Architecture des comportements cristallins

Résumé :

Ce document décrit la structure de l'intégration des comportements cristallins (confer R5.03.11) et les actions à entreprendre pour ajouter un nouveau comportement cristallin, dans le but d'effectuer des calculs d'agrégats, ou des calculs homogénéisés à l'aide de STAT_NON_LINE et SIMU_POINT_MAT.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Table des Matières

1 Description générale des comportements cristallins....3
2 Architecture de DEFI_COMPOR....3
2.1 Illustration sur un exemple : test SSNV194....3
2.1.1 Modélisation A : un petit agrégat comportant 10 grains :....3
2.1.2 Modélisations B et C : polycristal comportant 10 grains :....4
2.2 Description de DEFI_COMPOR....5
2.2.1 Structure :....5
2.2.2 Ajout d'un comportement cristallin au catalogue de DEFI_MATERIAU / DEFI_COMPOR....5
3 Architecture de l'intégration....7
3.1 Catalogues....7
3.2 Routines d'intégration des lois....8
3.3 Récupération des coefficients matériau....8
3.4 Intégration explicite – schéma de RUNGE - KUTTA....10
3.4.1 Cas du monocristal :....10
3.4.2 Cas du polycristal :....12
3.5 Intégration implicite du monocristal par Newton dans PLASTI....14
3.5.1 Architecture générale de PLASTI :....15
3.5.2 Calcul du résidu....16
3.5.3 Calcul de la matrice jacobienne....17
3.5.4 Critère de convergence et post-traitements....18
4 O bjets internes pour le stockage des informations....18
4.1 Dans le cas du monocristal :....18
4.2 Dans le cas du polycristal :....19

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

1 Description générale des comportements cristallins

Un comportement cristallin utilise, en plus de DEFI_MATERIAU, DEFI_COMPOR.

Le nom du comportement dans COMPORTEMENT est générique : MONOCRISTAL ou POLYCRISTAL.

L'algorithme de résolution est au choix :
- pour le MONOCRISTAL : NEWTON, NEWTON_RELI, NEWTON_PERT et RUNGE_KUTTA
- pour le POLYCRISTAL : RUNGE_KUTTA

On décrit dans ce document :
- l'architecture de DEFI_COMPOR
- l'architecture de la résolution explicite à erreur contrôlée par Runge-Kutta (cf. [R5.03.14])
- l'architecture de la résolution implicite par Newton (environnement PLASTI cf. [R5.03.14])
- les objets internes utiles à la résolution.

Pour appréhender ce document, la lecture de R5.03.11 est vivement conseillée.

2 Architecture de DEFI_COMPOR

2.1 Illustration sur un exemple : test SSNV194

Partons d'un exemple (test ssnv194).

2.1.1 Modélisation A : un petit agrégat comportant 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,
                                    ),)
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2.1.2 Modélisations B et C : polycristal comportant 10 grains :

Un point matériel, 10 grains de fractions volumiques identiques (0.1), et d'orientations similaires à celles de la modélisation A (ce qui permet de retrouver la même solution moyenne) :

```
MONO1=DEFI_COMPOR(MONOCRISTAL=(... identique à la modélisation 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,),),
),

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

Modélisation B:(maillage comportant un seul élément)

…

```
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
                                    ),),);
```

Modélisation C:(point matériel)

```
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=...
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                                    );
```

2.2 Description de DEFI_COMPOR

2.2.1 Structure :

La routine OP0050 a pour but de produire la structure de données décrite dans [D4.06.24] : les objets composant cette SD sont différents suivant que l'on traite un monocristal (routine OP5901) ou un polycristal (routine OP5902).

2.2.2 Ajout d'un comportement cristallin au catalogue de DEFI_MATERIAU / DEFI_COMPOR

Si le nouveau comportement cristallin utilise des paramètres matériau différents de ceux qui sont déjà disponibles dans les mots-clés MONO_* de DEFI_MATERIAU il suffit d'introduire ces nouveaux paramètres de comportement, soit sous un seul mot-clé (cas des comportements MONO_DD_*, soit en séparant les coefficients relatifs à l'écrouissage isotrope, à l'écrouissage cinématique, et à l'écoulement (cf. MONO_ISOT*, MONO_CINE*, MONO_VISC*,) . Ces paramètres seront exploités dans l'intégration (routines LCMMAT, LCMMAP), et les mot-clés facteurs correspondants seront utilisés dans DEFI_COMPOR.

Exemple: catalogue de 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, unités : s**-1"),
TAU_F =SIMP(statut='o',typ='R',fr="en unite de contraintes ex 20 MPa"),
A =SIMP(statut='f',typ='R',defaut=0.13,fr="paramètre A, sans unité"),
B =SIMP(statut='f',typ='R',defaut=0.005,fr="paramètre B, sans unité"),
N =SIMP(statut='f',typ='R',defaut=200.,fr="paramètre n, sans unité"),
Y =SIMP(statut='o',typ='R',fr="en unité de longueur ex 2.5 A"),
ALPHA=SIMP(statut='f',typ='R',defaut=0.35,fr="paramètre alpha"),
BETA =SIMP(statut='o',typ='R',fr="paramètre b, en unite de longueur"),
...),
```

Ceci permet de décrire chaque coefficient, son caractère facultatif (avec une éventuelle valeur par défaut) ou obligatoire (pour plus de précisions, se reporter à [D5.01.01]) .

