

---

## Modèles de grandes déformations GDEF\_LOG et GDEF\_HYPO\_ELAS

---

### Résumé :

Ce document présente les formulations hypoélastiques de grandes déformations utilisée dans Code\_Aster pour les lois de comportement de type von Mises.

GDEF\_LOG : ce modèle, dû à C.Miehe, N.Apel et M.Lambrecht [13] est un modèle de grandes déformations basé sur une mesure logarithmique, avec un tenseur de contraintes en dualité particulier. Il est valable quel que soit le comportement en petites déformations et présente l'avantage de fournir une matrice tangente symétrique. Aucune modification des variables internes cinématique n'est nécessaire.

GDEF\_HYPO\_ELAS : ce modèle est issu d'une approche due à Simo et Hughes [1]. Elle se base sur la notion d'une configuration tournée objective dans laquelle les dérivées sont réalisées. Cette approche générique permet de traiter les lois de comportement à écrouissages isotrope et cinématique, avec ou sans viscosité, avec une formulation de type hypoélastique, .

Ces deux modèles permettent une intégration incrémentalement objective des lois de comportement comme le modèle SIMO\_MIEHE. Toutefois, comme toutes les lois hypoélastiques, ces lois de comportement sont en toute rigueur limitées aux faibles déformations élastiques.

On illustre dans ce document les capacités de ces modèles et les avantages par rapport à l'approximation PETIT\_REAC.

## Table des Matières

1	Introduction.....	3
2	Algorithme GDEF_HYPO_ELAS.....	3
2.1	Éléments de cinématique.....	3
3	Lois de comportement hypo-élastoplastiques.....	5
3.1	Intégration objective : algorithme GDEF_HYPO_ELAS.....	6
3.2	Algorithme d'intégration .....	7
3.3	Choix du tenseur antisymétrique et intégration.....	9
3.4	Avantages et limitations de la méthode.....	9
4	Algorithme GDEF_LOG .....	11
4.1	Description sommaire .....	11
4.2	Algorithme .....	12
5	Validité des modèles de grandes déformations.....	15
5.1	Identification des paramètres.....	15
6	Comparaison avec PETIT_REAC .....	15
6.1	Approximation des grandes déformations par PETIT_REAC.....	15
6.2	Comparaison sur un exemple.....	17
7	Bibliographie.....	19
8	Description des versions du document.....	19
9	Annexe 1 : opérateur tangent pour GDEF_HYPO_ELAS.....	20
9.1	Principe du calcul de la matrice tangente : écriture variationnelle et expression générale .....	21
9.2	Calcul de la matrice tangente pour les deux formalismes.....	22
	Terme 1 .....	23
	Terme 2 .....	23
	Terme 3 .....	25
	Terme 4 .....	26
	Terme 5 .....	27
	Terme 6 .....	28
	Bilan .....	28
10	Annexe 2 : Calcul des déformations logarithmiques.....	30
10.1	Notations :.....	30
10.2	Expression des contraintes en configuration Lagrangienne .....	30
10.3	Expression de l'opérateur tangent en configuration Lagrangienne .....	31
10.4	Calcul effectif des déformations logarithmiques.....	31

## 1 Introduction

La plupart des lois de comportement de Code\_Aster sont utilisables sous l'hypothèse des petites perturbations (HPP), qui permet de confondre les configurations géométriques initiale et actuelle. Cependant, lorsque les déformations deviennent importantes (on fixe en général la limite à 5%), cette hypothèse n'est plus vérifiée. Les notions de dérivées particulaire et partielle sont alors différentes, et de ce fait les lois de comportement formulées incrémentalement perdent leur caractère objectif (indépendance de l'état mécanique en fonction de l'observateur) ; une conséquence ennuyeuse est l'évolution possible des contraintes pour un mouvement de corps rigide, contraire à la physique.

Afin de retrouver l'objectivité, indispensable donc pour garantir une bonne fiabilité du résultat, différentes stratégies grandes déformations sont possibles. L'objet de ce document est de présenter les formalismes mis en place dans Code\_Aster pour traiter les lois de comportement à écrouissages isotrope et cinématique et critère de Von Mises.

On présente premièrement le formalisme hypoélastique de Simo-Hughes [1] utilisable dans Code\_Aster *via* le mot clé `DEFORMATION=GDEF_HYPO_ELAS` de l'opérateur `COMP_INCR`. On présente ensuite un résultat permettant de vérifier l'objectivité de cette intégration et de vérifier que le formalisme `PETIT_REAC` est mis en défaut.

Dans une deuxième partie, on présente le formalisme `GDEF_LOG`, dû à C.Miehe, N.Apel et M.Lambrecht [13] qui est un modèle de grandes déformations basé sur une mesure logarithmique, avec un tenseur de contraintes en dualité particulier. Il est valable quelque soit le comportement en petites déformations et présente l'avantage de fournir une matrice tangente symétrique. Aucune modification des variables internes cinématique n'est nécessaire.

## 2 Algorithme GDEF\_HYPO\_ELAS

### 2.1 Éléments de cinématique

Les éléments cinématiques dans le cas continu peuvent être trouvés par exemple dans [3]. On s'intéressera ici au cas directement discrétisé en temps permettant de définir les grandeurs utilisées dans le formalisme présenté dans ce document.

On considère un domaine continu initial fermé  $\Omega_0 \subset \mathbb{R}^3$ , dont chaque point est repéré par ses coordonnées  $X \in \Omega_0$ , subissant un champ de déformation  $\varphi$  le faisant passer dans la configuration  $\Omega$  :

$$\varphi : \Omega_0 \rightarrow \Omega \subset \mathbb{R}^3 \quad \text{Eq1}$$

On notera  $x \in \Omega$  les coordonnées de ce point dans la configuration courante.

La déformée évoluant au cours du temps, on définit en réalité, par le truchement de la discrétisation temporelle, une famille de champ  $\varphi_n$  correspondant chacun à un instant  $t_n$  de l'histoire d'évolution du domaine.

Dans le cas du formalisme grandes déformations traité ici, il est nécessaire d'introduire 4 configurations pour le domaine et son évolution (cf. Figure 1) : la configuration  $\Omega_0$  initiale de référence (c'est-à-dire pour laquelle les déformations sont nulles), la configuration  $\Omega_n$  au début du pas de temps courant  $t_{n+1} = t_n + \Delta t$ , la configuration  $\Omega_{n+1}$  à la fin de ce pas de temps, et une configuration milieu du pas de temps,  $\Omega_{n+\frac{1}{2}}$ , le formalisme étant intégré avec une règle de point milieu.

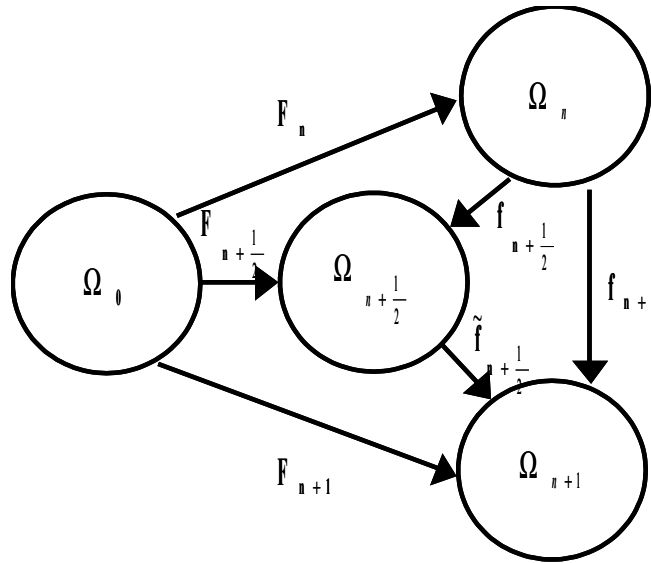


Figure 1 : Configurations nécessaires et gradients de transformation

A partir de ces configurations, on définit les champs de déplacements et les gradients de transformation pour passer de l'une à l'autre. Les grandeurs faisant passer de la configuration initiale à une configuration donnée sont notées en majuscule ( $U, F$ ) et les grandeurs reliant deux configurations déformées entre elles sont notées en minuscule ( $u, f$ ). Le Tableau 1 récapitule les différentes grandeurs et leurs expressions.

Configuration de départ	Configuration d'arrivée	Déplacement	Gradient de transformation
$\Omega_0$	$\Omega_n$	$U_n$	$F_n = I_d + grad_0 U_n$
$\Omega_0$	$\Omega_{n+1}$	$U_{n+1}$	$F_{n+1} = I_d + grad_0 U_{n+1}$
$\Omega_0$	$\Omega_{n+\frac{1}{2}}$		$F_{n+\frac{1}{2}} = \frac{1}{2}(F_n + F_{n+1})$
$\Omega_n$	$\Omega_{n+\frac{1}{2}}$		$f_{n+\frac{1}{2}} = I_d + \frac{1}{2} grad_n u$ $= F_{n+\frac{1}{2}} \cdot F_n^{-1}$ $= \frac{1}{2}(f_{n+1} + I_d)$
$\Omega_n$	$\Omega_{n+1}$	$u$	$f_{n+1} = I_d + grad_n u = \Delta F$ $= F_{n+1} \cdot F_n^{-1}$
$\Omega_{n+\frac{1}{2}}$	$\Omega_{n+1}$		$\tilde{f}_{n+\frac{1}{2}} = f_{n+1} \cdot f_{n+\frac{1}{2}}^{-1}$ $= F_{n+1} \cdot F_{n+\frac{1}{2}}^{-1}$

Tableau 1 : récapitulatif des déplacements et gradients de transformation

A partir de ces gradients de transformation, il est possible de définir les taux de déformation  $\mathbf{L}$  ainsi que les tenseurs taux de rotation  $\boldsymbol{\omega}$  et taux de déformation  $\mathbf{d}$  :

$$\begin{cases} \mathbf{L} \cdot &= \dot{\mathbf{F}} \cdot \mathbf{F}^{-1} \\ \mathbf{d} \cdot &= \frac{1}{2}(\mathbf{L} \cdot + \mathbf{L} \cdot^T) \\ \boldsymbol{\omega} \cdot &= \frac{1}{2}(\mathbf{L} \cdot - \mathbf{L} \cdot^T) \end{cases} \quad \text{Eq 2}$$

avec  $\cdot$  désignant la configuration  $n$ ,  $n+1$  ou  $n+\frac{1}{2}$ .

Le tenseur de déformation Eulérien entre les configurations  $\Omega_n$  et  $\Omega_{n+1}$  se déduit de ces définitions :

$$\mathbf{e}_{n+1} = \frac{1}{2}[\mathbf{I}_d - (\mathbf{f}_{n+1} \mathbf{f}_{n+1}^T)^{-1}] \quad \text{Eq 3}$$

ce qui permet d'écrire  $\mathbf{d}_{n+\frac{1}{2}} = \frac{1}{\Delta t} \mathbf{f}_{n+\frac{1}{2}}^{-1} \mathbf{e}_{n+1} \mathbf{f}_{n+\frac{1}{2}}^{\sim}$  (le taux de déformation est alors bien lié à la déformation eulérienne)

La dernière grandeur à introduire pour notre algorithme est le gradient de déplacement incrémental, relatif à la configuration  $\Omega_{n+\frac{1}{2}}$  et défini par :

$$\mathbf{h}_{n+\frac{1}{2}} = \text{grad}_n \mathbf{u} \mathbf{f}_{n+\frac{1}{2}}^{-1} \quad \text{Eq 4}$$

Ce dernier permet de déterminer le taux de rotation dans la même configuration par la relation :

$$\boldsymbol{\omega}_{n+\frac{1}{2}} = \frac{1}{2\Delta t} [\mathbf{h}_{n+\frac{1}{2}} - \mathbf{h}_{n+\frac{1}{2}}^T] \quad \text{Eq 5}$$

A partir de ces éléments de cinématique, il est possible de définir des lois de comportement hypoélastiques dont l'intégration est objective en grandes déformations. Le paragraphe suivant présente ce type de formulation des lois de comportement.

## 3 Lois de comportement hypo-élastoplastiques

Dans cette section, la classe de modèle phénoménologique de plasticité (ici indépendante du temps) avec hypoélasticité est considérée. Elle constitue une extension *ad hoc* de l'écriture des lois en petites déformations, ce qui permet une certaine généralité et représente un avantage dans le contexte d'un code de calcul : on verra dans le chapitre suivant qu'il est possible de réaliser son intégration numérique de façon équivalente à celle des petites déformations.

Cette classe de modèles est à opposer à la classe hyperélastique, basée sur l'approche thermodynamique des milieux continus. Dans ce contexte, une énergie libre, pouvant par exemple être considérée comme une fonction de la température et la déformation de Green-Lagrange, est définie ; les évolutions des contraintes et éventuellement des variables internes en découlent. On peut citer par exemple le cas de la loi hyperélastique de Signorini (cf. [4]) en élasticité et le formalisme Simo-Miehe en hyperélastoplasticité (cf. [3]).

Une loi de comportement hypo-élastoplastique se construit généralement en cinq étapes.

(i) À l'instar de la décomposition additive des petites déformations, le taux de déformation  $\mathbf{d}$  est tout d'abord décomposé en une partie élastique et une partie plastique :

$$\mathbf{d} = \mathbf{d}^e + \mathbf{d}^p \quad \text{Eq 6}$$

(ii) Une dérivée de la contrainte de Kirchhoff  $\boldsymbol{\tau} = \det(\mathbf{F})\boldsymbol{\sigma}$  est alors déterminée par une relation incrémentale fonction du taux élastique de déformation:

$$\dot{\boldsymbol{\tau}} = \mathbf{C} : [\mathbf{d} - \mathbf{d}^p] \quad \text{Eq7}$$

avec  $\dot{\boldsymbol{x}}$  une dérivée objective à définir et  $\mathbf{C}$  le tenseur d'élasticité.

(iii) On construit un domaine de réversibilité convexe définissant l'espace admissible des contraintes à partir d'une fonction  $f$  :

$$E_{\boldsymbol{\tau}} = \{(\boldsymbol{\tau}, \mathbf{q}, \alpha) \in \mathcal{S} \times \mathcal{R}^{m+1} \mid f(\boldsymbol{\tau}, \mathbf{q}, \alpha) \leq 0\} \quad \text{Eq 8}$$

avec  $\mathcal{S}$  l'espace des contraintes et  $\mathbf{q}$  l'ensemble des  $m$  variables internes représentant l'érouissage cinématique du matériau et  $\alpha$  les variables scalaires (y compris l'érouissage isotrope).

