

**Manuel de Descriptif Informatique**  
**Fascicule D5.04 : -**  
**Document : D5.04.01**

## Introduire une nouvelle loi de comportement

---

### Résumé :

Le présent document a pour but de fournir aux développeurs les principaux éléments nécessaires à l'implantation (ou la modification) d'une loi de comportement dans *Code\_Aster*. Il décrit les modifications à effectuer sur le catalogue de commandes, ainsi que la nouvelle routine FORTRAN à créer pour intégrer le comportement soit de façon explicite (méthode de RUNGE-KUTTA), soit de façon implicite, dans l'environnement PLASTI (méthode de NEWTON) ou de façon optimisée.

## Table des matières

1	Etapes essentielles pour introduire une nouvelle loi de comportement .....	3
1.1	Ecriture de la doc R .....	3
1.2	Modification du catalogue de <code>DEFI_MATERIAU</code> .....	3
1.3	Modification des catalogues de <code>STAT_NON_LINE</code> et <code>DYNA_NON_LINE</code> .....	4
1.4	Ecriture d'une routine relative à un point d'intégration d'un élément .....	4
1.5	Branchement de cette routine dans la routine <code>NMCOMP</code> (comportement inélastique) ou <code>NMCPEL</code> (comportement élastique) .....	5
2	Modifications des catalogues de commandes .....	5
2.1	<code>DEFI_MATERIAU</code> .....	5
2.2	<code>STAT_NON_LINE</code> , <code>DYNA_NON_LINE</code> , <code>DYNA_TRAN_EXPLI</code> .....	6
3	Modifications des routines .....	7
3.1	Dans quelle(s) routine(s) intervenir ? .....	7
3.2	Programmation d'une loi en passant par la routine d'aiguillage <code>NMCOMP</code> ou <code>NMCPEL</code> .....	7
3.2.1	Principe .....	7
3.2.2	Exemple d'une routine réalisation l'intégration d'une loi de comportement : <code>NMCINE</code> .....	10
3.3	Programmation d'une loi dans l'environnement <code>PLASTI</code> .....	13
3.3.1	Introduction .....	13
3.3.2	Algorithme de résolution du problème quasi-statique .....	14
3.3.3	Environnement <code>Plasti</code> .....	14
3.3.4	Formalisation des équations à résoudre .....	15

# 1 Etapes essentielles pour introduire une nouvelle loi de comportement

## 1.1 Ecriture de la doc R

En premier lieu, il faut écrire la documentation de référence de la loi de comportement reliant en un point donné les contraintes aux déformations. On se restreint ici aux milieux continus 2D et 3D. La méthode reste valable pour des modèles à comportement non linéaire local en contraintes planes, tels que les coques, les plaques et les tuyaux, ou les modélisations à comportement non linéaire mono-dimensionnel, comme les poutres multi-fibres, les barres, les grilles, via la méthode DE BORST [R5.03.03]. Cette méthode est applicable de façon générale en petites déformations, quel que soit le comportement, à condition que l'opérateur tangent du nouveau comportement soit accessible et efficace.

Pour résoudre le problème non linéaire posé sur la structure, le document [R5.03.01] décrit l'algorithme utilisé dans *Aster*.

A chaque itération  $n$  de la méthode Newton [R5.03.01 § 2.2.2.2] on doit calculer les forces nodales  $\mathbf{R}(\mathbf{u}_i^n) = \mathbf{Q}^T \boldsymbol{\sigma}_i^n$  (options `RAPH_MECA` et `FULL_MECA`) les contraintes  $\boldsymbol{\sigma}_i^n$  étant calculées à partir des déplacements  $\mathbf{u}_i^n$  par l'intermédiaire de la relation de comportement. On doit construire aussi l'opérateur tangent pour calculer  $\mathbf{K}_i^n$  (option `FULL_MECA`).

A la première itération, on calcule  $\mathbf{K}_{i-1}$ .

Le calcul de  $\mathbf{K}_{i-1}$  (option `RIGI_MECA_TANG`), qui est nécessaire à la phase d'initialisation [R5.03.01 § 2.2.2.1] correspond au calcul de l'opérateur tangent déduit du problème en vitesse ci-dessous.

Cet opérateur n'est pas identique à celui qui est utilisé pour calculer  $\mathbf{K}_i^n$  par l'option `FULL_MECA`, au cours des itérations de Newton. En effet, ce dernier opérateur est tangent au problème discrétisé de façon implicite.

## 1.2 Modification du catalogue de `DEFI_MATERIAU`

Le but de `DEFI_MATERIAU` est d'introduire des paramètres de comportement. Ces paramètres peuvent être communs à plusieurs relations de comportement.

Il faut éventuellement ajouter dans le catalogue de `DEFI_MATERIAU` un mot clé facteur correspondant au type de comportement que l'on veut modéliser et sous ce mot clé facteur, les mots clés représentant les paramètres de ce type de comportement.

### Remarque importante :

*D'un point de vue informatique, les mots clés facteurs doivent être des K10 (chaîne de caractères limitée à 10 caractères), et les mots clés sous jacents sont limités à 8 caractères. En pratique, cela signifie que si un mot-clé est plus long, seuls les 8 premiers caractères seront utilisés. Il a donc un risque de collision avec d'autres mots-clés ayant les 8 premiers caractères en commun.*

### 1.3 Modification des catalogues de STAT\_NON\_LINE et DYNA\_NON\_LINE

Dans les catalogues de ces commandes, on donne sous les mots-clés facteurs COMP\_INCR ou COMP\_ELAS, le nom de la relation de comportement après le mot clé RELATION.