Le catalogue de DEFI_COMPOR est, suivant les cas :

```
MONOCRISTAL =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="type d'écoulement viscoplastique"),
    ELAS =SIMP(statut='f', typ='TXM',),

    # cas d'un comportement de type MONO_VISC*
    b_non_dd=BLOC(condition="ECOULEMENT=='MONO_VISC1'
                          or ECOULEMENT=='MONO_VISC2',
                  ECRO_ISOT=SIMP(statut='f', typ='TXM', max=1,
                     fr="Donner le type d'écrouissage isotrope"),
                  ECRO_CINE=SIMP(statut='f', typ='TXM', max=1,
                     fr="Donner type d'écrouissage cinématique"),
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                    FAMI_SYST_GLIS=SIMP(statut='f',typ='TXM',
                       into=('OCTAEDRIQUE','BCC24','CUBIQUE1','CUBIQUE2',
                             'ZIRCONIUM','UNIAXIAL','UTILISATEUR'),),
                    b_util =BLOC(condition="FAMI_SYST_GLIS=='UTILISATEUR' ",
                     TABL_SYST_GLIS =SIMP(statut='f', typ=table_sdaster,),),),

    # cas d'un comportement de type DD
    b_dd_cc=BLOC(condition="ECOULEMENT=='MONO_DD_CC'",
                    FAMI_SYST_GLIS=SIMP(statut='f',typ='TXM',into=('CUBIQUE1','
                    UTILISATEUR',),
                    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=('NON','POST','CALC'),
                    defaut='NON',fr="rotation de reseau : NON, POST, CALC"),

POLYCRISTAL =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="fraction volumique »),
    ANGL_REP=SIMP(statut='f',typ='R',max=3,fr="angles nautiques en degrés"),
    ANGL_EULER=SIMP(statut='f',typ='R',max=3,fr="angles d'Euler en degrés"),

b_poly =BLOC( condition = "POLYCRISTAL!=None",
       MU_LOCA     =SIMP(statut='o',typ='R',max=1),
       LOCALISATION=SIMP(statut='f', typ='TXM', max=1, into=('BZ', 'BETA',),
       fr=tr("Donner le nom de la règle de localisation")),
       b_beta =BLOC( condition = "LOCALISATION=='BETA'",
                  DL            =SIMP(statut='o',typ='R',max=1),
                  DA            =SIMP(statut='o',typ='R',max=1),),
          ),
```

L'architecture des routines OP5901 et OP5902 est simple, et consiste à remplir la structure de données sd_compor, destinée à préparer les calculs. Pour cela, plusieurs informations sont déduites des données de l'utilisateur :

- Pour le monocristal :
  - le nombre de systèmes de glissement, soit en faisant appel à la routine **LCMMSG,** qui définit les familles de systèmes de glissement pré-établies, soit en lisant la table fournie pour chaque famille (qui est elle-même stockée dans la sd_compor)
  - le nombre de variables internes qui se déduit du nombre de systèmes de glissements total, qui sera associé au comportement MONOCRISTAL dans COMPORTEMENT.

- Pour le polycristal :
  - les différents monocristaux relatifs à chaque grain, avec la fraction volumique et l'orientation
  - le règle de localisation et ses paramètres.
  - Le nombre de variables internes total, déduit des monocristaux et du nombre de grains, qui sera associé au comportement POLYCRISTAL dans COMPORTMENT.

L'ajout d'un comportement cristallin se réduit donc, dans DEFI_COMPOR, à la modification du catalogue pour la partie MONOCRISTAL (pour la vérification syntaxique). L'ajout d'une famille de systèmes de glissement se traduit également pas une modification du catalogue de DEFI_COMPOR, avec d'éventuels blocs pour gérer les possibilités d'association entre lois d'écoulement et familles de systèmes de glissement.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Pour la partie POLYCRISTAL, l'ajout d'un comportement cristallin ne modifie pas le catalogue de DEFI_COMPOR, Une des seules modifications consisterait en l'ajout d'une règle de localisation.

On peut imprimer la structure de données produite à l'aide de IMPR_CO.

3 Architecture de l'intégration

3.1 Catalogues

Les lois de comportement cristallines sont utilisables dans C_COMPORTEMENT.capy via RELATION = 'MONOCRISTAL' ou RELATION='POLYCRISTAL'. Les données relatives aux lois cristallines spécifiques sotn quant à elle définies dans sd_compor, issue de DEFI_COMPOR et fournie sous le mot-clé COMPOR dans c_comportement.capy.

catapy.commun/c_comportement.capy

```
          b_monox     = BLOC(condition = "RELATION == 'MONOCRISTAL' ",
                             fr=tr("SD issue de DEFI_COMPOR"),
                        COMPOR =SIMP(statut='o',typ=compor_sdaster,max=1),),

          b_polyx     = BLOC(condition = "RELATION == 'POLYCRISTAL' ",
                             fr=tr("SD issue de DEFI_COMPOR"),
                        COMPOR =SIMP(statut='o',typ=compor_sdaster,max=1),),
```

Les catalogues des lois de comportement sont :

bibpyt/Comportement/monocristal.py

```
from cata_comportement import LoiComportement
loi = LoiComportement(
   nom            = 'MONOCRISTAL',
doc = """Ce modèle permet de décrire le comportement d'un monocristal dont
les relations de comportement sont fournies via le concept compor, issu de
DEFI_COMPOR. Le nombre de variables internes est fonction des choix
effectués dans DEFI_COMPOR ; pour plus de précisions voir [R5.03.11].""",
   num_lc         = 32,
   nb_vari        = 0,
   nom_vari       = None,
   mc_mater       = None,
   modelisation   = ('3D','AXIS','D_PLAN'),
   deformation    = ('PETIT', 'PETIT_REAC', 'SIMO_MIEHE'),
   nom_varc = ('TEMP'),
   algo_inte = ('NEWTON','NEWTON_RELI','RUNGE_KUTTA','NEWTON_PERT',),
   type_matr_tang = ('PERTURBATION', 'VERIFICATION'),
   proprietes     = None, )
```

bibpyt/Comportement/polycristal.py

```
from cata_comportement import LoiComportement
loi = LoiComportement(
   nom            = 'POLYCRISTAL',
doc ="""Comportement poly-cristallin homogénéisé, défini par DEFI_COMPOR""",
   num_lc         = 37,
   nb_vari        = 0,
   nom_vari       = None,
   mc_mater       = None,
   modelisation   = ('3D','AXIS','D_PLAN'),
   deformation    = ('PETIT', 'PETIT_REAC', 'GROT_GDEP'),
   nom_varc       = ('TEMP'),
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
algo_inte         = ('RUNGE_KUTTA'),
type_matr_tang = ('PERTURBATION', 'VERIFICATION'),
proprietes     = None, )
```

3.2 Routines d'intégration des lois

L'intégration des lois de comportement cristallines se fait comme pour toutes les lois de comportement au niveau de chaque point d'intégration des éléments finis pour les options non linéaires (FULL_MECA, RAPH_MECA, RIGI_MECA_TANG).