(iv) Les lois d'évolution de ces variables internes suivent un principe de normalité (on ne considère ici que les lois de comportement associées):

$$\begin{aligned} \mathbf{d}^p &= \gamma \frac{\partial f(\boldsymbol{\tau}, \mathbf{q}, \alpha)}{\partial \boldsymbol{\tau}} \\ \dot{\mathbf{q}} &= -\gamma \mathbf{g}(\boldsymbol{\tau}, \mathbf{q}, \alpha) \end{aligned} \quad \text{Eq9}$$

avec  $\gamma \geq 0$  le multiplicateur plastique,  $\frac{\partial f}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau}, \mathbf{q}, \alpha)$  définissant la direction d'écoulement plastique et  $\mathbf{g}(\boldsymbol{\tau}, \mathbf{q}, \alpha)$  l'évolution des autres variables internes.

(v) L'écriture des conditions de charge/décharge, classiquement représentées par Kuhn-Tucker et la condition de cohérence :

$$\begin{cases} \gamma \geq 0 \\ f(\boldsymbol{\tau}, \mathbf{q}, \alpha) \leq 0 \\ \gamma f(\boldsymbol{\tau}, \mathbf{q}, \alpha) = 0 \end{cases} \quad \text{Eq 10}$$

Cette classe de modèle se caractérise donc par une analogie forte avec les formalismes petites déformations, avec une écriture incrémentale des contraintes qui n'est pas sans poser quelques difficultés d'intégration numérique : en effet, afin d'empêcher les évolutions de contraintes par un mouvement de corps rigide, il convient d'avoir une intégration objective de l'équation (Eq7).

L'objet du chapitre suivant est de décrire l'algorithme d'intégration objectif utilisé dans Code\_Aster, dû à Simo-Hughes.

## 3.1 Intégration objective : algorithme GDEF\_HYPO\_ELAS

Le chapitre précédent a défini la classe de matériau hypo-élastique, dont l'une des caractéristiques est de calculer les contraintes incrémentalement. Il convient d'en avoir une intégration objective afin de se prémunir d'une évolution des contraintes lors d'un mouvement de corps rigide.

Pour cela, deux grand types d'approches existent : les approches par définition d'une dérivée objective (de type Jaumann ou Lie), et les approches par définition d'un référentiel tourné objectif dans lequel les dérivées sont calculées. Ces approches sont en fait équivalentes (voir [1]) pour le problème continu, mais l'intégration des dérivées objectives posent souvent problème (cf. [8]).

L'algorithme décrit ici se base sur un référentiel tourné, dû à Simo-Hughes [1]. L'idée est d'utiliser une description annulant les effets des rotations dans les dérivées temporelles. Ce type d'approche se formalise ainsi. Soit  $\tilde{\boldsymbol{\theta}}$  un tenseur antisymétrique d'ordre 2 quelconque. On peut construire l'équation d'évolution d'un tenseur orthogonal propre  $\boldsymbol{\Lambda}(\mathbf{X}, t) \in SO_3$  à partir de  $\tilde{\boldsymbol{\theta}}$  de la forme suivante :

$$\begin{aligned}\dot{\Lambda} &= (\tilde{\theta} \circ \varphi) \\ \Lambda|_{t=0} &= \mathbf{1}\end{aligned}\quad \text{Eq11}$$

À  $\tilde{\theta}$  donné, les solutions de (Eq11) génèrent un sous-groupe du groupe spécial orthogonal (sous-groupe défini par un paramètre), représentant un référentiel local tourné.

On peut alors définir la contrainte de Kirchhoff et le taux de déformation dans ce référentiel :

$$\begin{aligned}\Sigma &:= \Lambda^T \tau \Lambda \\ \mathbf{D} &= \Lambda^T d \Lambda\end{aligned}\quad \text{Eq12}$$

La dérivée temporelle de la contrainte de Kirchhoff tournée est alors :

$$\dot{\Sigma} = \Lambda^T [\dot{\tau} + \tau \theta - \theta \tau] \Lambda := \Lambda^T \overset{\circ}{\tau} \Lambda \quad \text{Eq13}$$

ce qui fait apparaître une dérivée objective (d'où l'équivalence entre les approches). Du choix de  $\tilde{\theta}$  dépend la dérivée objective obtenue.

La loi de comportement de type hypo-élastoplastique définie au paragraphe 3 est réécrite dans ce référentiel :

$$\begin{aligned}(i) \quad & \mathbf{D} := \Lambda^T \mathbf{D} \Lambda ; \mathbf{D} = \mathbf{D}^e + \mathbf{D}^p \\(ii) \quad & \dot{\Sigma} = \mathbf{a} : [\mathbf{D} - \mathbf{D}^p] \\(iii) \quad & E_{\Sigma} = \{(\Sigma, \mathbf{Q}, \alpha) \in \mathcal{S} X \mathcal{R}^m | F(\Sigma, \mathbf{Q}, \alpha) \leq 0\} \\(iv) \quad & \mathbf{D}^p = \gamma \frac{\partial F(\Sigma, \mathbf{Q}, \alpha)}{\partial \Sigma} ; \dot{\mathbf{Q}} = -\gamma \mathbf{g}(\Sigma, \mathbf{Q}, \alpha) \\(v) \quad & \gamma \geq 0 ; F(\tau, \mathbf{q}, \alpha) \geq 0 ; \gamma F(\tau, \mathbf{q}, \alpha) = 0 ; \gamma \dot{F}(\tau, \mathbf{q}, \alpha)\end{aligned}\quad \text{Eq14}$$

Ce système d'équations est intégré numériquement par un algorithme de point milieu décrit ci-après.

## 3.2 Algorithme d'intégration

(i) La première étape consiste à discrétiser temporellement les équations (Eq14). Une règle de point milieu classique est tout d'abord utilisée pour la dérivée de la contrainte de Kirchhoff dans le repère tourné (Eq14 (ii)) :

$$\begin{aligned}\Sigma_{n+1} - \Sigma_n &= \Delta t \dot{\Sigma}_{n+\frac{1}{2}} \\ &= \Delta t \mathbf{a}_{n+\frac{1}{2}} : \mathbf{D}_{n+\frac{1}{2}} \\ &= \mathbf{a}_{n+\frac{1}{2}} : \Lambda_{n+\frac{1}{2}}^T [\Delta t d_{n+\frac{1}{2}}] \Lambda_{n+\frac{1}{2}} \\ &= \mathbf{a}_{n+\frac{1}{2}} : \Lambda_{n+\frac{1}{2}}^T [\mathbf{f}_{n+\frac{1}{2}}^{\tilde{\tau}} e_{n+1} \mathbf{f}_{n+\frac{1}{2}}^{\sim}] \Lambda_{n+1}\end{aligned}\quad \text{Eq15}$$

et, finalement, en posant  $e_{n+\frac{1}{2}}^{\sim} = \mathbf{f}_{n+\frac{1}{2}}^{\tilde{\tau}} e_{n+1} \mathbf{f}_{n+\frac{1}{2}}^{\sim}$  :

$$\Sigma_{n+1} = \Sigma_n + \mathbf{a}_{n+\frac{1}{2}} : \Lambda_{n+\frac{1}{2}}^T e_{n+\frac{1}{2}}^{\sim} \Lambda_{n+\frac{1}{2}} \quad \text{Eq16}$$

L'équation (Eq16) permet de voir que l'algorithme est objectif (invariance de la contrainte par rotation de corps rigide) :

$$\mathbf{f}_{n+1} \in SO_3 \Leftrightarrow e_{n+1} = \mathbf{0} \Leftrightarrow \Sigma_{n+1} = \Sigma_n \quad \text{Eq17}$$

(ii) Cette intégration de la contrainte est ensuite couplée à un algorithme de retour radial tout à fait similaire à celui utilisé en petites déformations. L'algorithme complet qui découle pour un critère de

type von Mises avec écrouissage isotrope et/ou cinématique est décrit dans la Boîte 1 ci-après. En pratique, pour les cas de plasticité traités, la matrice élastique est constante, sauf à introduire une variable de commande comme la température ; dans ce cas toutefois, c'est la matrice élastique à l'instant  $t_{n+1}$  qui est en réalité utilisée.

On retrouve donc le même type de schéma que dans [R5.03.02] ou [R5.03.16].

**Remarque :**

L'intégration directe de l'équation (Eq14) (ii) par une dérivée de type Jaumann  $\overset{\circ}{\sigma}^J = a : [D - D^p] = \dot{\sigma} + \sigma \theta - \theta \sigma$  est nécessairement explicite sur le terme de transport de la dérivée, engendrant des contraintes parasites qui tendent vers 0 avec le pas de charge. C'est ce type de difficulté qui est contournée par l'utilisation d'un repère local objectif.

(i) Connaissant les champs de déplacement, calculer les gradients de transformation :

$$f_{n+1} = I_d + grad_n u, f_{n+\frac{1}{2}} = I_d + \frac{1}{2} grad_n u, f_{n+\frac{1}{2}}^{\sim} = f_{n+1} \cdot f_{n+\frac{1}{2}}^{-1}$$

(ii) Incrément de déformation dans le référentiel tourné :

$$e_{n+\frac{1}{2}}^{\sim} = \frac{1}{2} f_{n+\frac{1}{2}}^{\sim} [I_d - (f_{n+1} f_{n+1}^T)^{-1}]_{n+1} f_{n+1}$$

(iii) Tenseurs de rotation relatifs :

$$r_{n+1} = \Lambda_{n+1} \Lambda_n^T, r_{n+\frac{1}{2}} = \Lambda_{n+1} \Lambda_{n+\frac{1}{2}}^T$$

(iv) Prédiction Élastique :

$$\tau_{n+1}^{pred} = r_{n+1} \tau_n r_{n+1}^T + a : [r_{n+\frac{1}{2}} e_{n+\frac{1}{2}}^{\sim} r_{n+\frac{1}{2}}^T]$$

$$q_{n+1}^{pred} = r_{n+1} q_n r_{n+1}^T$$

$$\eta_{n+1}^{pred} = dev[\tau_{n+1}^{pred} - q_{n+1}^{pred}]$$

$$f_{n+1}^{pred} = \|\eta_{n+1}^{pred}\| - \sqrt{\frac{2}{3}} (\sigma_Y + K \alpha_n)$$

(v) Correction plastique si nécessaire :

SI  $f_{n+1}^{pred} < 0$  ALORS  $(\cdot)_{n+1} = (\cdot)_{n+1}^{pred}$

SINON

$$\Delta \gamma = \frac{\langle f_{n+1}^{pred} \rangle / 2\mu}{1 + \frac{H+K}{3\mu}}$$

$$\tau_{n+1} = \tau_{n+1}^{pred} - 2\mu \Delta \gamma \frac{\eta_{n+1}^{pred}}{\|\eta_{n+1}^{pred}\|}$$

$$q_{n+1} = q_{n+1}^{pred} + \frac{2}{3} \mu \Delta \gamma H \frac{\eta_{n+1}^{pred}}{\|\eta_{n+1}^{pred}\|}$$

$$\alpha_{n+1} = \alpha_n + \sqrt{\frac{2}{3}} \Delta \gamma$$



FIN SI

Tableau 3.2-1 : Boîte 1 - algorithme de résolution

(iii) Le dernier point concerne le choix du tenseur antisymétrique  $\tilde{\theta}$  et donc du calcul des tenseurs rotations  $\Lambda$  qui découlent de l'équation (Eq11).

### 3.3 Choix du tenseur antisymétrique et intégration

Le choix qui est fait dans Code\_Aster est d'utiliser le tenseur taux de rotation  $\omega$  ; ce choix conduit à un algorithme équivalent à une dérivée objective de Jaumann (mais avec une intégration incrémentalement objective).

L'étape (iii) de la Boîte 1 renferme par conséquent les étapes décrites ci-après dans la Boîte 2. La dernière étape de la Boîte 2, qui constitue l'intégration de l'équation (Eq11), est liée à la règle du point milieu dans  $SO_3$ . On pourra se référer à [1] pour une démonstration de cette relation.

(i)	Gradient de déplacement incrémental $\mathbf{h}_{n+\frac{1}{2}} = \text{grad}_n \mathbf{u} \mathbf{f}_{n+\frac{1}{2}}^{-1}$
(ii)	Tenseur taux de rotation $\boldsymbol{\omega}_{n+\frac{1}{2}} = \frac{1}{2\Delta t} [\mathbf{h}_{n+\frac{1}{2}} - \mathbf{h}_{n+\frac{1}{2}}^T]$
(iii)	Tenseurs rotation des incréments suivants $\Lambda_{n+\frac{1}{2}} = \exp\left[\frac{\Delta t}{2} \boldsymbol{\omega}_{n+\frac{1}{2}}\right] \Lambda_n$ $\Lambda_{n+1} = \exp[\Delta t \boldsymbol{\omega}_{n+\frac{1}{2}}] \Lambda_n$

Tableau 3.3-1 : Boîte 2 - Intégration des tenseurs rotation

De plus, le calcul du terme exponentiel dans la Boîte2 (iii) est en pratique réalisé par la formule d'Euler

Rodrigues :  $\exp(\theta) = 1 + \frac{\sin(\|\theta\|)}{\|\theta\|} + \dots$ , cf. [9]

### 3.4 Avantages et limitations de la méthode

L'algorithme décrit ci-dessus présente des avantages et des limitations qui sont rappelées ici. Une étude comparative des différentes approches en grandes déformations peut être trouvée dans [5].

