column 1      Column 2      Colonne3
             _____________________________________
 Line 1      Nb phases      Nb var.int.   Nb monocrystals
                                           different
   for each phase G  Num line G    Ind CPMONO    ind tailcoat flight
   ...................
   ...................
   for each phase
   for the localization  index coeff   Nb param      0
   phase G              Nb fam G       0            NVIg
     … and for each family of systems of slip:
  family 1  ind coeff       ind coeff      ind coeff
              flow     ecr Iso       ecr cin
    .....
          (ind means the index in COEFT (*)


TOUTMS: ALL TENSORS OF ORIENTATION FOR ALL THEM                          SYSTEMS OF
SLIP OF ALL THE MONOCRYSTALS.
```