La routine LC0032 (MONOCRISTAL) est appelée pour chaque point l'intégration par les routines suivantes :

- si DEFORMATION='PETIT' ou ='PETIT_REAC' :
  - en 3D : TE0139/NMPL3D/NMCOMP/REDECE/LC0000/ LC0032
  - en 2D : TE0100/NMPL2D/NMCOMP/REDECE/LC0000/ LC0032
- si DEFORMATION='SIMO_MIEHE' :
  - en 2D ou en 3D, NMPL2D ou NMPL3D sont remplacées par NMGPFI

La routine LC0032 fait appel suivant l'algorithme choisi, soit à PLASTI (intégration implicite par la méthode de Newton ou une variante) soit à NMVPRK (intégration par Runge-Kutta).

La routine LC0037 (POLYCRISTAL) est appelée pour chaque point l'intégration par les routines suivantes :

- si DEFORMATION='PETIT' ou ='PETIT_REAC' :
  - en 3D : TE0139/NMPL3D/NMCOMP/REDECE/LC0000/ LC0032
  - en 2D : TE0100/NMPL2D/NMCOMP/REDECE/LC0000/ LC0032

La routine LC0037 fait appel à NMVPRK (intégration par Runge-Kutta).

3.3 Récupération des coefficients matériau

Quelque soit le type d'intégration choisi (implicite ou explicite) la récupération des caractéristiques matériau et comportement, issues de DEFI_MATERIAU et DEFI_COMPOR, se fait par l'intermédiaire de la routine LCMMAT, appelée par LCMATE, pour le MONOCRISTAL , et par la routine LCMMAP, appelée également par LCMATE, pour le POLYCRISTAL.

Ces routines ont plusieurs fonctions :

- Récupération des valeurs des mot-clés définissant les paramètres, principalement à l'aide de la routine générale RCVALB, et stockage dans deux tableaux (différents seulement si les coefficients dépendent de la température) : MATERD définissant les paramètres à l'instant précédent, c'est à dire au début du pas de temps, et MATERF à l'instant actuel, donc à la fin du pas de temps). Ces tableaux permettent de passer les paramètres matériau aux routines de résolution ;
- Lecture de la sd_compor (issue de DEFI_COMPOR) et stockage des informations (familles de systèmes de glissement, matrice d'interaction, …) dans des tableaux utilisés lors de la résolution.

Architecture de LCMMAT :

LCMMJV : lecture de la sd_compor issue de DEFI_COMPOR
pour chaque famille de systèmes :
  LCMMSG fournit le nombre de systèmes de glissement
    LCMMJS (si il s'agit d'une famille « utilisateur »)
  LCMAFL récupère les coefficients matériau relatifs à l'écoulement
    LCMHSR+LCMHDD : appel spécifique dans cette routine pour MONO_DD_KR
  LCMAEC coefficients matériau relatifs à l'écrouissage cinématique,
  LCMAEI coefficients matériau relatifs à l'écrouissage isotrope,

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

LCMHSR : calcul ou lecture de la matrice d'interaction
DMAT3D, D1MA3D : opérateur d'élasticité et son inverse
CALCMM LCMMSG. : calcul et stockage des tenseurs d'orientation de tous les systèmes, définis dans le repère global, pour optimiser les performances.

Dans le cas d'un nouveau comportement monocristallin, il suffit a priori d'intervenir dans les routines LCMAFL, LCMAEI et éventuellement LCMAEC, en ajoutant dans chacune de ces routines le bloc d'instructions nécessaires à la récupération des coefficients matériau de ce comportement.

Il convient également de lui attribuer un numéro : en effet, pour optimiser les performances, il est préférable de lire et comparer des entiers plutôt que ces chaînes de caractères ; dans les routines d'intégration, plutôt que de tester :

```
IF (NECOUL.EQ.'MONO_VISC1') THEN..
```
on testera :
```
IF (NUCOUL.EQ.1) THEN....
```

La nomenclature des numéros de lois d'écoulement est définie dans LCMAFL :

Nom de la loi d'écoulement	Numéro associé
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

Tableau 3.3-1

Les numéros de lois d'écrouissage isotrope sont définis dans LCMAEI :

Nom de la loi d'écoulement	Numéro associé
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

Tableau 3.3-2

Les numéros de lois d'écrouissage isotrope sont définis dans LCMAEC :

Nom de la loi d'écoulement	Numéro associé
MONO_CINE1	1
MONO_CINE2	2

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Tableau 3.3-3

Architecture de LCMMAP :
Lecture de la sd_compor de type polycristal

Pour chaque comportement monocristal (5 au maximum) utilisé par l'ensemble des grains
pour chaque famille de systèmes : lecture des caractéristiques comme LCMMAT
LCMMSG fournit le nombre de systèmes de glissement
LCMAFL récupère les coefficients matériau relatifs à l'écoulement
LCMAEC coefficients matériau relatifs à l'écrouissage cinématique,
LCMAEI coefficients matériau relatifs à l'écrouissage isotrope, et matrice d'interaction
DMAT3D, D1MA3D : opérateur d'élasticité et son inverse.

Stockage des informations relatives aux monocristaux utilisés pour chaque phase dans des tableaux spécifiques (nbcomm, coeft/materf,cpmono, décrits en fin de ce document).