Parmi les avantages citons les suivants :

- Généricité de l'algorithme : il permet de traiter un grand nombre de lois de comportement sans modifications particulières ; ainsi, les lois suivantes de Code\_Aster sont actuellement disponibles, correspondant aux écrouissages isotrope et cinématique, ainsi qu'à la viscoplasticité : VMIS\_ISOT\_LINE, VMIS\_ISOT\_TRAC, VMIS\_ISOT\_PUIS, VMIS\_CINE\_LINE, VMIS\_CIN2\_MEMO, VMIS\_ECMI\_LINE, VMIS\_ECMI\_TRAC, VMIS\_CIN1\_CHAB, VMIS\_CIN2\_CHAB, LEMAITRE, VENDOCHAB, VISC\_CIN1\_CHAB, VISC\_CIN2\_CHAB, VISC\_CIN2\_MEMO
- Similitude avec les petites déformations : cette généralité est liée à la similitude entre l'algorithme de retour radial présenté et celui classiquement utilisé en petites déformations ; seules les contraintes et les variables cinématiques d'entrée et sortie doivent être affectées par une rotation relative de  $\mathbf{r}$

- Objectivité garantie : comme souligné par l'Eq17, l'algorithme permet de se prémunir de toute évolution de contraintes par un mouvement de corps rigide.

Il présente tout de même quelques inconvénients :

- Il est tout d'abord limité aux déformations élastiques isotropes et relativement faibles (hypoélasticité) ; au niveau plus théorique, les lois hypoélastiques peuvent conduire à une dissipation non nulle durant un cycle de déformation fermé élastique, ce qui n'est pas physique ; du fait de ces inconvénients, on préférera autant que possible le formalisme SIMO\_MIEHE pour les lois à écrouissage isotrope.
- Seules les lois de comportement de type VMIS sont traitées : on ne traite pas des cas de critère de Tresca ou de Hill par exemple ; de plus, pour les lois à écrouissage cinématique, il faut transporter le tenseur cinématique à l'instant précédent dans le référentiel objectif à l'instant actuel ;
- Les temps de calcul sont relativement importants ; ceci est dû à un algorithme coûteux et à une matrice tangente non symétrique qui n'est pas optimale (voir annexes), ce qui augmente le nombre d'itérations de Newton à convergence. On constate que le nombre d'itérations à convergence est notablement plus important que celui observé en utilisant le formalisme hyper-élastique de Simo Miehe . L'étude analytique de la possibilité d'écrire une matrice tangente pour ces modèles est décrite en annexe.

## 4 Algorithme GDEF\_LOG

Cet algorithme, dû à C.Miehe; N.Apel et M.Lambrecht [13] présente les mêmes avantages que ceux choisis jusqu'à présent mais ils basé sur une formulation énergétique et la matrice de rigidité est fournie dans [13].

### 4.1 Description sommaire

Le modèle se base sur une déformation logarithmique définie par :  $\mathbf{E} = \frac{1}{2} \log[\mathbf{F}^T \cdot \mathbf{F}]$ . La définition de cette expression est fournie en annexe 2.

La contrainte  $\mathbf{T}$  est définie dans l'espace logarithmique comme duale de  $\mathbf{E}$ , de sorte que la densité de puissance mécanique s'exprime  $\mathbf{T} : \dot{\mathbf{E}}$ . Ce n'est pas un tenseur de contraintes classique, mais on peut en déduire les tenseurs habituels :

$\mathbf{\Pi} = \mathbf{T} : \mathbf{P}_{\Pi}$ , avec  $\mathbf{P}_{\Pi}$  un tenseur de projection (d'ordre 4 en 3D) et  $\mathbf{\Pi}$  le tenseur des contraintes de Piola-Kirchhoff de première espèce. Les tenseurs de Cauchy  $\boldsymbol{\sigma}$  et de Kirchhoff  $\boldsymbol{\tau}$  s'écrivent de façon habituelle  $J \boldsymbol{\sigma} = \boldsymbol{\tau} = \mathbf{\Pi} \mathbf{F}^T$

En effet la puissance mécanique s'écrivant  $\mathbf{T} : \dot{\mathbf{E}} = \mathbf{\Pi} : \dot{\mathbf{F}}$ , on obtient :

$$\mathbf{T} : \dot{\mathbf{E}} = \mathbf{T} : \frac{\partial \mathbf{E}}{\partial \mathbf{F}} : \dot{\mathbf{F}} = \mathbf{\Pi} : \dot{\mathbf{F}} = \mathbf{T} : \mathbf{P}_{\Pi} : \dot{\mathbf{F}} \text{ ce qui définit le tenseur de projection } \mathbf{P}_{\Pi} = \frac{\partial \mathbf{E}}{\partial \mathbf{F}}.$$

On peut calculer également le second tenseur de Piola-Kirchhoff  $\mathbf{S}$  en fonction de  $\mathbf{T}$  :

$\mathbf{T} : \dot{\mathbf{E}} = \mathbf{S} : \dot{\Delta} = \mathbf{S} : \frac{1}{2} \dot{\mathbf{C}}$ , avec  $\Delta = \frac{1}{2} (\mathbf{C} - \mathbf{I}_d) = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}_d)$  le tenseur des déformations de Green-Lagrange.

Alors  $\mathbf{T} : \dot{\mathbf{E}} = \mathbf{T} : \frac{\partial \mathbf{E}}{\partial \mathbf{C}} : \dot{\mathbf{C}} = \mathbf{S} : \frac{1}{2} \dot{\mathbf{C}}$  donc  $\mathbf{S} = \mathbf{T} : \mathbf{P}$  avec  $\mathbf{P} = 2 \frac{\partial \mathbf{E}}{\partial \mathbf{C}}$

Si la physique choisie est particulière, le modèle permet toutefois de garder la décomposition additive des déformations élastiques et plastiques classique en HPP avec  $\mathbf{E}^p = \frac{1}{2} \log(\mathbf{F}^{pT} \cdot \mathbf{F}^p)$ .

Un tel choix est toujours licite. Cela revient simplement à adopter une définition pour la déformation élastique. Or, celle-ci s'avère cohérente avec une décomposition multiplicative, en l'absence de rotation (situation coaxiale). En outre, l'incompressibilité plastique est assurée car :  $tr \mathbf{E}^p = \log J^p$ .

L'énergie élastique  $\psi^e$  du modèle prend également la même forme que celle des petites déformations, mais en adoptant les notions de contrainte et déformation spécifiques à ce formalisme :

$$\psi^e = \frac{1}{2} \|\mathbf{E} - \mathbf{E}^p\|_E^2 = \frac{1}{2} \mathbf{T} : \mathbf{C}^{-1} : \mathbf{T}.$$

Cette formulation présente certains avantages :

- comme pour GDEF\_HYPO\_ELAS, la dimension cinématique du modèle est confinée en amont et en aval de l'intégration du comportement ; ceci était l'un des éléments principaux pour le choix du formalisme ; tous les modèles de comportement disponible en petites déformations sont *a priori* disponibles, à condition bien sûr que cela ait un sens physique ( les grandes déformations hypoélastiques sont bien adaptées aux comportements métalliques, et pas aux comportements du béton).
- si le modèle HPP admet une expression énergétique, il en sera de même du modèle grandes déformations : la matrice tangente est donc symétrique, ce qui n'était pas le cas de GDEF\_HYPO\_ELAS ;
- la seule difficulté semble *a priori* concentrée dans la définition de la déformation logarithmique, mais l'article [13] fournit un algorithme de calcul distinguant les cas difficiles (entités propres multiples) ;
- le modèle, d'après les exemples présentés par les auteurs, donnent des résultats très proches de ceux obtenus par un formalisme classique à décomposition multiplicative ;

- le modèle peut être étendu aux cas d'anisotropie (initiale ou induite), ce qui peut être une source importante de motivation (notamment pour le comportement des alliages à base de Zirconium).

De plus, l'article [13] fournit une expression de la matrice tangente dans la configuration utilisant  $\mathbf{II}$  (appelée nominale) ; toutefois, comme elle est basée sur une écriture à partir des contraintes de Piola-Kirchhoff de première espèce, non symétrique, ce qui n'est pas classique et jamais réalisé dans Code\_Aster, on préfère ici utiliser le second tenseur de Piola-Kirchhoff pour calculer les forces internes et la matrice tangente sur la configuration initiale, en se référant à [15] par exemple.

## 4.2 Algorithme

**Pré-traitement :**

$$\mathbf{E}_{n+1} = \frac{1}{2} \log \left[ \mathbf{F}_{n+1}^T \cdot \mathbf{F}_{n+1} \right] = \frac{1}{2} \log \left[ \mathbf{C}_{n+1} \right] \text{ calculé par décomposition spectrale :}$$

Si  $\lambda^{(i)}$  sont les valeurs propres de  $\mathbf{C}_{n+1}$  et  $\mathbf{N}^{(i)}$  les vecteurs propres associés, alors la mesure de déformation choisie s'écrit  $\mathbf{E}_{n+1} = \frac{1}{2} \sum_{i=1,3} \log(\lambda^{(i)}) \mathbf{N}^{(i)} \otimes \mathbf{N}^{(i)}$ .

Cette mesure permet d'obtenir une décomposition additive :  $\mathbf{E} = \mathbf{E}^e + \mathbf{E}^p$ , avec  $\mathbf{E}^p = \frac{1}{2} \log(\mathbf{F}^{pT} \cdot \mathbf{F}^p)$  et  $\mathbf{E}^e = \frac{1}{2} \log(\mathbf{F}^{eT} \cdot \mathbf{F}^e)$

De plus on peut écrire également, entre l'instant  $n$  et l'instant  $n+1$  :

$$\mathbf{E}_{n+1} = \mathbf{E}_n + \Delta \mathbf{E}$$

**On appelle alors la loi de comportement HPP**

Elle fournit le tenseur des contraintes  $\mathbf{T}$ , défini par  $\mathbf{T}_{n+1} = \hat{\mathbf{T}}(\Delta \mathbf{E}; \mathbf{E}_n, \mathbf{T}_n, \beta_n)$  où  $\beta_n$  représente l'ensemble des variables internes à l'instant  $n$  et  $\mathbf{T}_n$  les contraintes à l'instant  $n$

Il faut donc recalculer  $\mathbf{T}_n$  en fonction des contraintes de Cauchy  $\sigma_n$  stockées à l'instant  $n$  (contraintes qui s'écrivent :  $\mathbf{T}_n = \mathbf{S}_n : \mathbf{P}_n^{-1}$ . Toutefois cela nécessite la transformation de  $\sigma_n$  en  $\mathbf{S}_n$  et le calcul de  $\mathbf{P}_n^{-1}$  qui peut être coûteux. On choisit donc de stocker les tenseurs  $\mathbf{T}$  en tant que variables internes.

**Post-traitement :**

Le tenseur des contraintes de Piola-Kirchhoff de 2ème espèce s'obtient par :

$\mathbf{S}_{n+1} = \mathbf{T}_{n+1} : \mathbf{P}_{n+1}$ , avec  $\mathbf{P}_{n+1} = \partial_C(\mathbf{E}_{n+1})$  ; cette quantité est calculée via un algorithme présenté dans [13] et rappelé en annexe 2.

Le tenseur des contraintes de Cauchy est obtenu par  $\mathbf{J} \boldsymbol{\sigma} = \boldsymbol{\tau} = \mathbf{F} \cdot \mathbf{S} \cdot \mathbf{F}^T = \mathbf{F} \cdot (\mathbf{T} : \mathbf{P}) \cdot \mathbf{F}^T$ .

On obtient le module tangent, en configuration dite «Lagrangienne», par dérivation de

$$\mathbf{S}_{n+1} = \mathbf{T}_{n+1} : \mathbf{P}_{n+1} : \dot{\mathbf{S}}_{n+1} = \mathbf{C}_{n+1}^{ep} : \frac{1}{2} \dot{\mathbf{C}}_{n+1} \text{ avec } \mathbf{C}_{n+1}^{ep} = \mathbf{P}_{n+1}^T : \mathbf{E}_{n+1}^{ep} : \mathbf{P}_{n+1} + \mathbf{T}_{n+1} : \mathbf{L}_{n+1} \text{ où}$$

$$\mathbf{E}_{n+1}^{ep} = \frac{\partial \mathbf{T}_{n+1}}{\partial \mathbf{E}_{n+1}} \text{ représente l'opérateur tangent issu de la loi de comportement et } \mathbf{L}_{n+1} \text{ est le}$$

tenseur d'ordre 6 (en 3D) défini par  $\mathbf{L}_{n+1} = 4 \partial_{CC}^2(\mathbf{E}_{n+1})$

On peut alors calculer sur cette configuration les forces internes et la matrice tangente, sur la configuration initiale, comme dans [14].

Le formalisme est testé en particulier sur le cas test de traction-rotation pour des lois élastoplastiques de Von Mises avec écrouissage isotrope ou cinématique (cf. tests SSND106, SSND107).

Comme le montrent les figures ci-dessous, l'objectivité est préservée (contraintes invariantes par rotation en écrouissages isotrope et cinématique) et la précision bonne (résultats identiques à SIMO\_MIEHE en isotrope).