On donne aussi le nombre de variables internes associé à cette relation après le nom de cette relation.

Le nom de cette relation de comportement peut être différent du nom du type de comportement donné dans DEFI\_MATERIAU.

### 1.4 Ecriture d'une routine relative à un point d'intégration d'un élément

Le point d'intégration est un point de Gauss dans le cas de milieux continus, ou un point d'intégration dans l'épaisseur, pour des coques, par exemple.

Les arguments d'entrée sont :

- l'incrément de déformation totale,
- le tenseur de contraintes à l'instant du calcul précédent,
- les variables internes à l'instant du calcul précédent ; par exemple  $\varepsilon^P, p, X_i$ ,
- l'option de calcul : 3 options doivent être calculées : 'RIGI\_MECA\_TANG', 'RAPH\_MECA' et 'FULL\_MECA'.

Les arguments de sortie sont selon l'option de calcul :

- le tenseur des contraintes réactualisé (RAPH\_MECA et FULL\_MECA),
- les variables internes réactualisées (RAPH\_MECA et FULL\_MECA),
- la matrice de comportement tangente cohérente ou en vitesse (FULL\_MECA et RIGI\_MECA\_TANG).

#### Remarques importantes :

*Les tenseurs déformation, contraintes à l'instant précédent, et incrément de déformation, donnés en arguments d'entrée, sont tels que les composantes hors diagonale (cisaillement pour les contraintes, et distorsion pour les déformations, sont multipliés par  $\sqrt{2}$  avant appel à la routine d'intégration du comportement. Par voie de conséquence, les composantes de cisaillement du tenseur de contraintes en sortie doivent également être multipliées par le même coefficient  $\sqrt{2}$ .*

*On décrit ici l'intégration d'un nouveau comportement sous l'hypothèse des petites déformations. Les hypothèses disponibles dans Code\_Aster sur les déformations sont :*

- *PETIT : dans ce cas des tenseurs déformations sont calculés linéairement par rapport aux déplacements, sur la géométrie initiale (Hypothèse des Petites Perturbations : HPP) ;*
- *PETIT\_REAC : les déformations sont calculées linéairement à partir des déplacements sur la géométrie réactualisée. Rien ne change dans l'intégration du comportement ;*
- *GREEN : dans ce cas les déformations fournies sont les déformations de GREEN-LAGRANGE. Sous l'hypothèse de petites déformations (mais de grands déplacements), le comportement s'exprime de façon similaire au comportement HPP, mais relie cette fois les déformations de GREEN-LAGRANGE aux contraintes de PIOLA-KICHHOFF de 2<sup>ème</sup> espèce. La transformation des contraintes PK2 en contraintes de Cauchy est géré par les routines appelantes de NMCOMP. [R5.03.22].*
- *SIMO\_MIEHE : dans ce cas les arguments d'entrée correspondent au gradient de la transformation  $F$  à l'instant précédent et de gradient de la transformation entre la configuration à l'instant précédent et la configuration actuelle  $\Delta F$ . Mais dans ce cas il est nécessaire de formuler le modèle de comportement en grandes transformations, et on ne peut plus utiliser la formulation HPP comme précédemment [R5.03.21].*

## 1.5 Branchement de cette routine dans la routine NMCOMP (comportement inélastique) ou NMCPEL (comportement élastique)

L'aiguillage se fait selon le nom de la relation qui a été donné sous le mot clé RELATION de COMP\_INCR ou COMP\_ELAS.

## 2 Modifications des catalogues de commandes

### 2.1 DEFI\_MATERIAU

On introduit dans le catalogue de la commande DEFI\_MATERIAU un mot clé facteur sous lequel on va pouvoir donner les paramètres nécessaires à la description du comportement du matériau.

Ce mot clé facteur est un K16, dont seulement les 10 premiers caractères sont significatifs.

Exemples :

- pour décrire un comportement élastique, un utilisateur va employer le mot clé facteur ELAS dans DEFI\_MATERIAU : mater = DEFI\_MATERIAU (ELAS = ...),
- pour décrire un comportement élastoplastique avec écrouissage linéaire, un utilisateur va employer les mots clés facteurs ECRO\_LINE et ELAS : mater = DEFI\_MATERIAU (ELAS = ..., ECRO\_LINE = ...).

Sous les mots clés facteurs définissant le comportement du matériau, on donne les mots clés qui vont correspondre aux noms des paramètres de la loi et après lesquels on donne les valeurs de ces paramètres.

Ces mots clés sont des K8.

Les valeurs des paramètres sont soit des nombres réels, soit des fonctions (donc des K8).

Exemples :

- pour un matériau élastique, on doit donner le module d'Young E et le coefficient de Poisson nu.  
On a ainsi :  
mater = DEFI\_MATERIAU (ELAS= \_F ( E = yg, [R]  
NU = nu, [R]  
)  
)
- pour un matériau élastoplastique à écrouissage linéaire, on doit donner les caractéristiques élastiques et la courbe d'écrouissage linéaire qui est définie par la limite d'élasticité SY et la pente de la courbe de traction D\_SIGM\_EPSI, i.e.  $\frac{\Delta\sigma^*}{\Delta\epsilon^*}$ .