Juste avant l'appel à la résolution explicite par Runge-Kutta (routine GERPAS), appel à la routine spécifique CALCMS permettant de stocker dans un unique tableau les systèmes de glissement relatifs à tous les grains, définis en repère global, pour optimiser les performances.

3.4 Intégration explicite – schéma de RUNGE - KUTTA

3.4.1 Cas du monocristal :

C'est la façon la plus rapide (mais moins optimale que l'intégration implicite, pour un système de quelques dizaines d'équations) d'introduire un nouveau comportement monocristallin en petites déformations : il suffit de d'écrire les dérivées des variables internes dans la routine LCMMON, appelée par RDIF01.

La routine LCMMON a pour but de calculer les dérivées des variables internes. On doit résoudre un système de $6+3\times n_s$ équations différentielles, du type (cf. [R5.03.11]) : $\frac{dY}{dt}=F\left(Y,t\right)$, où Y représente l'ensemble des variables internes : $Y=\begin{cases}\alpha_s\\ \gamma_s\\ p_s\\ E^{vp}\end{cases}$

Le système d'équations différentielles à résoudre est :

- $\boldsymbol{\varepsilon}^{\dot{vp}}=\sum_s \boldsymbol{\mu}_s \dot{\gamma}_s$ 6 équations
- pour chaque système de glissement (sur l'ensemble des familles de systèmes) 3 relations :
  - $\dot{\gamma}_s=\dot{p}_s\left(\tau_s(\boldsymbol{\sigma}),\alpha_s,\gamma_s,R_s(p)\right)\eta\left(\tau_s,\alpha_s\right)$ où $\eta(\tau_s,\alpha_s)=\pm 1$
  - $\dot{\alpha}_s=h\left(\tau_s,\alpha_s,\gamma_s,p_s\right)$
  - $R_s(p)$

où $\tau_s=\boldsymbol{\mu}_s:\boldsymbol{\sigma}$ où $\boldsymbol{\sigma}$ est déduite de la relation : $\boldsymbol{\sigma}=\boldsymbol{\Lambda}\left(\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}^{vp}\right)$

En pratique, la résolution s'effectue de la façon suivante :

La routine LCMMON calcule les contraintes par la relation d'élasticité (isotrope ou orthotrope)

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

CALL CALSIG(...) ce qui fournit le tenseur SIG = σ

Puis elle calcule les $6+3\times n_s$ dérivées issues des équations différentielles ci-dessus, et les stocke dans un tableau DVIN :

- Boucle sur les familles de systèmes de glissement :

```
DO IFA=1,NBFSYS
   …
```

Récupération du nombre de systèmes de glissement
```
CALL LCMMSG(NOMFAM,NBSYS,0,PGL,MS,NG,LG,0,Q)
```

Boucle sur les systèmes de glissement de la famille IFA :

```
DO IS=1,NBSYS
```

CALL LCMMSG(..) relecture du tenseur d'orientation MuS

CALCUL DE LA CISSION REDUITE
TAUS= SIG(I)*MuS(I) pour i=1,6 $\tau_s = \mu_s : \sigma$

CALL LCMMFI(=> RP) **routine de calcul de l'écrouissage isotrope** $R_s(p)$

CALL LCMMFE(=> DGAMMA, DP) **routine de calcul de l'écoulement** $\dot{\gamma}_s, \dot{p}_s$

CALL LCMMEC(=> DALPHA) **routine de calcul de l'écrouissage cinématique** $\dot{\alpha}_s$

Calcul de la déformation viscoplastique globale
```
DO ITENS=1,6
   DEVI(ITENS)=DEVI(ITENS)+MuS(ITENS)*DGAMMA
ENDDO
```
$\dot{\varepsilon}^{vp} = \sum_s \mu_s \dot{\gamma}_s$

stockage des dérivées des variables internes pour le système de glissement IS
```
DVIN(NUVI-2)=DALPHA
DVIN(NUVI-1)=DGAMMA
DVIN(NUVI  )=DP
```

```
ENDDO
```

```
ENDDO
```

stockage du tenseur dérivée de la déformation viscoplastique.
```
DO ITENS=1,6
   DVIN(ITENS)= DEVI(ITENS)
ENDDO
```

L'algorithme de Runge-Kutta gère alors l'intégration de ces équations différentielles, en contrôlant l'erreur, et en raffinant le pas de temps jusqu'à obtenir une erreur inférieure à la précision demandée (RESI_INTE_RELA) [R5.03.14].

A priori, la structure de la routine LCMMON, ne doit pas évoluer lors de l'ajout d'un nouveau comportement monocristallin ; seules les routines LCMMFI, LCMMFE et LCMMEC sont à modifier.

Elles sont construites de la façon suivante :

LCMMFI

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
C-----------------------------------------------------------------
C     POUR UN NOUVEAU TYPE D'ECROUISSAGE ISOTROPE, AJOUTER UN BLOC 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
```

Remarque : on utilise ici les numéros associés à chaque type de comportement plutôt que les noms, pour optimiser les performances.

De même la structure de la routine LCMMFC est :

```
C-----------------------------------------------------------------
C   POUR UN NOUVEAU TYPE D'ECROUISSAGE CINEMATIQUE, AJOUTER UN BLOC 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
```

Et de façon similaire, la structure de la routine LCMMFC est :