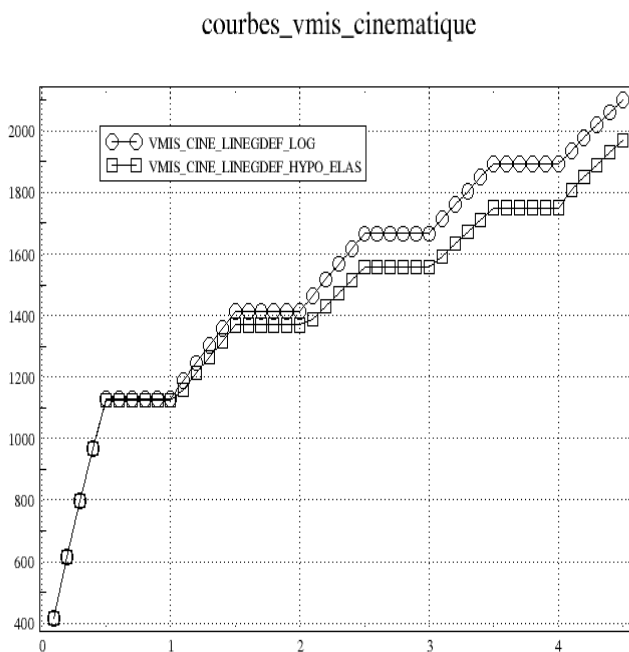
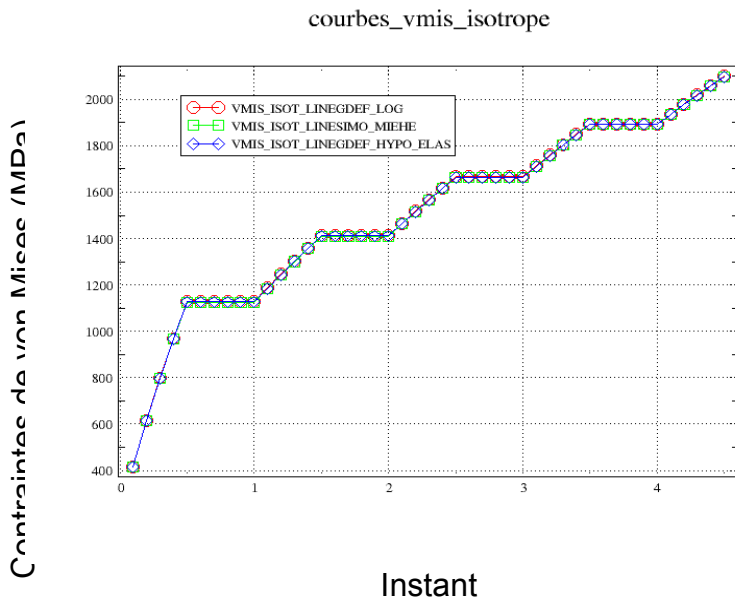


Figure 4 : Réponses pour le modèle GDEF\_LOG en traction - rotations

## 5 Validité des modèles de grandes déformations

### 5.1 Identification des paramètres

Il faut également préciser que le formalisme ici présenté et étudié n'étend pas la validité des lois de comportement dans le domaine des grandes déformations, il ne fait qu'en proposer une dérivation objective. Pour clarifier ce propos, il est possible de considérer le cas d'une loi élastoplastique à écrouissage isotrope linéaire. Ce type de loi de plasticité est valide physiquement en petites déformations ; son utilisation est étendue aux grandes déformations, mais sa validité physique peut être justement remise en question.

De plus, une identification faite sur des essais en petites déformations doit être potentiellement reconsidérée ; sur la figure 2, on présente une courbe de traction modélisée par un écrouissage isotrope linéaire : le module tangent doit forcément être défini par rapport à la plage de déformation considérée. En petites déformations, il semble plus judicieux d'utiliser  $E_{T1}$  ; si les déformations sont plus importantes, il semble plus judicieux d'utiliser  $E_{T2}$  comme valeur de la pente d'écrouissage. Mais on sent bien que c'est la validité physique de la loi elle-même qui devrait être remise en cause.

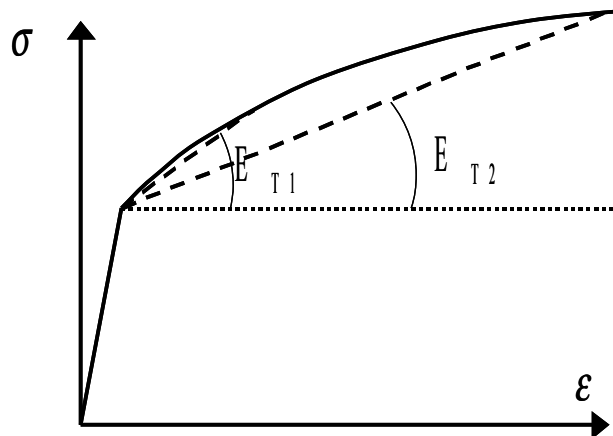


Figure 2 : Identification en grandes ou petites déformations

## 6 Comparaison avec PETIT\_REAC

### 6.1 Approximation des grandes déformations par PETIT\_REAC

Le principe de la formulation PETIT\_REAC consiste simplement à réactualiser la géométrie du problème au cours des itérations de Newton (et non à la fin de chaque pas de temps). Ceci signifie que toutes les quantités intervenant dans les équations du problème sont évaluées sur la configuration actuelle. Rien d'autre n'est modifié par rapport au cas petites perturbations.

#### 6.1.1 Description cinématique

La description cinématique est la même que celle des petites perturbations. Ceci signifie qu'un incrément de déformation est calculé par :

$$\Delta \varepsilon = \frac{1}{2} (\nabla_{\Omega_i}(\Delta u) + \nabla_{\Omega_i}^T(\Delta u))$$

Tableau 1

$\Omega_i$  étant la configuration réactualisée. La déformation totale est ensuite la somme de chacun de ces incréments de déformation linéarisée, calculés sur des configurations différentes. Il est donc délicat de lui donner un sens physique et mieux vaut l'utiliser comme un indicateur du niveau de déformation atteint. On applique l'hypothèse de décomposition additive des déformations.

### 6.1.2 Relation de comportement élastoplastique

Dans l'expression de la relation contraintes-déformations élastiques, on a vu (eq.7) la nécessité d'utiliser une dérivée objective :  $\dot{\sigma} = C : [\dot{\varepsilon} - \dot{\varepsilon}^p]$ . Avec PETIT\_REAC on remplace la dérivée objective par la simple dérivée en temps : elle n'est donc pas objective. En conséquence, l'emploi de PETIT\_REAC n'est donc pas approprié aux grandes rotations mais il l'est aux grandes déformations, sous certaines conditions [10] :

- très petits incréments,
- très petites rotations (ce qui implique un chargement quasi-radial)
- déformation élastiques petites devant les déformations plastiques,
- comportement isotrope.

### 6.1.3 Équilibre et matrice tangente

En terme d'éléments finis, la résolution par PETIT\_REAC implique à chaque pas de charge la résolution du même système non linéaire qu'en petites déformations [11]:

$$\begin{aligned} L^{\text{int}}(u_i, t_i) + B^T \cdot \lambda_i &= L^{\text{ext}}(t_i) \\ B \cdot u_i &= u^d(t_i) \end{aligned}$$

Tableau 6.1.3-1

A la différence près que les forces internes sont formellement calculées par :

$$L^{\text{int}}(u_i, t_i) = Q^T(u_i) : \sigma$$

Tableau 6.1.3-2

où l'opérateur  $Q$  dépend des déplacements. Dans ce cadre, le calcul de la matrice tangente mène à :

$$K_i^n = \frac{\partial L^{\text{int}}}{\partial u} \Big|_{(u_i^n, t_i)} = Q(u) : \frac{\partial \sigma}{\partial u} \Big|_{(u_i^n, t_i)} + \frac{\partial Q(u)}{\partial u} \Big|_{(u_i^n, t_i)} : \sigma$$

Tableau 6.1.3-3

Le premier terme est la contribution du comportement, similaire à ce qui a été présenté en petites transformations, à la différence que cette contribution est ici évaluée en configuration actuelle. Le second terme est la contribution de la géométrie qui n'est pas présente en petites transformations. Dans le cadre de la résolution PETIT\_REAC, ce terme n'est pas présent dans le calcul de la matrice tangente. On a donc :



$$K_i^n = Q(u) : \frac{\partial \sigma}{\partial u} \Big|_{(u_i^n, t_i)}$$

Tableau 6.1.3-4

L'absence de la contribution géométrique dans la matrice tangente peut parfois rendre la convergence difficile.

## 6.2 Comparaison sur un exemple

Le formalisme `PETIT_REAC` (cf. [6]) se base sur une actualisation de la géométrie pour le calcul de l'incrément de déformation avant d'intégrer le comportement de façon identique aux petites déformations. Ceci permet un traitement simple des grandes déformations, mais de façon très approchée, non objective et pouvant engendrer de grandes erreurs.

Pour s'en convaincre, considérons la traction-rotation alternée d'un cube ; pour plus de détails sur le test, on se référera à [7] par exemple.

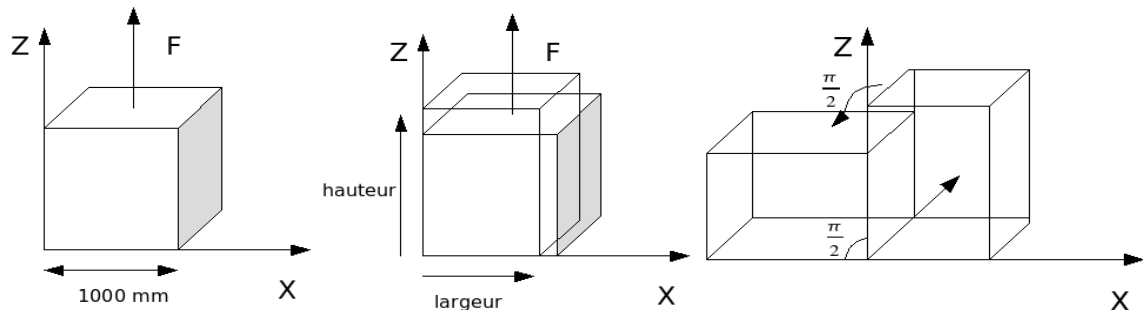


Figure 3 : Exemple de traction rotation d'un cube

Durant les phases de rotation, la contrainte doit rester constante : un mouvement de corps rigide n'engendre pas de contraintes (en statique tout du moins et sans viscosité).

La figure ci-dessous présente la réponse obtenue avec un comportement `VMIS_ISOT_LINE` pour les déformations `PETIT_REAC` et `GDEF_HYPO_ELAS`. Le type de déformation `PETIT_REAC` est mis en défaut alors que `GDEF_HYPO_ELAS` est valide (et fournit une réponse identique à `SIMO_MIEHE`).

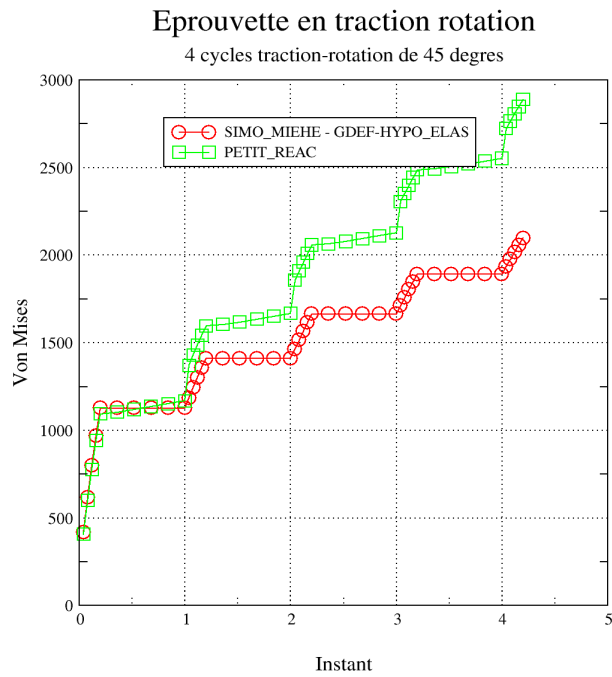


Figure 4 : Contrainte de von Mises en traction-rotation

## 7 Bibliographie

- 1 Simo J.C., Hughes T.J.R., Computational Inelasticity, Mechanics and Materials, Interdisciplinary Applied Mathematics Vol. 7 , Edition Springer-Verlag, New York, 1998.
- 2 Manuel utilisation de Aster U4.51.11, Comportement Non Linéaires
- 3 Manuel de référence Aster R5.03.21, Modélisation élasto-(visco)-plastique avec écrouissage isotrope en grandes déformations,
- 4 Manuel de référence Aster R5.03.19, Loi de comportement hyperélastique. Matériau presque incompressible.
- 5 Note interne EDF HI-74/98/006/0, Introduction dans le Code\_Aster d'un modèle de comportement en grandes déformations élastoplastique avec écrouissage isotrope, V. Cano E . Lorentz, 1998.
- 6 Manuel utilisation Aster U4.51.11, Comportements Non Linéaires.
- 7 Manuel validation Aster ssna106, Traction-rotation alternée d'un élément. Loi plastique isotrope.
- 8 Grandes transformations, contraintes planes, déformations planes généralisées, Leblond, 1989.
- 9 Manuel de références Aster R5.03.40, Modélisation statique et dynamique des poutres en grandes rotations.
- 10 Compte-rendu EDF/DER-MMN 1536/07 : Grandes déformations plastiques. Modélisation dans Aster par PETIT\_REAC. E.Lorentz, 24 juin 1996
- 11 Manuel de références Aster R5.03.01, Algorithme de STAT\_NON\_LINE
- 12 CR-AMA-09.201. « Modèles de grandes déformations compatibles avec la plupart des comportements ». Rapport de stage de Julie Coloigner. R. Bargellini, J.-M. Proix.
- 13 Miehe C., Apel N., Lambrecht M.: Anisotropic additive plasticity in the logarithm strain space : modular kinematic formulation and implementation based on incremental minimization principles for standard materials., Computer Methods in Applied Mechanics and Engineering, 191, pp.5383-5425, 2002.
- 14 Manuel de référence Aster R5.03.20, relation de comportement élastique non linéaire en grands déplacements.
- 15 CR-AMA-11-007. « Amélioration de la matrice tangente du formalisme de grandes déformations GDEF\_HYPO\_ELAS ». R. Bargellini, J.-M. Proix.
- 16 « Computational methods for plasticity » Wiley 2008, de EA de Souza Neto, D. Peric, DRJ. Owen

## 8 Description des versions du document

Version Aster	Auteur(s) Organisme(s)	Description des modifications
10.2	R.BARGELLINI, J.M.PROIX R&D/AMA J.COLOIGNER Univ. Rennes	Texte initial
10.4	R.BARGELLINI, J.M.PROIX R&D/AMA	Ajout de la description de l'algorithme utilisant la dérivée de Lie, du calcul de l'opérateur tangent, et de l'algorithme GDEF_LOG

Tableau 8-1

## 9 Annexe 1 : opérateur tangent pour GDEF\_HYPO\_ELAS

Dans la référence [1], deux algorithmes sont proposés pour résoudre les équations de l'élasto-plasticité avec une formulation hypo-élastique : celui présenté dans ce document (GDEF\_HYPO\_ELAS), et un algorithme plus simple, utilisant une approximation de la dérivée de Lie, que nous décrivons ici.