On a ainsi :

```

mater = DEFI_MATERIAU ( ELAS= _F ( E = yg,      [R]
                                Nu = nu,      [R]
                                )
                        ECRO_LINE = _F( SY = sy,  [R]
                        D_SIGM_EPSI = dsde,    [R]
                        ),
  )

```

## 2.2 STAT\_NON\_LINE, DYNA\_NON\_LINE, DYNA\_TRAN\_EXPLI

On se place dans le cas des lois non-linéaires. Il faut modifier les catalogues des commandes STAT\_NON\_LINE et DYNA\_NON\_LINE en donnant le nom de la relation après le mot clé RELATION sous les mots clés facteurs COMP\_INCR ou COMP\_ELAS.

Exemples :

- dans le cas d'une relation de comportement d'élasto-plasticité de von Mises à écrouissage isotrope linéaire, on a :

```
statnl = STAT_NON_LINE (  
    MODELE = mod,  
    CHAM_MATER = chmat,  
    COMP_INCR = _F(RELATION = 'VMIS_ISOT_LINE'),  
    ...)
```

- dans le cas d'une relation de comportement d'élasto-plasticité de Von Mises à écrouissage cinématique linéaire, on a :

```
statnl = STAT_NON_LINE (  
    MODELE = mod,  
    CHAM_MATER = chmat,  
    COMP_INCR = _F(RELATION = 'VMIS_CINE_LINE'),  
    ...)
```

Il faut remarquer que ces deux relations utilisent les mêmes paramètres de DEFI\_MATERIAU mais les comportements sont différents et les nombres des variables internes sont différents :

2 pour VMIS\_ISOT\_LINE :  $p, \chi$   
7 pour VMIS\_CINE\_LINE :  $X, \chi$

- $p$  désigne la formation plastique cumulée,
- $X$  désigne le tenseur de rappel (il est symétrique, il a donc 6 composantes),
- $\chi$  désigne un indicateur de plasticité en un point donné :
  - si  $X = 1$  le point est 'plastique',
  - si  $\chi = 0$  il ne l'est pas.

## 3 Modifications des routines

### 3.1 Dans quelle(s) routine(s) intervenir ?

Pour réaliser le calcul du comportement en un point d'intégration [§1.4], c'est-à-dire le calcul des contraintes et des variables internes, et le calcul du comportement tangent, trois solutions sont envisageable dans *Code\_Aster* :

- soit utiliser l'architecture de l'environnement *PLASTI*. Il s'agit d'un ensemble de routines (routines d'aiguillage et routines utilitaires) permettant d'introduire un nouveau modèle "à peu de frais", c'est-à-dire en définissant quelques routines spécifiques. Par contre, *PLASTI* ne permet pas d'obtenir des modèles optimisés en temps calcul. Cet environnement est décrit en [§3.3]. Ce cadre général est utilisé en particulier pour l'intégration des modèles de comportement des monocristaux [R5.03.11] ;
- soit, ce qui n'est pas conseillé pour obtenir une bonne convergence, mais peut être plus rapide dans une phase de test, utiliser l'architecture de l'intégration explicite par la méthode de RUNGE-KUTTA [R5.03.14]. En effet la simplicité vient du fait que seules les équations différentielles décrivant l'évolution des variables internes sont à programmer, puisque dans ce cas il n'y a pas d'opérateur tangent ;
- soit créer une routine complète d'intégration du comportement, ce qui par le biais de l'exemple des autres routines existantes, permet souvent d'obtenir des modèles performants (par exemple, en réduisant le système à résoudre à une seule équation scalaire, non linéaire). Ce procédé est décrit au [§3.2.2].

En version 8, il sera également possible de définir un nouveau comportement dans le formalisme de Zmat (module de comportement du code Zebulon) via le couplage Aster-Zmat. Le but de cette fonctionnalité est de prototyper de nouveaux modèles, mais pas de l'utiliser sur des calculs de structures de taille importante, car le temps calcul est augmenté de façon significative. L'interface sera accessible uniquement dans le cadre du partenariat Ecole des Mines de Paris – EDF R&D.

### 3.2 Programmation d'une loi en passant par la routine d'aiguillage NMCOMP OU NMCPXL

#### 3.2.1 Principe

On se place dans le cas de la routine *NMCOMP* qui fait l'intégration des lois de comportement incrémentales (donc relatives à *COMP\_INCR*).

La routine *NMCOMP* est appelée au niveau du calcul des éléments, soit des TE.

En fait *NMCOMP* n'est pas appelée directement par les TE mais par des routines elles-mêmes appelées par les TE.

Ces routines sont :

- *NMPL2D* et *NMPL3D* pour les éléments massifs 2D et 3D en petites déformations,
- *NMGP2D* et *NMGP3D* pour les éléments massifs 2D et 3D en grandes déformations (*SIMO\_MIEHE*),
- *DKQNLI* et *DKTNLI* pour les éléments *DKQ* et *DKT*,
- *VDXNLR* pour les coques épaisses 3D,
- *TE0329* pour les coques 1D,
- *TUFULL* pour les éléments tuyaux.
- *NMCO1D* pour les comportements 1D des éléments *BARRE*, Poutres multi-Fibres, Grilles.

Les calculs effectués au niveau de *NMCOMP* sont relatifs à un point d'intégration donné d'un élément donné.