```
C-------------------------------------------------------------
C     POUR UN NOUVEAU TYPE D'ECOULEMENT, CREER UN BLOC IF
C-------------------------------------------------------------
C     IF (NECOUL.EQ.'MONO_VISC1') THEN
      IF (NUECOU.EQ.1) THEN
          DP=...
          DGAMMA= ...
      ELSEIF (NUECOU.EQ.2) THEN
      ENDIF
```

Remarque : Les routines LCMMFE, LCMMFI, LCMMFC sont également utilisées par l'intégration explicite du polycristal et par l'intégration implicite. Ceci simplifie la mise en œuvre d'un nouveau comportement cristallin.

3.4.2 Cas du polycristal :

L'intégration du polycristal s'appuie largement sur celle du monocristal : on calcule là encore les dérivées des variables internes pour chaque monocristal de chaque grain g, dans la routine LCMMOP, appelée par RDIF01.

On doit résoudre un système de $6+n_g\left(6+3\times n_s(g)\right)$ équations différentielles, du type :

- pour chaque grain défini par une orientation et une proportion f_g, une relation de localisation des contraintes, de la forme générale :

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

$\boldsymbol{\sigma}_g = L\left(\boldsymbol{\Sigma}, \boldsymbol{E}^{vp}, \varepsilon_g^{vp}, \beta_g\right)$ avec, $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}(\boldsymbol{\Lambda}_-^{-1})\boldsymbol{\Sigma}^- + \boldsymbol{\Lambda}(\Delta \boldsymbol{E} - \Delta \boldsymbol{E}^{th} - \Delta \boldsymbol{E}^{vp})$

- pour chacun des $n_s(g)$ systèmes de glissement de chaque grain g, 3 relations :
  - $\dot{\gamma}_s = \dot{p}_s\left(\tau_s(\boldsymbol{\sigma}), \alpha_s, \gamma_s, R_s(p)\right)\eta(\tau_s, \alpha_s)$ où $\eta(\tau_s, \alpha_s) = \pm 1$
  - $\dot{\alpha}_s = h\left(\tau_s, \alpha_s, \gamma_s, p_s\right)$
  - $R_s(p)$

- à l'échelle du grain, lcalcul de la déformation plastique : $\dot{\boldsymbol{\varepsilon}}_g^{vp} = \sum_s \boldsymbol{\mu}_s \dot{\gamma}_s$
- calcul de la déformation plastique macroscopique : $\dot{E}^{vp} = \sum_g f_g \dot{\varepsilon}_g^{vp}$

En pratique, la résolution s'effectue de la façon suivante :

La routine `LCMMOP` calcule les contraintes par la relation d'élasticité (isotrope ou orthotrope)
`CALL CALSIG(...)` ce qui fournit le tenseur `SIG` ($\boldsymbol{\Sigma}$)

Puis elle calcule les $6 + n_g\left(6 + 3 \times n_s(g)\right)$ dérivées issues des équations différentielles ci-dessus, et les stocke dans un tableau DVIN :

- Boucle sur les grains :

`DO IGRAIN=1,NGRAIN`
`CALL LCLOCA()` relation de localisation permettant de calculer `SIGG` $(\boldsymbol{\sigma}_g)$

- Boucle sur les familles de systèmes de glissement :

`DO IFA=1,NBFSYS`
…
Récupération du nombre de systèmes de glissement
`CALL LCMMSG(...)`

Boucle sur les systèmes de glissement de la famille `IFA` :
`DO IS=1,NBSYS`

`CALL LCMMSG(..)` relecture du tenseur d'orientation `MuS(IGRAIN, IS)`

CALCUL DE LA CISSION REDUITE
`TAUS= SIGG(I)*MuS(I) pour i=1,6`

`CALL LCMMFI( => RP)` **routine de calcul de l'écrouissage isotrope**

`CALL LCMMFE(=> DGAMMA, DP)` **routine de calcul de l'écoulement**

`CALL LCMMEC( => DALPHA )` **routine de calcul de l'écrouissage cinématique**

Calcul de la déformation viscoplastique globale
`DO ITENS=1,6`
`c DEVG(ITENS)=DEVG(ITENS)+MuS(ITENS)*DGAMMA` $(\mathrm{DEVG} = \boldsymbol{\varepsilon}_g^{vp})$
`ENDDO`

stockage des dérivées des variables internes pour le système de glissement `IS`
`DVIN(NUVI-2)=DALPHA`
`DVIN(NUVI-1)=DGAMMA`
`DVIN(NUVI )=DP`

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
        ENDDO

    ENDDO
    homogénéisation des déformations viscoplastiques
    DO I=1,6
       DEVI(I)=DEVI(I)+FV*DEVG(I)
    ENDDO
```

stockage du tenseur dérivée de la déformation viscoplastique.

```
DO ITENS=1,6
    DVIN(ITENS)= DEVI(ITENS)
ENDDO
```

La septième variable interne contient la déformation viscoplastique équivalente cumulée

```
DVIN(7)= DVINEQ
```

L'algorithme de Runge-Kutta gère alors l'intégration de ces équations différentielles, en contrôlant l'erreur, et en raffinant le pas de temps jusqu'à obtenir une erreur inférieure à la précision demandée (RESI_INTE_RELA) [R5.03.14].

La structure de la routine LCMMOP, ne doit pas évoluer lors de l'ajout d'un nouveau comportement monocristallin ; seules les routines LCMMFI, LCMMFE et LCMMEC sont à modifier (ce qui est déjà fait en principe pour l'intégration du monocristal). Une modification supplémentaire est à effectuer dans la routine LCLOCA lors de l'ajout d'une nouvelle règle de localisation. Enfin, pour certains post-traitements spécifiques au niveau du point d'intégration, il faut intervenir dans la routine LCDPEQ.

3.5 Intégration implicite du monocristal par Newton dans PLASTI

On intègre cette fois le comportement monocristallin par une méthode de Newton. Cette méthode est programmée dans PLASTI [R5.03.14]. Il faut donc fournir à cet algorithme écrire le système d'équations à résoudre sous forme purement implicite, de la façon suivante : $\boldsymbol{R}(\boldsymbol{Y})=0$

$$R(Y)=\left\{\begin{array}{l}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Sigma}-(\boldsymbol{\Lambda}_{-}^{-1})\boldsymbol{\Sigma}^{-}-(\Delta\boldsymbol{E}-\Delta\boldsymbol{E}^{\mathrm{th}}-\Delta\boldsymbol{E}^{\mathrm{vp}})\\ \Delta\boldsymbol{E}^{\mathrm{vp}}-\sum_s \mu_s\Delta\gamma_s\\ n_s\left\{\begin{array}{l}\Delta\alpha_s-h(\tau_s^+,\alpha_s^+,\gamma_s^+,p_s^+)\\ \Delta\gamma_s-g(\tau_s^+,\alpha_s^+,\gamma_s^+,p_s^+)\\ \Delta p_s-f(\tau_s^+,\alpha_s^+,\gamma_s^+,p_s^+)\end{array}\right.\end{array}\right. \quad =\mathbf{0}$$