<p style="text-align: center;"><b>GDEF_HYPO_ELAS</b> (dérivée de Jaumann)</p>	<p style="text-align: center;"><b>SIMO_HUGHES_1</b> (dérivée de Lie)</p>
<p>Connaissant les champs de déplacements, on calcule les grandeurs suivantes :</p> $\mathbf{f}_{n+1} = \mathbf{I}_d + \mathbf{grad}_n \mathbf{u}, \quad \mathbf{f}_{n+\alpha} = \mathbf{I}_d + \alpha \mathbf{grad}_n \mathbf{u} = (1-\alpha)\mathbf{I}_d + \alpha \mathbf{f}_{n+1}, \quad \tilde{\mathbf{f}}_{n+\alpha} = \mathbf{f}_{n+1} \cdot \mathbf{f}_{n+\alpha}^{-1}$	
<p><b>Définition des tenseurs de rotations</b></p> <p>1/ gradient incrémental de déplacement</p> $\mathbf{h}_{n+1/2} = \mathbf{grad}_n \mathbf{u} \mathbf{f}_{n+1/2}^{-1} = \mathbf{I}_d - \mathbf{f}_{n+1/2}^{-1}$ <p>2/ Taux de rotation <math>\omega</math> et tenseur de rotation <math>\Theta</math></p> $\Theta = \Delta t \omega_{n+1/2} = \frac{1}{2} [\mathbf{h}_{n+1/2} - \mathbf{h}_{n+1/2}^T] = \mathbf{f}_{n+1/2}^{-T} - \mathbf{f}_{n+1/2}^{-1}$ <p>3/Tenseur orthogonal propre <math>\Lambda</math> et de rotation <math>\mathbf{r}</math></p> $\Lambda_{n+1} = \exp[\Theta] \Lambda_n, \quad \Lambda_{n+1/2} = \exp[\Theta/2] \Lambda_n$ $\mathbf{r}_{n+1} = \Lambda_{n+1} \Lambda_n^T, \quad \mathbf{r}_{n+1/2} = \Lambda_{n+1} \Lambda_{n+1/2}^T$	<p style="text-align: center;"><b>Rien</b></p>
<p><b>Définition des déformations :</b></p> $\mathbf{e}_{n+1} = \frac{1}{2} [\mathbf{I}_d - (\mathbf{f}_{n+1} \mathbf{f}_{n+1}^T)^{-1}], \quad \tilde{\mathbf{e}}_{n+\alpha} = \tilde{\mathbf{f}}_{n+\alpha}^T \mathbf{e}_{n+1} \tilde{\mathbf{f}}_{n+\alpha}$	<p><b>Définition des déformations :</b></p> $\mathbf{e}_{n+1} = \frac{1}{2} [\mathbf{I}_d - (\mathbf{f}_{n+1} \mathbf{f}_{n+1}^T)^{-1}],$ $\tilde{\mathbf{e}}_{n+\alpha} = \tilde{\mathbf{f}}_{n+\alpha}^T \mathbf{e}_{n+1} \tilde{\mathbf{f}}_{n+\alpha}$
<p style="text-align: center;"><b>Données pour l'intégration du comportement</b></p> $\left\{ \begin{array}{ll} \sigma_n = \mathbf{r}_{n+1} \tau_n \mathbf{r}_{n+1}^T & \text{contrainte à l'instant } n \\ \varepsilon_n = [\mathbf{r}_{n+1/2} \tilde{\mathbf{e}}_{n+1/2} \mathbf{r}_{n+1/2}^T] & \text{déformations} \\ \mathbf{x}_n = \mathbf{r}_{n+1} \mathbf{q}_n \mathbf{r}_{n+1}^T & \text{variables cinématiques} \end{array} \right.$	
<p style="text-align: center;"><b>GDEF_HYPO_ELAS</b> (dérivée de Jaumann)</p>	

Tableau 9-1 : Boîte 2 – Deux algorithmes de résolution

Quelque soit la valeur de  $\alpha$ , Ces deux algorithmes sont incrémentalement objectifs : considérons une rotation pure :  $e_{n+1} = \frac{1}{2} [I_d - (f_{n+1} f_{n+1}^T)^{-1}] = 0$  alors  $\tau_{n+1} = r_{n+1} \tau_n r_{n+1}^T$

## 9.1 Principe du calcul de la matrice tangente : écriture variationnelle et expression générale

Ce paragraphe est générique pour tous les formalismes grandes déformations et lois de comportements. Il s'inspire notamment de [5] et permet de définir la matrice tangente, dont le calcul dans le cas des formalismes décrits avant sera ensuite détaillé.

Le cadre de l'analyse est évidemment celui des transformations finies. L'objectif est de résoudre le problème mécanique défini par l'équilibre. Ceci revient à déterminer le champ de déplacement admissible  $U$  qui annule la fonctionnelle des travaux virtuels  $G(U, V)$ , ce qui peut s'écrire, dans la configuration initiale :

$$\forall V \in V_{ad}^0, G(U, V) = \int_{\Omega_0} (P : \text{grad}_0 V - b V) d\Omega_0 - \int_{\partial\Omega_0} t \cdot V d\Omega_0 = 0$$

avec  $V_{ad}^0$  l'espace des déplacements admissibles,  $b$  et  $t$  les efforts volumiques et de surfaces et  $P$  le premier tenseur des contraintes de Piola-Kirchhoff.

Par comparaison, son équivalent en petites déformations est :

$$\forall V \in V_{ad}^0, G(U, V) = \int_{\Omega} (\sigma : \text{grad} V - b V) d\Omega - \int_{\partial\Omega} t \cdot V d\Omega$$

En particulier, la notion de configuration de domaine est en petites déformations sans objet, puisque toutes sont équivalentes.

La dépendance de  $G$  en  $U$  provient de la dépendance de  $P$  en  $U$  à travers le gradient de transformation  $F = I_d + \text{grad}_0 U$ .

Afin de résoudre le problème  $G(U, V) = 0 \forall V \in V_{ad}^0$ , on le linéarise par rapport à l'inconnue  $U$  autour d'un état arbitraire défini par  $U^*$ . Le problème linéarisé consiste alors à déterminer le champ  $\delta U$  qui annule la fonctionnelle linéarisée des travaux virtuels, c'est à dire :

$$\text{trouver } \delta U \text{ tel que } \forall V \in V_{ad}^0, L(\delta U, V) = G(U^*, V) + DG(U^*, V)[\delta U] = 0$$

avec  $DG(U^*, V)[\delta U]$  la dérivée directionnelle de  $G(U, V)$  au point  $U^*$  dans la direction  $\delta U$ .

On introduit pour résoudre le problème un déplacement infinitésimal  $\epsilon \delta U$  autour de  $U^*$  ; le gradient de transformation s'écrit alors comme une fonction de  $\epsilon$  telle que  $F(\epsilon) = F(U^*) + \epsilon \delta U$ , et la dérivée directionnelle de  $G(U, V)$  est définie par :

$$\begin{aligned} DG(U^*, V)[\delta U] &= \frac{d}{d\epsilon} \left[ \int_{\Omega_0} (P(F(\epsilon)) : \text{grad}_0 V - b V) d\Omega_0 - \int_{\partial\Omega_0} t \cdot V d\Omega_0 \right] (\epsilon=0) \\ &= \frac{d}{d\epsilon} \left[ \int_{\Omega_0} (P(F(\epsilon)) : \text{grad}_0 V) d\Omega_0 \right] (\epsilon=0) \end{aligned}$$

En appliquant simplement les règles de dérivée composée, on peut exprimer cette dérivée directionnelle comme suit :

$$DG(U^*, V)[\delta U] = \int_{\Omega_0} A : \text{grad}_0 \delta U : \text{grad}_0 V d\Omega_0$$

avec  $A = \frac{\partial P}{\partial F}(F(U^*))$  la matrice tangente matérielle.

Ceci permet alors de définir l'équation linéarisée des travaux virtuels dans la configuration matérielle :

$$\text{trouver } \delta U \text{ tel que } \forall V \in V_{ad}^0, \int_{\Omega_0} A : \text{grad}_0 \delta U : \text{grad}_0 V d\Omega_0 = - \int_{\Omega_0} (P : \text{grad}_0 V - b V) d\Omega_0 + \int_{\partial\Omega_0} t \cdot V d\Omega_0$$

Pour des raisons de simplicité, il est préférable de rester en configuration initiale pour la définition des intégrales (notamment pour la définition des forces volumiques et surfaciques) et de calculer les gradients dans la configuration actuelle ; c'est d'ailleurs le choix qui avait été retenu pour l'intégration du formalisme de Simo-Miehe (voir [6]). On se sert donc de la relation  $\mathbf{grad}_0 \mathbf{a} = \mathbf{grad}_n \mathbf{a} \cdot \mathbf{F}$  (avec  $\mathbf{grad}_n$  le gradient par rapport à la configuration actuelle) pour écrire la dérivée directionnelle comme suit :

$$DG(\mathbf{U}^*, \mathbf{V})[\delta \mathbf{U}] = \int_{\Omega_0} \mathbf{A} : \left[ (\mathbf{grad}_n \delta \mathbf{U}) \cdot \mathbf{F} \right] : \left[ (\mathbf{grad}_n \mathbf{V}) \cdot \mathbf{F} \right] d\Omega_0$$

De plus, l'utilisation de la première contrainte de Piola-Kirchhoff, non symétrique, est problématique pour la définition des forces intérieures et des matrices tangentes ; on se sert donc de sa définition  $\mathbf{P} = J \boldsymbol{\sigma} \mathbf{F}^{-T} = \boldsymbol{\tau} \mathbf{F}^{-T}$  pour exprimer  $\mathbf{A}$ , en notation indicielle:

$$A_{ijkl} = \frac{\partial P_{ij}}{\partial F_{kl}} = \frac{\partial \tau_{ip} F_{jp}^{-1}}{\partial F_{kl}} = \frac{\partial \tau_{ip}}{\partial F_{kl}} F_{jp}^{-1} - \tau_{ip} F_{jk}^{-1} F_{lp}^{-1}$$

car pour un tenseur d'ordre 2  $\mathbf{X}$ ,  $\frac{\partial X_{jp}^{-1}}{\partial X_{kl}} = -X_{jk}^{-1} X_{lp}^{-1}$ .

D'où, :

$$\begin{aligned} DG(\mathbf{U}^*, \mathbf{V})[\delta \mathbf{U}] &= \int_{\Omega_0} A_{ijkl} \left[ (\mathbf{grad}_n \delta \mathbf{U})_{kr} F_{rl} \right] \left[ (\mathbf{grad}_n \mathbf{V})_{is} F_{sj} \right] d\Omega_0 \\ &= \int_{\Omega_0} \left[ \frac{\partial \tau_{ip}}{\partial F_{kl}} F_{jp}^{-1} - \tau_{ip} F_{jk}^{-1} F_{lp}^{-1} \right] \left[ (\mathbf{grad}_n \delta \mathbf{U})_{kr} F_{rl} \right] \left[ (\mathbf{grad}_n \mathbf{V})_{is} F_{sj} \right] d\Omega_0 \\ &= \int_{\Omega_0} \left[ \frac{\partial \tau_{ip}}{\partial F_{kl}} F_{jp}^{-1} F_{rl} F_{sj} - \tau_{ip} F_{jk}^{-1} F_{lp}^{-1} F_{rl} F_{sj} \right] \left[ (\mathbf{grad}_n \delta \mathbf{U})_{kr} \right] \left[ (\mathbf{grad}_n \mathbf{V})_{is} \right] d\Omega_0 \\ &= \int_{\Omega_0} \left[ \frac{\partial \tau_{is}}{\partial F_{kl}} F_{rl} - \tau_{ir} \delta_{ks} \right] \left[ (\mathbf{grad}_n \delta \mathbf{U})_{kr} \right] \left[ (\mathbf{grad}_n \mathbf{V})_{is} \right] d\Omega_0 \\ &= \int_{\Omega_0} \mathbf{C} : (\mathbf{grad}_n \delta \mathbf{U}) : (\mathbf{grad}_n \mathbf{V}) d\Omega_0 \end{aligned}$$

avec

$$C_{iskr} = \frac{\partial \tau_{is}}{\partial F_{kl}} F_{rl} - \tau_{ir} \delta_{ks}$$

la matrice tangente du problème. Cette matrice tangente est composée de deux termes distincts. Le second est appelé « rigidité géométrique », et le premier est lié au formalisme de grandes déformations utilisé, et éventuellement à la loi de comportement : c'est lui qu'il faut ici calculer. On notera que cette expression est en accord avec [6] ; d'un point de vue intégration de code, cela permet de ne pas avoir à modifier trop fondamentalement celui-ci dans la mesure où il sera possible d'utiliser les routines dédiées à Simo-Miehe.

On rappelle qu'en petites déformations, la matrice tangente s'écrit simplement  $C_{iskr}^{HPP} = \frac{\partial \sigma_{is}}{\partial \varepsilon_{kr}}$ .

## 9.2 Calcul de la matrice tangente pour les deux formalismes

Comme annoncé en introduction, ce paragraphe est volontairement très détaillé afin qu'un lecteur attentif puisse corriger les éventuelles erreurs de raisonnement et/ou de calcul.

Le paragraphe précédent aboutit à la conclusion que le terme à calculer est  $\frac{\partial \boldsymbol{\tau}}{\partial \mathbf{F}} \cdot \mathbf{F}^T$ . Si on discrétise temporellement les configurations comme présenté en Figure 1, il s'écrit indifféremment :

$$\frac{\partial \boldsymbol{\tau}}{\partial \mathbf{F}_{n+1}} \cdot \mathbf{F}_{n+1}^T = \frac{\partial \boldsymbol{\tau}}{\partial \mathbf{f}_{n+1}} \cdot \mathbf{f}_{n+1}^T$$

Il faut donc principalement évaluer le calcul de la dérivée de la contrainte de Kirchhoff.