Ces calculs consistent à déterminer à l'instant actuel le tenseur des contraintes, les variables internes et la matrice de comportement tangente.

Le but du jeu est d'écrire une routine faisant ces calculs, cette routine étant appelée par NMCOMP. Le débranchement vers cette routine dans NMCOMP se fait à partir d'un test consistant à comparer la variable COMPOR(1) avec le nom de la relation de comportement donnée sous le mot clé RELATION de COMP\_INCR et traitée dans la routine.

Si on a défini une relation de nom 'TOTO', i.e. on a écrit :

```
statnl = STAT_NON_LINE (  
    MODELE = mod,  
    CHAM_MATER = chmat,  
    COMP_INCR = _F(RELATION = 'TOTO'),  
    ...)
```

On va écrire une routine NMTOTO qui va être appelée par NMCOMP de la manière suivante :

```
IF (COMPOR(1) (1 :4).EQ'TOTO') then CALL NMTOTO (...,...)  
  
ELSE  
  
...  
  
ENDIF
```

Prenons NMTOTO comme une routine générique pour réaliser l'intégration d'une loi de comportement.

Les arguments en sortie de NMTOTO seront :

Nom	Type	Signification
SIGP(6)	R	contraintes à l'instant actuel
VIP(NBVARI)	R	variables internes à l'instant actuel
DSIDEP(6,6)	R	matrice de comportement tangente. C'est une matrice carrée dimensionnée 'en dur' 6 x 6 pour les milieux continus 2D et 3D

Les arguments en entrée de NMTOTO seront :

Nom	Type	Signification
NDIM	I	Dimension de l'espace (2 ou 3, adresse du matériau codé). <ul style="list-style-type: none"><li>• TYPMOD(1) est le type de modélisation : 3D, D_PLAN, AXIS ou C_PLAN,</li><li>• TYPMOD(2) est égal à 'INCO' pour les éléments incompressibles.</li></ul>
COMPOR(3)	K16	Tableau de 3 K16 relatifs à relation de comportement. <ul style="list-style-type: none"><li>• COMPOR(1) est le nom de la relation de comportement,</li><li>• COMPOR(2) est le nombre de variables internes par point d'intégration,</li><li>• COMPOR(3) est un K16 désignant une hypothèse sur les déformations.</li></ul>

crit(5)	R	Tableau de critères de convergence locaux.
		<ul style="list-style-type: none"> <li>crit(1) : nombre d'itérations maximum à la convergence,</li> <li>crit(2) : type de matrice tangente à l'instant <math>t+dt</math>,</li> <li>si crit(2) = 0, on a une formulation en vitesse et la matrice est symétrique,</li> <li>si crit(2) = 1, on a une formulation incrémentale et la matrice peut être non-symétrique,</li> <li>crit(3) est la valeur de tolérance de convergence,</li> <li>crit(4) est le nombre d'incrémentes pour le redécoupage local du pas de temps,</li> <li>si crit(4) = - 1, 0 ou 1, il n'y a pas de redécoupage,</li> <li>crit(5) est le type d'intégration locale pour la loi de comportement,</li> <li>si crit(5)=0, l'intégration est Euler-implicite,</li> <li>si crit(5) = 1, on fait une intégration de RUNGE_KUTTA.</li> </ul>
instam	I	Instant du calcul précédent
instap	I	Instant du calcul
TM	R	Température à l'instant du calcul précédent
TP	R	Température à l'instant du calcul
TREF	R	Température de référence
EPSM(6)	R	Déformations à l'instant du calcul précédent (voir remarque a).
DEPS(6)	R	Incrément de déformation, i.e., il s'agit de $B.\Delta U$ en HPP (voir remarque a).
SIGM(6)	R	Contraintes à l'instant du calcul précédent
VIM(NBVARI)	R	variables internes à l'instant du calcul précédent ; NBVARI est en entier entré 'en dur' dans la routine, propre à la relation de comportement et non une variable
option	K16	Option de calcul demandée. On a le choix entre : <ul style="list-style-type: none"> <li>RIGI_MECA_TANG : cette option sert lors de la prédiction, les variables internes et les contraintes ne sont pas calculées, (voir remarque b),</li> <li>FULL_MECA : on réactualise la matrice tangente à chaque itération et on met à jour les contraintes et les variables internes,</li> <li>RAPH_MECA : on ne réactualise pas la matrice tangente ; on met à jour les contraintes et les variables internes.</li> </ul>

Titre : Introduire une nouvelle loi de comportement  
Auteur(s) : J.M. PROIX, G. BERTRAND

Date : 22/06/05  
Clé : D5.04.01-B Page : 10/16

## Remarques :

- a. Comme précisé précédemment, les tenseurs déformation, contraintes à l'instant précédent, et incrément de déformation, donnés en arguments d'entrée, sont tels que les composantes hors diagonale (cisaillement pour les contraintes, et distorsion pour les déformations, sont multipliés par  $\sqrt{2}$  avant appel à la routine NMCOMP. Ceci a été établi pour faciliter les calculs de normes intervenant dans un certain nombre de comportements non linéaires, dépendant du second invariant des tenseurs. Il est donc nécessaire de prendre en compte cette particularité dans la routine d'intégration à écrire.  
Par voie de conséquence, les composantes de cisaillement du tenseur de contraintes en sortie doivent également être multipliées par le même coefficient  $\sqrt{2}$ . Ceci n'a pas en principe de conséquence sur la matrice d'élasticité, ni sur la matrice tangente.
- b. L'argument OPTION est important car il permet de déterminer les calculs à effectuer. En particulier, l'option *RIGI\_MECA\_TANG* n'est destinée à calculer qu'une matrice tangente de prédiction, construite à partir de *DSIDEP*. Il faut prendre garde dans la programmation à ne pas utiliser dans ce cas les arguments *SIGP* et *VIP*, dont la place mémoire n'est pas allouée pour cette option.