C'est un système de $6+6+3n_s$ équations non linéaires (de même taille que celui qui est intégré par la méthode de Runge-Kutta en explicite). Mais ce n'est pas le système effectivement résolu : afin d'optimiser les performances, on résout en fait un système d'équations réduit, de taille $6+n_s$, construit de la façon suivante [R5.03.11] :

- Dans l'expression des 6 composantes du tenseur des contraintes, $\Delta\boldsymbol{E}^{vp}$ peut être exprimée en fonction de $\sum_s \mu_s\Delta\gamma_s$ donc 6 équations peuvent être éliminées du système global à résoudre.
- Comme $\Delta p_s=|\Delta\gamma_s|$, l'équation en Δp_s peut être éliminée
- pour tous les comportements cristallins considérés actuellement, soit $\Delta\alpha_s$ s'exprime directement en fonction de $\Delta\gamma_s$, dans le cas de l'écrouissage cinématique, soit $\Delta\gamma_s$ s'exprime en fonction de $\Delta\alpha_s$ qui représente alors la densité de dislocation (à un facteur près) pour les comportements de type MONO_DD_*. Il y a donc une seule inconnue par système de glissement : $\Delta\beta_s$, qui représente soit $\Delta\gamma_s$, soit $\Delta\alpha_s$

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

On obtient donc le système suivant :

- **Petites déformations:**

$$\boldsymbol{R_1}(\boldsymbol{\sigma},\Delta\beta)=\boldsymbol{\Lambda}^{-1}.\Delta\boldsymbol{\sigma}-\Delta\boldsymbol{\varepsilon}+\Delta\boldsymbol{\varepsilon}^{th}+\sum_s \Delta\gamma_s\boldsymbol{\mu_s}=0$$
$$\boldsymbol{R_2}(\boldsymbol{\sigma},\Delta\beta)=\Delta\beta_s-k_s(\tau_s(\boldsymbol{\sigma}),\Delta\beta)=0$$

où l'inconnue est : $\boldsymbol{Y}=\begin{bmatrix}\boldsymbol{\sigma}\\ \Delta\beta\end{bmatrix}$

avec $\tau_s(\boldsymbol{\sigma})=\boldsymbol{\sigma}:\boldsymbol{\mu_s}$

- **Grandes déformations**

$$\boldsymbol{R_1}(\boldsymbol{S},\Delta\beta)=\boldsymbol{\Lambda}^{-1}.\boldsymbol{S}-\frac{1}{2}\left(\boldsymbol{F}^{eT}\boldsymbol{F}^{e}-\boldsymbol{I_d}\right)=0$$
$$\boldsymbol{R_2}(\boldsymbol{S},\Delta\beta)=\Delta\beta_s-k_s(\tau_s(\boldsymbol{S}),\Delta\beta)=0$$

où l'inconnue est : $\boldsymbol{Y}=\begin{bmatrix}\boldsymbol{S}\\ \Delta\beta\end{bmatrix}$

avec $\tau_s(\boldsymbol{S})=\left[\left(2\boldsymbol{\Lambda}^{-1}\boldsymbol{S}+\boldsymbol{I_d}\right)\boldsymbol{S}\right]:\boldsymbol{m_s}\otimes\boldsymbol{n_s}$ et $\boldsymbol{F}^{e}_{n+1}=\Delta\boldsymbol{F}\,\boldsymbol{F}^{e}_{n}\left(\Delta\boldsymbol{F}^{p}(\Delta\gamma_s)\right)^{-1}$

Et, suivant le comportement considéré,

- $\Delta\gamma_s=\Delta p_s(\tau_s,\Delta\beta_s)\xi_s$ et $\xi_s=\frac{\tau_s}{|\tau_s|}$ *ou* $\frac{\tau_s-f(\alpha)}{|\tau_s-f(\alpha)|}$ correspond au signe de l'écoulement

- $\Delta\beta_s$ représente soit l'incrément de glissement plastique $\Delta\gamma_s$, pour les lois `MONO_VISC*` , soit la variation de densité de dislocations $\Delta\omega_s$ pour les lois `MONO_DD_*`

Remarque : l'extraction des inconnues du système à partir des variables internes (qui restent au nombre de 3 par système de glissement) se fait dans la routine `LCAFYD`. Inversement, après la résolution par NEWTON, le calcul des 3 variables internes par système de glissement en fonction de la variable principale utilisée dans la résolution se fait dans la routine `LCPLNF`.

La forme générale de l'algorithme résolu par Newton est

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

Il faut donc définir les valeurs initiales ΔY_0 (0 par défaut, dans la routine `LCMMIN)` , et calculer le résidu Y_k , ainsi que la matrice jacobienne du système : $\frac{dR}{dY_k}$

3.5.1 Architecture générale de PLASTI :

L'algorithme de Newton utilisé dans `PLASTI` est décrit en [R5.03.14].

Lecture des coefficients matériau et stockage dans les objets `NBCOMM, MATERF/MATERD, CPMONO`

`CALL LCMATE => LCMMAT, identique à RUNGE_KUTTA`

Prédiction élastique
`CALL LCELAS`
Calcul du `SEUIL`
`CALL LCCNVX => routine LCMMVX évaluation du seuil pour MONOCRISTAL.`

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Calcul de la solution élasto-visco-plastique par la méthode de Newton :
```
IF ( SEUIL .GE. 0.D0 ) THEN
   CALL LCPLAS / CALL LCPLNL
ENDIF
```

Calcul de l'opérateur tangent :
```
IF ( OPT .EQ. 'RIGI_MECA_TANG' .OR. OPT .EQ. 'FULL_MECA' ) THEN
   CALL LCJPLC => CALL LCMMJP
ENDIF
```

Remarque : L'opérateur tangent est calculé automatiquement en fonction de la matrice jacobienne du système d'équations local [R5.03.11]. Ceci est réalisé en petites et en grandes déformations dans la routine LCMMJP. Il n'y a donc a priori rien à modifier pour ce calcul lors de l'ajout d'un nouveau comportement cristallin.