Les formalismes présentés et considérés ici utilisent les lois de comportement élasto-plastiques écrites en petites déformations. Dans ce cadre, on a toujours  $\boldsymbol{\tau}_{n+1} = \mathbf{g}(\boldsymbol{\sigma}_n, \boldsymbol{\varepsilon}_n, \mathbf{x}_n; \beta_n)$ , où  $\beta_n$  représente l'ensemble

des variables internes scalaires d'écouissage isotrope et  $\mathbf{x}_n$  les variables tensorielles d'écouissage cinématique ; l'expression du comportement tangent est alors cherchée sous la forme :

$$\frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \mathbf{f}_{n+1}} = \frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \boldsymbol{\varepsilon}_n} \cdot \frac{\partial \boldsymbol{\varepsilon}_n}{\partial \mathbf{f}_{n+1}} + \frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \boldsymbol{\sigma}_n} \cdot \frac{\partial \boldsymbol{\sigma}_n}{\partial \mathbf{f}_{n+1}} + \frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \mathbf{x}_n} \cdot \frac{\partial \mathbf{x}_n}{\partial \mathbf{f}_{n+1}}$$

Pour les deux algorithmes considérés ici, il faut donc évaluer chacun des termes :

1.  $\frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \boldsymbol{\varepsilon}_n}$  : fourni par la loi de comportement appelée en petites déformations.
2.  $\frac{\partial \boldsymbol{\varepsilon}_n}{\partial \mathbf{f}_{n+1}}$  : spécifique à chaque algorithme, il contient toutefois un terme commun :  $\frac{\partial \tilde{\boldsymbol{\varepsilon}}_{n+\alpha}}{\partial \mathbf{f}_{n+1}}$
3.  $\frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \boldsymbol{\sigma}_n}$  : non calculé actuellement par les lois de comportement. En élasticité, il vaut l'identité. En plasticité, son expression exacte est spécifique à chaque loi de comportement.
4.  $\frac{\partial \boldsymbol{\sigma}_n}{\partial \mathbf{f}_{n+1}}$  : spécifique à chaque algorithme.

Et dans le cas d'un écouissage cinématique :

5.  $\frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \mathbf{x}_n}$  : là encore spécifique à chaque loi de comportement,
6.  $\frac{\partial \mathbf{x}_n}{\partial \mathbf{f}_{n+1}}$  : spécifique à chaque algorithme.

Deux aspects sont à mettre en avant dès à présent. Tout d'abord le sentiment que le calcul de la matrice tangente va être long, avec un grand nombre de termes ; d'un point de vue numérique, ceci risque d'avoir un coût conséquent. De plus, la présence de deux termes spécifiques à la loi de comportement appelée par les formalismes : on perd là d'avance le caractère générique des formalismes, ce qui était un de leur point fort pour le calcul lui-même ; en outre, on comprend pourquoi les auteurs eux-mêmes avancent que « la notion de module tangent consistant n'est pas disponible pour la classe de formalismes » considérée ici (voir [1], Remark 8.3.2, pp 292)

La suite présente successivement le calcul de chacun des termes.

**Terme 1**  $\frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \boldsymbol{\varepsilon}_n}$

Ce terme est directement issu de la loi de comportement en petites déformations ; aucun calcul n'est donc nécessaire.

**Terme 2**  $\frac{\partial \boldsymbol{\varepsilon}_n}{\partial \mathbf{f}_{n+1}}$

Ce terme, comme précisé avant, dépend du formalisme considéré. On verra toutefois qu'il contient des termes communs.

## 9.2.1 Pour l'algorithme SIMO\_HUGHES\_1

Pour cet algorithme, on a  $\boldsymbol{\varepsilon}_n = \tilde{\boldsymbol{\varepsilon}}_{n+\alpha} = \tilde{\mathbf{f}}_{n+\alpha}^T \mathbf{e}_{n+1} \tilde{\mathbf{f}}_{n+\alpha}$ .

D'où

$$\frac{\partial \boldsymbol{\varepsilon}_n}{\partial \mathbf{f}_{n+1}} = \frac{\partial \tilde{\boldsymbol{\varepsilon}}_{n+\alpha}}{\partial \mathbf{f}_{n+1}} = \frac{\partial \tilde{\mathbf{f}}_{n+\alpha}^T}{\partial \mathbf{f}_{n+1}} \mathbf{e}_{n+1} \tilde{\mathbf{f}}_{n+\alpha} + \tilde{\mathbf{f}}_{n+\alpha}^T \frac{\partial \mathbf{e}_{n+1}}{\partial \mathbf{f}_{n+1}} \tilde{\mathbf{f}}_{n+\alpha} + \tilde{\mathbf{f}}_{n+\alpha}^T \mathbf{e}_{n+1} \frac{\partial \tilde{\mathbf{f}}_{n+\alpha}}{\partial \mathbf{f}_{n+1}}$$

Deux dérivées sont donc à calculer ici, dont l'une est nécessaire deux fois ; nous allons les expliciter.

$$\begin{aligned} \bullet \frac{\partial (\mathbf{e}_{n+1})_{ij}}{\partial (\mathbf{f}_{n+1})_{kl}} &= -\frac{1}{2} \tilde{\mathbf{f}}_{n+\alpha}^T \frac{\partial (\mathbf{f}_{n+1}^{-T} \mathbf{f}_{n+1}^{-1})}{\partial \mathbf{f}_{n+1}} \tilde{\mathbf{f}}_{n+\alpha} \\ &= \frac{1}{2} [(\mathbf{f}_{n+1}^{-1})_{li} (\mathbf{f}_{n+1}^{-1})_{mj} (\mathbf{f}_{n+1}^{-1})_{mk} + (\mathbf{f}_{n+1}^{-1})_{lj} (\mathbf{f}_{n+1}^{-1})_{mi} (\mathbf{f}_{n+1}^{-1})_{mk}] \\ &= \frac{1}{2} [(\mathbf{f}_{n+1}^{-1})_{li} (\mathbf{f}_{n+1}^{-1})_{mj} + (\mathbf{f}_{n+1}^{-1})_{lj} (\mathbf{f}_{n+1}^{-1})_{mi}] (\mathbf{f}_{n+1}^{-1})_{mk} \\ \bullet \frac{\partial (\tilde{\mathbf{f}}_{n+\alpha})_{ij}}{\partial (\mathbf{f}_{n+1})_{kl}} &= \frac{\partial [(\mathbf{f}_{n+1})_{ip} (\mathbf{f}_{n+\alpha}^{-1})_{pj}]}{\partial (\mathbf{f}_{n+1})_{kl}} = \delta_{ik} \delta_{pl} (\mathbf{f}_{n+\alpha}^{-1})_{pj} + (\mathbf{f}_{n+1})_{ip} \frac{\partial (\mathbf{f}_{n+\alpha}^{-1})_{pj}}{\partial (\mathbf{f}_{n+1})_{kl}}, \\ &= \delta_{ik} \delta_{pl} (\mathbf{f}_{n+\alpha}^{-1})_{pj} + (\mathbf{f}_{n+1})_{ip} \frac{\partial (\mathbf{f}_{n+\alpha}^{-1})_{pj}}{\partial (\mathbf{f}_{n+\alpha})_{rs}} \frac{\partial (\mathbf{f}_{n+\alpha})_{rs}}{\partial (\mathbf{f}_{n+1})_{kl}} \\ &= \delta_{ik} (\mathbf{f}_{n+\alpha}^{-1})_{lj} - (\mathbf{f}_{n+1})_{ip} (\mathbf{f}_{n+\alpha}^{-1})_{pr} (\mathbf{f}_{n+\alpha}^{-1})_{sj} \alpha \delta_{rk} \delta_{sl} \\ &= [\delta_{ik} - \alpha (\mathbf{f}_{n+1})_{ip} (\mathbf{f}_{n+\alpha}^{-1})_{pk}] (\mathbf{f}_{n+\alpha}^{-1})_{lj} \\ \text{car } \mathbf{f}_{n+\alpha} &= (1-\alpha) \mathbf{I}_d + \alpha \mathbf{f}_{n+1} \end{aligned}$$

En récapitulant ces deux résultats, on peut donc écrire le premier terme (la dérivée de la déformation étant calculée dans une routine dédiée, on la laisse dans les expressions sans l'expliquer) :

$$\begin{aligned} \frac{\partial (\boldsymbol{\varepsilon}_n)_{ij}}{\partial (\mathbf{f}_{n+1})_{kl}} &= \frac{\partial ((\tilde{\mathbf{f}}_{n+\alpha}^T)_{ip} (\mathbf{e}_{n+1})_{pq} (\tilde{\mathbf{f}}_{n+\alpha})_{qj})}{\partial (\mathbf{f}_{n+1})_{kl}} \\ &= \frac{\partial (\tilde{\mathbf{f}}_{n+\alpha})_{pi}}{\partial (\mathbf{f}_{n+1})_{kl}} (\mathbf{e}_{n+1})_{pq} (\tilde{\mathbf{f}}_{n+\alpha})_{qj} + (\tilde{\mathbf{f}}_{n+\alpha})_{pi} \frac{\partial (\mathbf{e}_{n+1})_{pq}}{\partial (\mathbf{f}_{n+1})_{kl}} (\tilde{\mathbf{f}}_{n+\alpha})_{qj} + (\tilde{\mathbf{f}}_{n+\alpha})_{pi} (\mathbf{e}_{n+1})_{pq} \frac{\partial (\tilde{\mathbf{f}}_{n+\alpha})_{qj}}{\partial (\mathbf{f}_{n+1})_{kl}} \\ &= (\tilde{\mathbf{f}}_{n+\alpha})_{pi} \frac{\partial (\mathbf{e}_{n+1})_{pq}}{\partial (\mathbf{f}_{n+1})_{kl}} (\tilde{\mathbf{f}}_{n+\alpha})_{qj} + [\delta_{ik} - \alpha (\mathbf{f}_{n+1})_{ir} (\mathbf{f}_{n+\alpha}^{-1})_{rk}] (\mathbf{e}_{n+1})_{st} [(\mathbf{f}_{n+\alpha}^{-1})_{li} (\tilde{\mathbf{f}}_{n+\alpha})_{sj} + (\tilde{\mathbf{f}}_{n+\alpha})_{si} (\mathbf{f}_{n+\alpha}^{-1})_{lj}] \end{aligned}$$

## 9.2.2 Pour l'algorithme GDEF\_HYPO\_ELAS

Pour cet algorithme, on prend toujours  $\alpha = \frac{1}{2}$  et la déformation s'écrit :  $\boldsymbol{\varepsilon}_n = \left[ \mathbf{r}_{n+\frac{1}{2}} \tilde{\boldsymbol{\varepsilon}}_{n+\frac{1}{2}} \mathbf{r}_{n+\frac{1}{2}}^T \right]$

$$\text{D'où } \frac{\partial (\boldsymbol{\varepsilon}_n)_{qp}}{\partial (\mathbf{f}_{n+1})_{kl}} = \frac{\partial (\mathbf{r}_{n+1/2})_{qa}}{\partial (\mathbf{f}_{n+1})_{kl}} (\tilde{\boldsymbol{\varepsilon}}_{n+1/2})_{ab} (\mathbf{r}_{n+1/2})_{pb} + \frac{\partial (\tilde{\boldsymbol{\varepsilon}}_{n+1/2})_{ab}}{\partial (\mathbf{f}_{n+1})_{kl}} (\mathbf{r}_{n+1/2})_{qa} (\mathbf{r}_{n+1/2})_{pb} + (\mathbf{r}_{n+1/2})_{qa} (\tilde{\boldsymbol{\varepsilon}}_{n+1/2})_{ab} \frac{\partial (\mathbf{r}_{n+1/2})_{pb}}{\partial (\mathbf{f}_{n+1})_{kl}}$$

Par rapport à SIMO\_HUGHES\_1, ce terme est plus long car il fait intervenir les dérivées des tenseurs de rotation  $\mathbf{r}_{n+1/2} = \boldsymbol{\Lambda}_{n+1} \boldsymbol{\Lambda}_{n+1/2}^T$ , avec  $\boldsymbol{\Lambda}_{n+1} = \exp[\boldsymbol{\Theta}] \boldsymbol{\Lambda}_n$  et  $\boldsymbol{\Lambda}_{n+1/2} = \exp[\boldsymbol{\Theta}/2] \boldsymbol{\Lambda}_n$ .

On calcule donc :



$$\begin{aligned}
 \frac{\partial(\mathbf{r}_{n+1/2})_{qa}}{\partial(\mathbf{f}_{n+1})_{kl}} &= \frac{\partial(\Lambda_{n+1})_{qc}}{\partial(\mathbf{f}_{n+1})_{kl}} (\Lambda_{n+1/2})_{ca}^{-1} + (\Lambda_{n+1})_{qc} \frac{\partial(\Lambda_{n+1/2})_{ca}^{-1}}{\partial(\mathbf{f}_{n+1})_{kl}} \\
 &= \frac{\partial \exp(\Theta)_{qd}}{\partial(\mathbf{f}_{n+1})_{kl}} (\Lambda_n)_{dc} (\Lambda_{n+1/2})_{ca}^{-1} + (\Lambda_{n+1})_{qc} \frac{\partial(\Lambda_{n+1/2})_{ca}^{-1}}{\partial(\Lambda_{n+1/2})_{mn}} \frac{\partial(\Lambda_{n+1/2})_{mn}}{\partial(\mathbf{f}_{n+1})_{kl}} \\
 &= \frac{\partial \exp(\Theta)_{qd}}{\partial(\Theta)_{mn}} \frac{\partial(\Theta)_{mn}}{\partial(\mathbf{f}_{n+1/2})_{rs}} \frac{\partial(\mathbf{f}_{n+1/2})_{rs}}{\partial(\mathbf{f}_{n+1})_{kl}} (\Lambda_n)_{dc} (\Lambda_{n+1/2})_{ca}^{-1} \\
 &\quad - (\Lambda_{n+1})_{qc} (\Lambda_{n+1/2})_{cm}^{-1} (\Lambda_{n+1/2})_{na}^{-1} \frac{\partial \exp(\Theta/2)_{mw}}{\partial(\Theta/2)_{rs}} \frac{\partial(\Theta/2)_{rs}}{\partial(\mathbf{f}_{n+1/2})_{de}} \frac{\partial(\mathbf{f}_{n+1/2})_{de}}{\partial(\mathbf{f}_{n+1})_{kl}} (\Lambda_n)_{wn} \\
 &= \frac{1}{2} \left[ \frac{\partial \exp(\Theta)_{qd}}{\partial(\Theta)_{mn}} \frac{\partial(\Theta)_{mn}}{\partial(\mathbf{f}_{n+1/2})_{rs}} \delta_{rk} \delta_{sl} (\Lambda_n)_{dc} (\Lambda_{n+1/2})_{ca}^{-1} \right. \\
 &\quad \left. - (\Lambda_{n+1})_{qc} (\Lambda_{n+1/2})_{cm}^{-1} (\Lambda_{n+1/2})_{na}^{-1} \frac{\partial \exp(\Theta/2)_{mw}}{\partial(\Theta/2)_{rs}} \frac{\partial(\Theta/2)_{rs}}{\partial(\mathbf{f}_{n+1/2})_{de}} \delta_{dk} \delta_{el} (\Lambda_n)_{wn} \right] \\
 &= \frac{1}{2} \left[ \frac{\partial \exp(\Theta)_{qd}}{\partial(\Theta)_{mn}} \frac{\partial(\Theta)_{mn}}{\partial(\mathbf{f}_{n+1/2})_{kl}} (\Lambda_n)_{dc} (\Lambda_{n+1/2})_{ca}^{-1} \right. \\
 &\quad \left. - (\Lambda_{n+1})_{qc} (\Lambda_{n+1/2})_{cm}^{-1} (\Lambda_{n+1/2})_{na}^{-1} \frac{\partial \exp(\Theta/2)_{mw}}{\partial(\Theta/2)_{rs}} \frac{\partial(\Theta/2)_{rs}}{\partial(\mathbf{f}_{n+1/2})_{kl}} (\Lambda_n)_{wn} \right]
 \end{aligned}$$

Pour finir le calcul de ce terme, il manque encore deux dérivées.