La routine NMTOTO sera organisée de la manière suivante :

```
subroutine NMTOTO (NDIM, IMATE, TYPMOD, COMPOR, crit, instam, instap, TM, TP, TREF, EPSM, DEPS, SIGM, VIM, OPTION, SIGP, VIP, DSI DEP)
```

- Lecture des caractéristiques du matériau (élastiques et autres) et calcul de ces caractéristiques aux instants instam et instap en utilisant la routine RCVALA.  
Par exemple, ces caractéristiques peuvent être  $E, \nu, E_T, s_y$ .

On va donc calculer  $E^-, \nu^-, E_T^-, s_y^-$  (i.e. à l'instant instam) et  $E^+, \nu^+, E_T^+, s_y^+$  (i.e. à l'instant instap).

- Quand on va manipuler les contraintes et les déformations, on ne va pas faire des boucles de 1 à 6 mais des boucles de 1 à NDIMSI.

NDIMSI = 4 pour le 2D  
NDIMSI = 6 pour les 3D

- Calcul du seuil (pour les lois à seuils).
- Pour les options *FULL\_MECA* et *RAPH\_MECA* : calcul des contraintes et des variables internes.
- Pour les options *FULL\_MECA* et *RIGI\_MECA\_TANG* : calcul de la matrice de comportement tangente  $\left[ \frac{\partial \sigma}{\partial \varepsilon} \right]$  ou  $\left[ \frac{\partial \sigma}{\partial \Delta \varepsilon} \right]$ .

### 3.2.2 Exemple d'une routine réalisation l'intégration d'une loi de comportement : NMCINE

NMCINE réalise l'intégration d'une relation de comportement d'élasto-plasticité de von Mises à écrouissage cinématique linéaire.

Pour l'intégration de cette relation, on se reportera à la doc [R5.03.02].

Les arguments de NMCINE figurent parmi ceux de la routine générique NMTOTO décrite au [3.1].

On a :

```
subroutine NMCINE(NDIM, IMATE, COMPOR, CRIT, INSTAM, INSTAP, TM, TP, TREF, EPSM, DEPS, SIGM, VIMP, OPTION, SIG, VIP, DSI DEP)
```

Cette routine est organisée de la manière suivante :

- Calcul des caractéristiques élastiques du matériau avec températures TM et TP.  
 Pour cela, on utilise la routine RCVALA de la manière suivante :

```

C
C LECTURE DES CARACTERISTIQUES ELASTIQUES DU MATERIAU (TEMPS - ET +)
      NOMRES(1)='E'
      NOMRES(2)='NU'
      NOMRES(3)='ALPHA4'
      CALL RCVALA (IMATE, 'ELAS',1,'TEMP',TM,2,NOMRES,VALRES,CODRET,FB2)
      CALL RCVALA (IMATE, 'ELAS',1,'TEMP',TM,1,
+               NOMRES(3),VALRES(3) = 0.D0
      IF (CODRET(3) .NE. 'OK') VALRES(3) = 0.D0
      EM      = VALRES(1)
      NUM      = VALRES(2)
      DEUMUM   = EM/(1.D0+NUM)
      TROIKM   = EM/(1.D0-2.D0*NUM0)
      ALPHAM   = VALRES(3)

      CALL RCVALA (IMATE,'ELAS',1,'TEMP',TP,2,NOMRES,VALRES,CODRET,FB2)
      CALL RCVALA (IMATE, 'ELAS',1,'TEMP',TP,1?
+               NOMRES(3),VALRES(3),CODRET(3),BL2)
      IF (CODRET(3) .NE. 'OK') VALRES(3) = 0.D0
      E      = VALRES(1)
      NU      = VALRES(2)
      LAMBDA  = E*NU/551.D0-2*NU)*(1.D0+NU))
      DEUXMU  = E/(1.D0+NU)
      ALPHAP  = VALRES(3)
  
```

On utilise plutôt les coefficients de Lamé  $\lambda$  et  $\mu$  et le modèle de compressibilité K.

- Calcul des caractéristiques d'érouissage ET, SY et C aux températures TM et TP ; pour ET et SY, on utilise la routine RCVALA comme précédemment :

```

C
C LECTURE DES CARACTERISTIQUES D'ECROUISSAGE
      NOMRES(1)='D_SIGM_EPSI'
      NOMRES(2)='SY'
      CALL RCVALA (IMATE, 'ECRO_LINE',1,'TEMP',TM,2,
+               NOMRES,VALRES,CODRET,FB2)
      DSDEM=VALRES(1)
      SIGYM=VALRES(2)
      CM=2.D0/3.D0*DSDEM/(1.D0-DSDEM/EM)
      NOMRES(1)='D_SIGM_EPSI'
      NOMRES(2)='SY'
      CALL RCVALA (IMATE, 'ECRO_LINE',1,'TEMP',TP,2,
+               NOMRES,VALRES,CODRET,FB2)
      DSDE=VALRES(1)
      SIGY=VALRES(2)
      C = 2.D0/3.D0*DSDE/(1.D0-DSDE/E)
  