La routine LCCNVX permet de détecter si le seuil est franchi pour au moins un système de glissement. Elle appelle dans le cas du monocristal la routine LCMMVX . Sa structure est la suivante :
```
SEUIL=0.D0
DO IFA=1,NBFSYS
   DO IS=1,NBSYS
      CALL LCMMFI
C     ECOULEMENT VISCOPLASTIQUE
      CALL LCMMFE => DP  calculé à partir de la prédiction élastique
      IF (DP.GT.0.D0) SEUIL=1.D0
   ENDDO
ENDDO
```

On utilise donc les mêmes routines LCMMFE et LCMMFI que pour l'intégration explicite et implicite.

La routine LCPLNL réalise **la boucle de Newton**. Sa structure (générique à l'ensemble des lois de comportement sous PLASTI) est la suivante :

LCPLNL
- LCAFYD (extraction des variables internes utiles au système réduit)
- LCINIT => LCMMIN : initialisation de ΔY_0 , 0 par défaut
- LCRESI => LCMMRE : calcul du résidu
  - Soit LCJACB => LCMMJA : calcul de la matrice jacobienne
  - Soit (si ALGO_INTE=NEWTON_PERT) LCJACP calcul de la matrice jacobienne par perturbation, qui appelle LCRESI
- MGAUSS résolution
- LCRELI recherche linéaire (si ALGO_INTE=NEWTON_RELI), qui appelle LCRESI...
- LCCONV => LCMMCV critère de convergence
- LCPLNF => LCDPEC calcul des toutes les variables internes.

3.5.2 Calcul du résidu

La routine LCMMRE calcule le résidu. Sa structure est la suivante :
```
DO IFA=1,NBFSYS
   DO IS=1,NBSYS
      CALTAU calcul de τ_s (petites ou grandes déformations)
      LCMMLC calcule des quantités relatives au système de glissement :
         CALL LCMMFI( => RP)        routine de calcul de l'écrouissage isotrope
         CALL LCMMFE( => DGAMMA, DP)   routine de calcul de l'écoulement
         CALL LCMMEC( => DALPHA ) routine de calcul de l'écrouissage cinématique
      calcul de k_s et stockage dans R_2(σ, Δβ) = Δβ_s − k_s(τ_s(σ), Δβ)
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

En petites déformations : déformation viscoplastique globale
```
DO ITENS=1,6
   DEVI(ITENS)=DEVI(ITENS)+MuS(ITENS)*DGAMMA
ENDDO
```
En grandes déformations, calcul des termes nécessaires à $\Delta \boldsymbol{F}^{p}(\Delta \gamma_s)$
```
   ENDDO
ENDDO
```

- en petites déformations, calcul de $\boldsymbol{R}_1(\boldsymbol{\sigma},\Delta\beta)=\boldsymbol{\Lambda}^{-1}.\Delta\boldsymbol{\sigma}-\Delta\boldsymbol{\varepsilon}+\Delta\boldsymbol{\varepsilon}^{th}+\sum_s \Delta\gamma_s \boldsymbol{\mu}_s=0$
- en grandes déformations

  CALCFE calcul de $\boldsymbol{F}^{e}_{n+1}=\Delta\boldsymbol{F}\,\boldsymbol{F}^{e}_{n}\left(\Delta\boldsymbol{F}^{p}(\Delta\gamma_s)\right)^{-1}$

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

  et $\boldsymbol{R}_1(\boldsymbol{S},\Delta\beta)=\boldsymbol{\Lambda}^{-1}.\boldsymbol{S}-\frac{1}{2}\left(\boldsymbol{F}^{eT}\boldsymbol{F}^{e}-\boldsymbol{I}_d\right)$

On constate donc que l'on utilise là encore les même routines que pour l'intégration explicite : LCMMFI, LCMMFE, LCMMEC sont a priori les seules routines à modifier lors de l'ajout d'un comportement, que ce soit en petites ou en grandes déformations, pour le calcul du résidu, lors de l'intégration implicite.

3.5.3 Calcul de la matrice jacobienne

La routine LCMMJA calcule la matrice jacobienne (sauf si ALGO_INTE= 'NEWTON_PERT').

Sa structure est la suivante :
```
DO IFA=1,NBFSYS
   DO IS=1,NBSYS
      LCMMJB : calcul des termes dérivés
         LCMMJ2 : calcul des termes dérivés pour MONO_DD_KR
         LCMMJD : calcul des termes dérivés pour MONO_DD_CFC, MONO_DD_CC (plus _IRRA)
         LCMMJ1 : calcul des termes dérivés pour MONO_VISC1, MONO_VISC2
   ENDDO
ENDDO
```

Les dérivées de ces équations pour le calcul de la matrice jacobienne peuvent être écrites de façon générale :

HPP		GDEF	
$J_{11}=\frac{\partial \boldsymbol{R}_1(\boldsymbol{\sigma},\Delta\beta)_i}{\partial \boldsymbol{\sigma}_j}$	$J_{12}=\frac{\partial \boldsymbol{R}_1(\boldsymbol{\sigma},\Delta\beta)_i}{\partial \beta_s}$	$J_{11}=\frac{\partial \boldsymbol{R}_1(\boldsymbol{S},\Delta\beta)_i}{\partial \boldsymbol{S}_j}$	$J_{12}=\frac{\partial \boldsymbol{R}_1(\boldsymbol{S},\Delta\beta)_i}{\partial \beta_s}$
$J_{21}=\frac{\partial \boldsymbol{R}_2(\boldsymbol{\sigma},\Delta\beta)_s}{\partial \boldsymbol{\sigma}_i}$	$J_{22}=\frac{\partial \boldsymbol{R}_2(\boldsymbol{\sigma},\Delta\beta)_s}{\partial \beta_r}$	$J_{21}=\frac{\partial \boldsymbol{R}_2(\boldsymbol{S},\Delta\beta)_s}{\partial \boldsymbol{S}_i}$	$J_{22}=\frac{\partial \boldsymbol{R}_2(\boldsymbol{S},\Delta\beta)_s}{\partial \beta_r}$

Tableau 3.5.3-1

Les termes intervenant dans chaque sous-matrice ont en commun des termes spécifiques à chaque comportement, qui sont calculés dans les routines LCMMJ* [R5,03,11 annexe 5] :

- $\frac{\partial \Delta\gamma_s}{\partial \tau_s}=\frac{\partial \Delta p_s}{\partial \tau_s}\xi_s$,

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

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

Dans le cas d'un nouveau comportement, il faut donc soit ajouter le calcul de ces termes dérivés dans une routine LCMMJ* existante, soit en ajouter une nouvelle.

Remarque : pour un premier test, on peut se passer du calcul de la matrice jacobienne, en utilisant la construction automatique de la matrice jacobienne (par perturbation, ALGO_INTE='NEWTON_PERT'). De ce fait, il est très rapide d'introduire un nouveau comportement cristallin dans l'environnement PLASTI: il suffit de calculer le résidu dans la routine LCMMRE, appelée par LCRESI. Par contre, le temps calcul ne sera optimisé qu'avec une matrice jacobienne programmée.

3.5.4 Critère de convergence et post-traitements

Le critère de convergence est générique a priori, et ne dépend pas du comportement, mais peut être éventuellement modifié :

- LCCONV => LCMMCV critère de convergence

La dernière routine à modifier (éventuellement, si on veut calculer et ajouter aux variables internes en sortie des valeurs utiles au post-traitement) est :

- LCPLNF => LCDPEC qui recalcule toutes les variables internes à partir de la solution du système réduit. Elle fait appel encore une fois à la routine de comportement LCMMLC.

4 O bjets internes pour le stockage des informations

L a lecture des coefficients matériau est effectuée une seule fois par point d'intégration dans les routines LCMMAT / LCMMAP . Les quantités lues (ou calculées, dans la cas des tenseurs d'orientation) doivent être transmises aux routines d'intégration. Pour cela on utilise des tableaux spécifiques : nbcomm, coeft/materf,cpmono, toutms...)

4.1 Dans le cas du monocristal :

Stockage des informations relatives aux monocristal dans des tableaux spécifiques (nbcomm, coeft/materf, materd,cpmono,).