- $(\Theta) = \mathbf{f}_{n+1/2}^{-T} - \mathbf{f}_{n+1/2}^{-1}$  donc

$$\begin{aligned}
 \frac{\partial(\Theta)_{mn}}{\partial(\mathbf{f}_{n+1/2})_{rs}} &= \frac{\partial(\Theta)_{mn}}{\partial(\mathbf{f}_{n+1/2})_{tu}} \frac{\partial(\mathbf{f}_{n+1/2})_{tu}}{\partial(\mathbf{f}_{n+1/2})_{rs}} = -(\delta_{nt} \delta_{mu} - \delta_{mt} \delta_{nu}) (\mathbf{f}_{n+1/2})_{tr}^{-1} (\mathbf{f}_{n+1/2})_{su}^{-1} \\
 &= (\mathbf{f}_{n+1/2})_{mr}^{-1} (\mathbf{f}_{n+1/2})_{sn}^{-1} - (\mathbf{f}_{n+1/2})_{nr}^{-1} (\mathbf{f}_{n+1/2})_{sm}^{-1}
 \end{aligned}$$

- la dérivée de l'exponentielle tensorielle. Celle-ci est présentée en Annexe ; on retiendra surtout ici qu'elle est itérative et demande un assez grand nombre de calculs.

Ce terme est donc d'une écriture complexe et forcément d'un coût assez élevé pour cet algorithme.

### Terme 3 $\frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \boldsymbol{\sigma}_n}$

Ce terme, commun aux deux formalismes, empêche la matrice tangente d'être d'expression exacte unique pour tout comportement. En effet, il faudrait que chaque routine d'intégration des comportements le fournisse, ce qui n'est pas le cas actuellement. Les algorithmes présentés et étudiés perdent donc un de leurs attraits majeurs qui était leur généralité. Néanmoins, on peut calculer son expression par exemple dans le cas de la loi de Von Mises à écrouissage isotrope et cinématique linéaires.

Le critère est de la forme :  $f(\boldsymbol{\tau}_{n+1}, \mathbf{x}_{n+1}, p_{n+1}) = \|\text{dev}(\boldsymbol{\tau}_{n+1} - \mathbf{x}_{n+1})\| - \sqrt{\frac{2}{3}} (\sigma_y + K p_{n+1}) \leq 0$

avec  $\mathbf{x}_{n+1} = C \boldsymbol{\varepsilon}_{n+1}$

La solution est obtenue par :  $\boldsymbol{\tau}_{n+1} = \boldsymbol{\tau}_{n+1}^{\text{pred}} - 2\mu \Delta \boldsymbol{\gamma}_n = \sigma_n + \mathbf{a} : \Delta \boldsymbol{\varepsilon}_{n+1} - 2\mu \Delta \boldsymbol{\gamma}_n$  avec

$$\mathbf{n} = \frac{\text{dev}(\boldsymbol{\tau}_{n+1}^{\text{pred}} - \mathbf{x}_n)}{\|\text{dev}(\boldsymbol{\tau}_{n+1}^{\text{pred}} - \mathbf{x}_n)\|} \text{ et } \Delta \boldsymbol{\gamma}_n = \frac{\|\text{dev}(\boldsymbol{\tau}_{n+1}^{\text{pred}} - \mathbf{x}_n)\| - \sqrt{\frac{2}{3}} (\sigma_y + K p_n)}{2\mu + \frac{2}{3} K + C} \text{ avec } \Delta \boldsymbol{\gamma}_n = \sqrt{\frac{3}{2}} \Delta p$$

Ce type d'algorithme est tout à fait semblable à la méthode utilisée en petites déformations (retour radial), ce qui permet de ne pas modifier la structure du code et était un critère de sélection de ces algorithmes.

Dans ce cas,

$$\begin{aligned} \frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \boldsymbol{\sigma}_n} &= \frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \boldsymbol{\tau}_{n+1}^{pred}} \frac{\partial \boldsymbol{\tau}_{n+1}^{pred}}{\partial \boldsymbol{\sigma}_n} = \frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \boldsymbol{\tau}_{n+1}^{pred}} \mathbf{I}_d = \mathbf{I}_d - 2\mu \frac{\partial(\Delta \boldsymbol{\gamma} \mathbf{n})}{\partial (\boldsymbol{\tau}_{n+1}^{pred})^{dev}} \frac{\partial (\boldsymbol{\tau}_{n+1}^{pred})^{dev}}{\partial \boldsymbol{\tau}_{n+1}^{pred}} \\ &= \mathbf{I}_d - 2\mu \left( \frac{\partial(\Delta \boldsymbol{\gamma})}{\partial (\boldsymbol{\tau}_{n+1}^{pred})^{dev}} \mathbf{n} + \Delta \boldsymbol{\gamma} \frac{\partial(\mathbf{n})}{\partial (\boldsymbol{\tau}_{n+1}^{pred})^{dev}} \right) \frac{\partial (\boldsymbol{\tau}_{n+1}^{pred})^{dev}}{\partial \boldsymbol{\tau}_{n+1}^{pred}} \end{aligned}$$

On voit intervenir trois termes distincts :

- la dérivée du déviateur :  $\frac{\partial (\boldsymbol{\sigma}^{dev})_{ij}}{\partial \sigma_{kl}} = \frac{\partial \left( \sigma_{ij} - \frac{tr \boldsymbol{\sigma}}{3} \delta_{ij} \right)}{\partial \sigma_{kl}} = \frac{1}{2} (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) - \frac{1}{3} \delta_{ij} \delta_{kl}$

- la dérivée du multiplicateur plastique :  $\frac{\partial(\Delta \boldsymbol{\gamma})}{\partial (\boldsymbol{\tau}_{n+1}^{pred})^{dev}} = \frac{1}{2\mu + \frac{2}{3}K + C} \frac{(\boldsymbol{\tau}_{n+1}^{pred})^{dev}}{\|(\boldsymbol{\tau}_{n+1}^{pred})^{dev}\|} = \frac{1}{2\mu + \frac{2}{3}K + C} \mathbf{n}_{kl}$

- la dérivée de la normale à la surface de charge :  $\frac{\partial(\mathbf{n}_{ij})}{\partial (\boldsymbol{\tau}_{n+1}^{pred})^{dev}} = \frac{1}{\|(\boldsymbol{\tau}_{n+1}^{pred})^{dev}\|} \left( \frac{1}{2} (\delta_{ik} \delta_{jl} + \delta_{jk} \delta_{il}) - \mathbf{n}_{ij} \mathbf{n}_{kl} \right)$

Au final, en notant  $\mathbf{I}_d^{4s}$  le premier tenseur identité d'ordre 4 défini par :  $\mathbf{I}_d^{4s} = \frac{1}{2} (\delta_{ik} \delta_{jl} + \delta_{jk} \delta_{il})$ , on obtient pour ce terme :

$$\frac{\partial (\boldsymbol{\tau}_{n+1})_{ij}}{\partial (\boldsymbol{\sigma}_n)_{kl}} = (\mathbf{I}_d^{4s})_{ijkl} - 2\mu \left( \frac{1}{2\mu + \frac{2}{3}K + C} \mathbf{n}_{ij} \mathbf{n}_{mn} + \frac{\Delta \boldsymbol{\gamma}}{\|(\boldsymbol{\tau}_{n+1}^{pred})^{dev}\|} \left( (\mathbf{I}_d^{4s})_{ijmn} - \mathbf{n}_{ij} \mathbf{n}_{mn} \right) \right) \left( (\mathbf{I}_d^{4s})_{mnkl} - \frac{1}{3} \delta_{mn} \delta_{kl} \right)$$

Cette expression n'est malheureusement pas valable pour les écrouissages non linéaires...

**Terme 4**  $\frac{\partial \boldsymbol{\sigma}_n}{\partial \mathbf{f}_{n+1}}$

Ce terme est de nouveau spécifique à chaque algorithme considéré ici.

## 9.2.1 Pour l'algorithme Simo\_Hughes\_1

Dans cet algorithme, la contrainte de Cauchy s'écrit  $\boldsymbol{\sigma}_n = \mathbf{f}_{n+\alpha} \boldsymbol{\tau}_n \mathbf{f}_{n+\alpha}^T$

D'où :

$$\begin{aligned} \left( \frac{\partial \boldsymbol{\sigma}_n}{\partial \mathbf{f}_{n+1}} \right)_{ijkl} &= \frac{\partial (\mathbf{f}_{n+\alpha})_{ip} (\boldsymbol{\tau}_n)_{pq} (\mathbf{f}_{n+\alpha})_{jq}}{\partial (\mathbf{f}_{n+1})_{kl}} = \alpha \delta_{ik} \delta_{pl} (\boldsymbol{\tau}_n)_{pq} (\mathbf{f}_{n+\alpha})_{jq} + (\mathbf{f}_{n+\alpha})_{ip} (\boldsymbol{\tau}_n)_{pq} \alpha \delta_{jk} \delta_{ql} \\ &= \alpha \delta_{ik} (\boldsymbol{\tau}_n)_{lq} (\mathbf{f}_{n+\alpha})_{jq} + (\mathbf{f}_{n+\alpha})_{ip} (\boldsymbol{\tau}_n)_{pl} \alpha \delta_{jk} \\ &= \alpha (\boldsymbol{\tau}_n)_{lq} \left( \delta_{ik} (\mathbf{f}_{n+\alpha})_{jq} + (\mathbf{f}_{n+\alpha})_{lq} \delta_{jk} \right) \end{aligned}$$

## 9.2.2 Pour l'algorithme GDEF\_HYPO\_ELAS

Dans cet algorithme, la contrainte de Cauchy s'écrit  $\boldsymbol{\sigma}_n = \mathbf{r}_{n+1} \boldsymbol{\tau}_n \mathbf{r}_{n+1}^T$

Donc :

$$\frac{\partial(\sigma_n)_{qp}}{\partial(\mathbf{f}_{n+1})_{kl}} = \frac{\partial((\mathbf{r}_{n+1})_{qa}(\boldsymbol{\tau}_n)_{ab}(\mathbf{r}_{n+1})_{pb})}{\partial(\mathbf{f}_{n+1})_{kl}} = \frac{\partial(\mathbf{r}_{n+1})_{qa}}{\partial(\mathbf{f}_{n+1})_{kl}}(\boldsymbol{\tau}_n)_{ab}(\mathbf{r}_{n+1})_{pb} + (\mathbf{r}_{n+1})_{qa}(\boldsymbol{\tau}_n)_{ab} \frac{\partial(\mathbf{r}_{n+1})_{pb}}{\partial(\mathbf{f}_{n+1})_{kl}}$$

Et comme  $(\mathbf{r}_{n+1})_{qa} = (\boldsymbol{\Lambda}_{n+1})_{qx}(\boldsymbol{\Lambda}_n)_{ax} = \exp[\boldsymbol{\Theta}]_{qb}(\boldsymbol{\Lambda}_n)_{bx}(\boldsymbol{\Lambda}_n)_{ax}$ , et en utilisant les résultats précédents,

$$\begin{aligned} \frac{\partial(\mathbf{r}_{n+1})_{qa}}{\partial(\mathbf{f}_{n+1})_{kl}} &= \frac{\partial \exp[\boldsymbol{\Theta}]_{qb}}{\partial(\boldsymbol{\Theta})_{mn}} \frac{\partial(\boldsymbol{\Theta})_{mn}}{\partial(\mathbf{f}_{n+1/2})_{rs}} \frac{\partial(\mathbf{f}_{n+1/2})_{rs}}{\partial(\mathbf{f}_{n+1})_{kl}} (\boldsymbol{\Lambda}_n)_{bx}(\boldsymbol{\Lambda}_n)_{ax} \\ &= \frac{1}{2} \frac{\partial \exp[\boldsymbol{\Theta}]_{qc}}{\partial(\boldsymbol{\Theta})_{mn}} \left( -(\mathbf{f}_{n+1/2}^{-1})_{nk}(\mathbf{f}_{n+1/2}^{-1})_{lm} + (\mathbf{f}_{n+1/2}^{-1})_{mk}(\mathbf{f}_{n+1/2}^{-1})_{ln} \right) (\boldsymbol{\Lambda}_n)_{cx}(\boldsymbol{\Lambda}_n)_{ax} \end{aligned}$$

Au final :

$$\begin{aligned} \frac{\partial(\sigma_n)_{qp}}{\partial(\mathbf{f}_{n+1})_{kl}} &= \frac{1}{2} (\boldsymbol{\tau}_n)_{ab} \left( -(\mathbf{f}_{n+1/2}^{-1})_{nk}(\mathbf{f}_{n+1/2}^{-1})_{lm} + (\mathbf{f}_{n+1/2}^{-1})_{mk}(\mathbf{f}_{n+1/2}^{-1})_{ln} \right) (\boldsymbol{\Lambda}_n)_{cx} \\ &\quad - \left( (\mathbf{r}_{n+1})_{pb} \frac{\partial \exp[\boldsymbol{\Theta}]_{qc}}{\partial(\boldsymbol{\Theta})_{mn}} (\boldsymbol{\Lambda}_n)_{ax} + (\mathbf{r}_{n+1})_{qa} \frac{\partial \exp[\boldsymbol{\Theta}]_{pc}}{\partial(\boldsymbol{\Theta})_{mn}} (\boldsymbol{\Lambda}_n)_{bx} \right) \end{aligned}$$

La dérivée de l'exponentielle tensorielle est donnée en Annexe.