```

- Calcul de la contrainte d'essai et de sa norme au sens de von Mises :

$$\tilde{\sigma}_e = \tilde{\sigma}^- + 2\gamma\Delta\tilde{\epsilon}$$

Titre : Introduire une nouvelle loi de comportement  
 Auteur(s) : J.M. PROIX, G. BERTRAND

Date : 22/06/05  
 Clé : D5.04.01-B Page : 12/16

Pour affecter les termes de  $\tilde{\mathcal{E}}$ , on fait une boucle de 1 à NDIMSI ainsi qu'on l'a vu au [§3.1] :

```

C
C CALCUL DES CONTRAINTES ELASTIQUES
DO 110 K=1,3
  DEPSTH(K) = DEPS(K) - (ALPHA*(TP-TREF)-ALPHAM*(TM-FREF))
  DEPSTH(K+3) = DEPS(K+3)
110 CONTINUE
  EPSMO = (DEPSTH(1)+ DEPSTH(2)+ DEPSTH(3))/3/D0
  DO 115 K=1,NDIMSI
    DEPSDV(K) = DEPSTH(K) - EPSMO * KRON(K)
115 CONTINUE
C CALCUL DU CRITERE DE VON MISES DE SIGEL
C ON VOIT ICI L'INTERET DU COEF RACINE(2) SUR
C LES CISAILLEMENTS.
  SIGMO = (SIGM(1)+ SIGM(2)+ SIGM(3))/3/D0
  SIELEQ = 0.D0
  DO 114 K=1,NDIMSI
    SIGDV(K) = SIGM(K) - SIGMO*KRON(K)
    SIGDV(K) = DEUXMU/DEUMUM*SIGDV(K)
    SIGEL(K) = SIGDV(K) + DEUXMU*DEPSDV(K)
    SIELEQ = SIELEQ + (SIGEL(K)-C/CM*VIM(K))**2
114 CONTINUE
  SIGMO = TROISK/TROIKM*SIGMO
  SIELEQ = SQRT(1.D5D0*SIELEQ)

```

- Calcul du seuil de plasticité

$$\text{Seuil} = \|\tilde{\mathcal{E}}\| - s_y$$

$$\text{SEUIL} = \text{SIELEQ} - \text{SIGY}$$

- Pour les options RAPH\_MECA et FULL\_MECA, calcul des contraintes et des variables internes à l'instant actuel

- si seuil < 0

On est dans le domaine élastique et les incréments des variables internes sont nuls :

```

C
C CALCUL DES CONTRAINTES ELASTO-PLASTIQUES ET DES VARIABLES INTERNES
  IF (OPTION (1 :9) .EQ.'RAPH_MECA' . OR.
+   OPTION (1 :9) .EQ.'FULL_MECA' THEN
  IF (SEUIL.LT.0.D0) THEN
    VIP(7) = 0.D0
    DP = 0.D0
    SIELEQ = 1.D0
    A1 + 0.D0
    A2 + 0.D0

```

- si seuil ≥

On est dans le domaine élastoplastique et on calcule les incréments des contraintes et des variables internes.

```

  ELSE
    VIP(7) = 1.D0
    DP = SEUIL/(1.5D0*(DEUXMU+C))
    A1 = (DEUXMU/(DEUXMU+C))*(SEUIL/SIELEQ)
    A2 = (C/(DEUXMU+C))*(SEUIL/SIELEQ)
  ENDIF
  PLASTI=VIP(7)
  DO 160 K = 1,NDIMSI
    SIGDV(K) = SIGEL(K) - A1*SIGEL(K)-VIM(K)*C/CM
    SIGP(K) = SIGDV(K) + (SIGMO+TROISK*EPSMO)*KRON(K)
    VIP(K) = VIM(K)*C/CM + A2*(SIGEL(K)-VIM(K)*C/CM)
160 CONTINUE
  ENDIF

```

- Pour les options RIGI\_MECA\_TANG et FULL\_MECA, calcul de la matrice de comportement tangente :

RIGI\_MECA\_TANG calcule  $\frac{\partial \sigma}{\partial \dot{\epsilon}}$

FULL\_MECA calcule  $\frac{\partial \sigma}{\partial \Delta \epsilon}$

Mais on démontre dans notre cas que  $\left[ \frac{\partial \sigma}{\partial \dot{\epsilon}} \right] = \left[ \frac{\partial \sigma}{\partial \Delta \epsilon} \right]$  si  $\Delta p = 0$ , ce qui correspond bien à l'utilisation de RIGI\_MECA\_TANG lors de la phase de prédiction.

On calcule donc la matrice  $\left[ \frac{\partial \sigma}{\partial \Delta \epsilon} \right]$ .

Ce calcul n'est pas l'objet de ce document et est trop long pour être exposé clairement ici [R5.03.02].

On admet donc que l'on a affecté la matrice carrée dsidep avec les valeurs de  $\left[ \frac{\partial \sigma}{\partial \Delta \epsilon} \right]$ .

#### Remarque :

| Cas particulier des contraintes planes.