```
MATERF :  coefficients matériau a t+dt
          mater(*,1) = caractéristiques   élastiques
          mater(*,2) = caractéristiques   plastiques

MATERD :  coefficients matériau à t

NBCOMM(*,3) : POSITION DES COEF POUR CHAQUE  SYSTEME
tableau d'entiers
             Colonne 1      Colonne 2      Colonne3
             _____________________________________
 Ligne 1     1              Nb var.int.   Nb monocristaux
 pour chaque famille de systèmes de glissement :
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
famille 1  ind coef       ind coef      ind coef
           ecoulement     ecr iso       ecr cin
  .....
ind signifie l'indice dans MATERF/MATERD(*,2)
dernière ligne :          nb de familles

CPMONO : tableau de chaînes de caractères, copie de l'objet sd_compor.CPRK
         par famille de systèmes de glissement
            Nom de la famille
            Nom du matériau
            Nom de la loi d'écoulement
            Nom de la loi d'écrouissage isotrope
            Nom de la loi d'écrouissage cinématique
```

TOUTMS tableau réel de dimension (Nombre de familles, nombre de systèmes, 6) contenant tous les tenseurs mus=sym(ms*ns) en hpp, et tous les vecteurs ms et ns en grandes déformations.

4.2 Dans le cas du polycristal :

STRUCTURE DES OBJETS CRÉES

```
COEFT(*) : tableau de réels
            Nombre de monocristaux
            indice (dans COEFT) du début du premier monocristal
            indice début deuxième monocristal
            ….
            indice début dernier monocristal
            indice des paramètres de localisation
            Fv et 3 angles par phase
pour chaque monocristal différent
      par famille de système de glissement
         nb coef écoulement
            numéo de la loi d'écoulement + coef,
         nb coef écrou isot + num_loi + coef,
         nb coef ecou cine + num_loi + coef
             puis les paramètres de la loi de localisation

CPMONO(*) : tableau de chaînes de caractères
      nom de la méthode de localisation
      puis, pour chaque matériau différent
      nom du monocristal, nombre de familles SG, et,
         par famille de systèmes de glissement
            Nom de la famille
            Nom du matériau
            Nom de la loi d'éécoulement
            Nom de la loi d'écrouissage isotrope
            Nom de la loi d'écrouissage cinématique
            Nom de la loi d'élasticité (ELAS ou ELAS_ORTH)

NBCOMM(*,3) : tableau d'entiers
             Colonne 1      Colonne 2      Colonne3
            _____________________________________________
 Ligne 1     Nb phases      Nb var.int.   Nb monocristaux
                                          différents
   pour chaque phase g  Num ligne g    Ind CPMONO    ind frac vol
   ..................
   ..................
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
   pour chaque phase
   pour la localisation  indice coef   nb param      0
   phase g              nb fam g       0            NVIg
     ... et pour chaque famille de systèmes de glissement :
  famille 1  ind coef       ind coef      ind coef
             ecoulement     ecr iso       ecr cin
    .....
         (ind signifie l'indice dans COEFT(*)
```

TOUTMS : TOUS LES TENSEURS D'ORIENTATION POUR TOUS LES SYSTEMES DE GLISSEMENT DE TOUS LES MONOCRISTAUX.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)