Pour écrire la matrice tangente, on utilise le fait que, quand on écrit :

$$\Delta \sigma = \left[ \frac{\partial \sigma}{\partial \Delta \epsilon} \right] \Delta \epsilon$$

$$\Delta \sigma_{zz} = 0$$

On en déduit  $\Delta \epsilon_{zz}$  en fonction de  $\Delta \epsilon_{xx}$ ,  $\Delta \epsilon_{yy}$  et  $\Delta \epsilon_{xy}$ , et on injecte cette expression de  $\Delta \epsilon_{zz}$  dans les autres relations.

C'est pour cela que dans le traitement des matrices tangentes dans le cas des contraintes planes, on trouve les instructions suivantes :

```
C -- 8.3 CORRECTION POUR LES CONTRAINTES PLANES :
  IF (CPLAN) THEN
    DO 136 K=1, NDIMSI
      IF (K.EQ.3) GO TO 136
      DO 137 L=1, NDIMSI
        IF (L.EQ.3) GO TO 137
        DSIDEP (K,L)=DSIDEP(K,L)
          + 1.D0/DSIDEP(3,3)*DSIDEP(K,3)*DSIDEP(3,L)
137      CONTINUE
136      CONTINUE
      ENDIF
```

## 3.3 Programmation d'une loi dans l'environnement PLASTI

### 3.3.1 Introduction

Informatiquement, on passe par la routine NMCOMP qui appelle la routine REDECE. PLASTI est appelée par REDECE.

L'environnement PLASTI est décrit dans la documentation [R5.03.10] : 'Relation de comportement élasto-viscoplastique du LMARC'.

### 3.3.2 Algorithme de résolution du problème quasi-statique

On cherche à vérifier l'équilibre de la structure à chaque instant. Sous forme incrémentale, il s'agit d'un problème non linéaire dont la formulation variationnelle dans le cas des petites déformations peut se mettre sous la forme :

Trouver  $\Delta u$  tel que :

$$\begin{cases} \int_{\Omega} \sigma(\varepsilon(u + \Delta u), t) \varepsilon(v) d\Omega = L(t) & \forall v \text{ cinématiquement} \\ Bu = u^d(t) & \text{admissible et } \forall t \end{cases}$$

où  $u$  désigne le champ de déplacement,  $Bu = u^d(t)$  correspond aux conditions aux limites en déplacement et  $L(t)$  est le chargement à l'instant  $t$ .

On est donc conduit à résoudre, pour chaque incrément de temps  $\Delta t$  :

$$F_{t+\Delta t}(u_t + \Delta u) = 0 \quad \text{en partant d'un état à l'équilibre } F_0 = 0$$

$\Delta u$  étant l'incrément de la solution  $u$  sur  $\Delta t$ ,  $u_t$  étant connu

Le schéma général adopté par *Aster* pour résoudre ce système global discrétisé est une méthode de Newton qui s'écrit,  $k$  étant l'indice d'itération :

$$\begin{cases} \frac{\partial F}{\partial \Delta u_k} d(\Delta u_k) = -F(\Delta u_k) \\ \Delta u_{k+1} = \Delta u_k + d(\Delta u_k) \end{cases}$$

Ce schéma nécessite, à partir de l'estimation des déplacements à l'itération  $k$ , de calculer en chaque point de Gauss :

$\sigma_{t+\Delta t}$  qui vérifie la loi de comportement

$$M_{t+\Delta t}^C = \left( \frac{\partial \sigma}{\partial \varepsilon} \right)_{t+\Delta t} \quad \text{l'opérateur de comportement tangent}$$

$$\left( \frac{\partial F}{\partial \Delta u} = K = \sum_e K_e \text{ avec } K_e = \int_{\Omega} B^T \left( \frac{\partial \sigma}{\partial \varepsilon} \right) B d\Omega \right)$$

### 3.3.3 Environnement Plasti

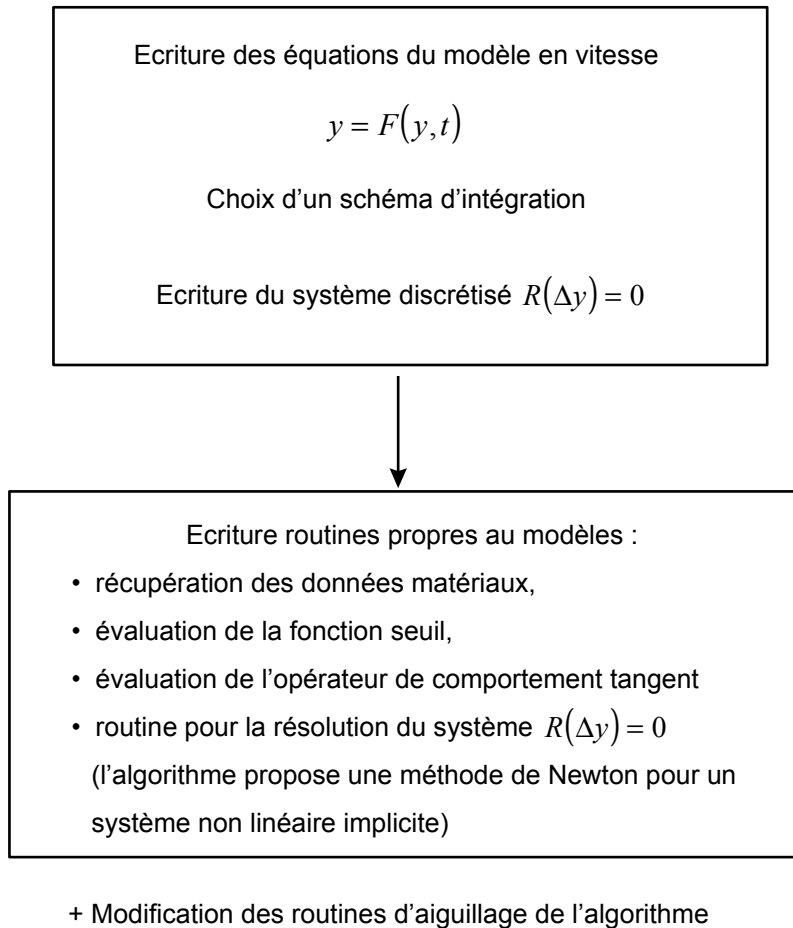
Il est donc nécessaire, à chaque itération globale et en chaque point de Gauss, d'intégrer les équations du modèle pour le calcul  $\sigma_{t+\Delta t}$  et de calculer l'opérateur de comportement tangent.

Un environnement a été créé dans *Code\_Aster* dans le but de paramétrer l'implantation de modèles elastoviscoplastiques présentant une fonction seuil (domaine d'élasticité).

Cet algorithme :

- gère les choix d'intégration élastique ou (visco) plastique,
- propose différentes routines pour aider à la résolution du système non linéaire (local) formé par les équations du modèle,
- met à jour les variables en fin d'incrément,
- appelle les routines utilisateur pour le calcul de l'opérateur de comportement tangent.

La démarche pour implanter un nouveau modèle peut se schématiser de la manière suivante :



### 3.3.4 Formalisation des équations à résoudre

On a à résoudre les équations suivantes :

- La loi de comportement reliant l'incrément des contraintes à l'incrément des déformations totales avec des variables internes (déformation totale cumulée, centre de la surface de charge, ...).

$$\text{Soit } g(\Delta\sigma, \Delta\varepsilon, \Delta\varepsilon^p, \Delta p, \Delta\text{vari}, \dots) = 0 \quad \text{éq 3.3.4-1}$$

- les lois d'évolution des différentes variables internes :

$$\text{Soit } l(\Delta\sigma, \Delta\varepsilon, \Delta\varepsilon^p, \Delta p, \Delta\text{vari}, \dots) = 0 \quad \text{éq 3.3.4-2}$$

- le critère de plasticité

$$f(\sigma, X_i, p) = 0$$

$$\text{Soit } \|\sigma - X_i\| - R(p) \quad \text{éq 3.3.4-3}$$

- vari* désigne les variables internes autres que  $\varepsilon^p$  et  $p$ ,
- $X_i$  centre de la surface de charge est un exemple de composante de *vari*,
- L'incrément de déformation plastique cumulée  $\Delta p$  est calculé avec [éq 3.3.4-3].

L'équation [éq 3.3.4-1] à 6 inconnues (les 6 composantes du tenseur symétrique des contraintes).

L'équation [éq 3.3.4-3] à 1 inconnue :  $\Delta p$ .

Le nombre d'inconnues relatives à l'équation [éq 3.3.4-2] est égal au nombre de composantes des variables internes autres que la déformation (visco-) plastique et la déformation (visco-) plastique cumulée.

On a à résoudre, comme indiqué au [§3.2.2] :

$$F(\Delta y) = 0$$

$$\text{avec } \Delta y^T = (\Delta \sigma \quad \Delta \text{vari} \quad \Delta p)$$

On résout ce système par une méthode de Newton, soit :

$$\begin{aligned} \frac{\partial F}{\partial y_{k+1}} d[\Delta y_k] &= -F(\Delta y_k) \\ \Delta y_{k+1} &= \Delta y_k + d(\Delta y_k) \end{aligned}$$

D'autre part, on a à calculer la matrice tangente.

On considère que le système  $F(\Delta y) = 0$  est vérifié en fin d'incrément. On perturbe  $F$  selon une petite variation. On considère  $\varepsilon$  comme une variable et non comme un paramètre.

Le système reste à l'équilibre et on vérifie donc que  $dF = 0$ .

$$\text{Soit } \frac{\partial F}{\partial \Delta \sigma} d\Delta \sigma + \frac{\partial F}{\partial \Delta \varepsilon} d\Delta \varepsilon + \frac{\partial F}{\partial \Delta \text{vari}} d\Delta \text{vari} + \frac{\partial F}{\partial \Delta p} d\Delta p = 0$$

On est ainsi conduit à utiliser la même matrice jacobienne qui a servi pour calculer  $F(\Delta y) = 0$  car on peut écrire :

$$\frac{\partial F}{\partial \Delta y} d(\Delta y) = x$$

$$\text{avec } \Delta y^T = [\Delta \sigma \quad \Delta \text{vari} \quad \Delta p]$$

$$\text{et } x = [H d\Delta \varepsilon \quad 0 \quad 0]$$

Par substitutions et éliminations successives, on obtient  $K d\Delta \sigma = H d\Delta \varepsilon$

$$\text{d'où l'opérateur tangent recherché } \left[ \frac{\partial}{\partial \Delta \varepsilon} \Delta \sigma \right]_{t+\Delta t} = [K^{-1} H]$$

L'expression de  $K^{-1}$  est difficile à déterminer, aussi utilise-t-on un solveur LU pour l'évaluer.