**Terme 5**  $\frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \mathbf{x}_n}$

Ce terme, présent uniquement dans le cas d'un écrouissage cinématique, est du même type que le terme  $\frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \boldsymbol{\sigma}_n}$ , dans le sens qu'il est spécifique à chaque loi de comportement à écrouissage cinématique et donc empêche l'écriture d'une matrice tangente générique pour les formalismes.

Donnons son expression par exemple dans le cas de la loi de Von Mises à écrouissage cinématique linéaire.

On a alors :  $\boldsymbol{\tau}_{n+1} = \boldsymbol{\sigma}_n + \mathbf{a} : \Delta \boldsymbol{\varepsilon}_{n+1} - 2\mu \Delta \boldsymbol{\gamma} \mathbf{n}$  avec  $\mathbf{n} = \frac{\boldsymbol{\xi}^{pred}}{\|\boldsymbol{\xi}^{pred}\|}$  et  $\boldsymbol{\xi}^{pred} = dev(\boldsymbol{\sigma}_n + \mathbf{a} : \Delta \boldsymbol{\varepsilon}_{n+1} - \mathbf{x}_n)$

D'où  $\frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \mathbf{x}_n} = -2\mu \frac{\partial(\Delta \boldsymbol{\gamma} \mathbf{n})}{\partial \boldsymbol{\xi}^{pred}} \frac{\partial \boldsymbol{\xi}^{pred}}{\partial \mathbf{x}_n^{dev}} \frac{\partial(\mathbf{x}_n)^{dev}}{\partial \mathbf{x}_n}$  qui nécessite le calcul de quatre termes :

- la dérivée du déviateur, de la même forme que pour l'écrouissage isotrope :

$$\frac{\partial(\mathbf{x}_n^{dev})_{ij}}{\partial(\mathbf{x}_n)_{kl}} = \frac{1}{2} (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) - \frac{1}{3} \delta_{ij} \delta_{kl}$$

- la dérivée du prédicteur :  $\frac{\partial \boldsymbol{\xi}^{pred}}{\partial \mathbf{x}_n^{dev}} = -\mathbf{I}_d$

- la dérivée du multiplicateur  $\Delta \boldsymbol{\gamma} = \frac{\|\boldsymbol{\xi}^{pred}\| - \sqrt{\frac{2}{3}(\sigma_y + K p_n)}}{2\mu + \frac{2}{3}K + C}$ , d'où  $\frac{\partial(\Delta \boldsymbol{\gamma})}{\partial(\boldsymbol{\xi}^{pred})_{kl}} = \frac{1}{2\mu + \frac{2}{3}K + C} \mathbf{n}_{kl}$

- la dérivée de la normale à la surface de charge :  $\frac{\partial(\mathbf{n}_{ij})}{\partial(\boldsymbol{\xi}^{pred})_{kl}^{dev}} = \frac{1}{\|\boldsymbol{\xi}^{pred}\|} \left( \frac{1}{2} (\delta_{ik} \delta_{jl} + \delta_{jk} \delta_{il}) - \mathbf{n}_{ij} \mathbf{n}_{kl} \right)$

En combinant les différentes relations, on obtient finalement :

$$\frac{\partial(\boldsymbol{\tau}_{n+1})_{ij}}{\partial(\mathbf{x}_n)_{kl}} = 2\mu \left( \frac{1}{2\mu + \frac{2}{3}K + C} \mathbf{n}_{ij} \mathbf{n}_{mn} + \frac{\Delta\gamma}{\|\boldsymbol{\xi}^{pred}\|} \left( (\mathbf{I}_d^{4s})_{ijmn} - \mathbf{n}_{ij} \mathbf{n}_{mn} \right) \right) \left( (\mathbf{I}_d^{4s})_{mnkl} - \frac{1}{3} \delta_{mn} \delta_{kl} \right)$$

A l'instar du terme  $\frac{\partial \boldsymbol{\tau}_{n+1}}{\partial \boldsymbol{\sigma}_n}$ , l'expression donnée ici n'est valable que pour l'écouissage cinématique linéaire.

**Terme 6**  $\frac{\partial \mathbf{x}_n}{\partial \mathbf{f}_{n+1}}$

Ce dernier terme n'existe évidemment que pour l'écouissage cinématique. Il dépend de plus du formalisme choisi.

### 9.2.1 Pour SIMO\_HUGHES\_1

Dans ce cas, on a  $\mathbf{x}_n = \mathbf{f}_{n+\alpha} \mathbf{q}_n \mathbf{f}_{n+\alpha}^T$ , d'où :  $\left( \frac{\partial \mathbf{x}_n}{\partial \mathbf{f}_{n+1}} \right)_{ijkl} = \frac{\partial(\mathbf{f}_{n+\alpha})_{ip} (\mathbf{q}_n)_{pq} (\mathbf{f}_{n+\alpha})_{jq}}{\partial(\mathbf{f}_{n+1})_{kl}}$ .

On remarque que cette expression est de la même forme que pour  $\frac{\partial(\boldsymbol{\sigma}_n)_{ij}}{\partial(\mathbf{f}_{n+1})_{kl}}$ .

Par analogie, on obtient donc :

$$\frac{\partial(\mathbf{x}_n)_{ij}}{\partial(\mathbf{f}_{n+1})_{kl}} = \alpha(\mathbf{x}_n)_{lq} \left( \delta_{ik} (\mathbf{f}_{n+\alpha})_{jq} + (\mathbf{f}_{n+\alpha})_{iq} \delta_{jk} \right)$$

### 9.2.2 Pour GDEF\_HYPO\_ELAS

Dans ce cas on a  $\mathbf{x}_n = \mathbf{r}_{n+1} \mathbf{q}_n \mathbf{r}_{n+1}^T$ , d'où :  $\frac{\partial \mathbf{x}_n}{\partial \mathbf{f}_{n+1}} = \frac{\partial(\mathbf{r}_{n+1} \mathbf{q}_n \mathbf{r}_{n+1}^T)}{\partial \mathbf{f}_{n+1}}$ , qui a de nouveau la même forme que :

$$\frac{\partial \boldsymbol{\sigma}_n}{\partial \mathbf{f}_{n+1}}$$

On obtient donc par analogie :

$$\frac{\partial(\mathbf{x}_n)_{qp}}{\partial(\mathbf{f}_{n+1})_{kl}} = \frac{1}{2} (\mathbf{q}_n)_{ab} \left( -(\mathbf{f}_{n+1/2}^{-1})_{nk} (\mathbf{f}_{n+1/2}^{-1})_{lm} + (\mathbf{f}_{n+1/2}^{-1})_{mk} (\mathbf{f}_{n+1/2}^{-1})_{ln} \right) (\boldsymbol{\Lambda}_n)_{cx} \\ \left( (\mathbf{r}_{n+1})_{pb} \frac{\partial \exp[\boldsymbol{\Theta}]}{\partial(\boldsymbol{\Theta})_{mn}} (\boldsymbol{\Lambda}_n)_{ax} + (\mathbf{r}_{n+1})_{qa} \frac{\partial \exp[\boldsymbol{\Theta}]}{\partial(\boldsymbol{\Theta})_{mn}} (\boldsymbol{\Lambda}_n)_{bx} \right)$$

### Bilan

Deux points importants sont à retenir principalement :

- les expressions sont lourdes, avec de nombreux termes à calculer, ce qui augure de temps de calcul importants,
- du fait de certains termes dépendants de la loi de comportement utilisée, il n'est pas possible de donner une expression générique pour les matrices tangentes. Par conséquent, à moins de programmer pour chaque loi de comportement le calcul qui lui correspond, la matrice tangente ne peut pas être exacte.

On trouvera dans la réf [15], un bilan de l'implémentation de ces expressions en termes de performances et de vitesse de convergence à travers des exemples.

## 10 Annexe 2 : Calcul des déformations logarithmiques

### 10.1 Notations :

$A$  désigne un tenseur d'ordre 2, et  $\bar{A}$  un tenseur d'ordre 4

nous adoptons dans la mise en oeuvre la notation de Voigt, (voir par exemple [16]) définie par :

$$A = \begin{pmatrix} A_{11} \\ A_{22} \\ A_{33} \\ \sqrt{2} A_{12} \\ \sqrt{2} A_{13} \\ \sqrt{2} A_{23} \end{pmatrix} \quad \bar{A} = \begin{pmatrix} A_{1111} & A_{1122} & A_{1133} & \sqrt{2} A_{1112} & \sqrt{2} A_{1123} & \sqrt{2} A_{1113} \\ A_{2211} & A_{2222} & A_{2233} & \sqrt{2} A_{2212} & \sqrt{2} A_{2223} & \sqrt{2} A_{2213} \\ A_{3311} & A_{3322} & A_{3333} & \sqrt{2} A_{3312} & \sqrt{2} A_{3323} & \sqrt{2} A_{3313} \\ \sqrt{2} A_{1211} & \sqrt{2} A_{1222} & \sqrt{2} A_{1233} & 2 A_{1212} & 2 A_{1223} & 2 A_{1213} \\ \sqrt{2} A_{1311} & \sqrt{2} A_{1322} & \sqrt{2} A_{1333} & 2 A_{1312} & 2 A_{1323} & 2 A_{1313} \\ \sqrt{2} A_{2311} & \sqrt{2} A_{2322} & \sqrt{2} A_{2333} & 2 A_{2312} & 2 A_{2323} & 2 A_{2313} \end{pmatrix}$$

où les composantes relatives à la notation de Voigt seront désignées par une lettre grecque.

On a alors les propriétés suivantes :

$$\|A_{ij}\| = \|A_{\alpha}\|$$

$$A : B = A_{ij} B_{ij} = A_{\alpha} B_{\alpha} \quad \bar{A} : B = A_{ijkl} B_{kl} = A_{\alpha\beta} B_{\beta} \quad \bar{A} : \bar{B} = A_{ijkl} B_{klmn} = A_{\alpha\beta} B_{\beta\gamma}$$

L'inverse d'un tenseur d'ordre 4 comportant des symétries mineures ( $A_{ijrs} = A_{jirs} = A_{ijsr}$ ) s'écrit :

$$\bar{A} : \bar{A}^{-1} = \bar{I}_d \quad A_{ijrs} A_{rskl}^{-1} = I_{ijkl} = \frac{1}{2} (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) \quad \text{et avec la notation de Voigt : } A_{\alpha\gamma} A_{\gamma\beta}^{-1} = I_{\alpha\beta} = \delta_{\alpha\beta}$$

### 10.2 Expression des contraintes en configuration Lagrangienne

La puissance des efforts intérieurs s'écrit :

$$p_{\text{int}} = T : \dot{E} = S : \bar{P}^{-1} \dot{E} \quad \text{avec} \quad \bar{P} = 2 \frac{\partial E}{\partial C} \quad \text{ce qui permet de calculer : } S = T : P \quad \text{ou} \quad S_{ij} = T_{kl} : P_{klj}$$

Pour calculer le tenseur des contraintes de Cauchy, il suffit d'écrire :  $\sigma = \frac{1}{\det F} F \cdot S \cdot F^T$

#### Cas particulier des contraintes planes :

Dans ce cas, on ne connaît pas entièrement  $\det F$  : en effet, la composante  $zz$  du tenseur de déformations logarithmiques  $E$  est inconnue, car dépendante de la loi de comportement. En se limitant aux comportements tels que  $\det F^P = 0$  (incompressibilité plastique), on a alors :  $\det F = \det F^e$ .

Suivant [16] on peut calculer cette expression :  $\det F^e = \exp(E_{xx}^e + E_{yy}^e + E_{zz}^e)$  où  $E^e$  représente la partie élastique des déformations logarithmiques, connue pour toute loi de comportement élasto-plastique ou élasto-viscoplastique par la loi de Hooke :  $E^e = \Lambda^{-1} T$ .

En effet la définition des déformations logarithmiques :  $E_{ij} = \frac{1}{2} \sum_{k=1,3} \log(\lambda^{(k)}) N_i^{(k)} \otimes N_j^{(k)}$  conduit à  $\det F^e = \sqrt{\det F^T F} = \sqrt{\lambda_1 \lambda_2 \lambda_3}$   $\lambda_i$  étant les valeurs propres de  $F^T F$ . Donc  $\log(\det F^e) = \frac{1}{2} \log(\lambda_1) + \log(\lambda_2) + \log(\lambda_3)$  d'où le résultat.

## 10.3 Expression de l'opérateur tangent en configuration Lagrangienne

En dérivant l'expression  $S = T : P$  par rapport au temps :

$$\dot{S} = \dot{T} : \bar{P} + T : \dot{\bar{P}} = \left( \frac{\partial T}{\partial E} : \dot{E} \right) : \bar{P} + T : \left( \frac{\partial \bar{P}}{\partial C} : \dot{C} \right) = \left[ \frac{\partial T}{\partial E} : \left( \frac{\partial E}{\partial C} : \dot{C} \right) \right] : \bar{P} + T : \left( \frac{\partial \bar{P}}{\partial C} : \dot{C} \right) \text{ soit}$$

$$\dot{S} = \left( \bar{P}^T : \bar{E}^p : \bar{P} + T : \bar{L} \right) : \frac{1}{2} \dot{C}, \text{ avec } \bar{L} = 4 \frac{\partial^2 E}{\partial C \partial C} \text{ et } \bar{E}^p = \frac{\partial T}{\partial E}$$

ce qui définit l'opérateur tangent  $\bar{C}^{ep} = \left( \bar{P}^T \bar{E}^p \bar{P} + T : \bar{L} \right)$  qui vérifie  $\dot{S} = \bar{C}^{ep} : \frac{1}{2} \dot{C}$

ou encore, en fonction des déformations de Green-Lagrange  $\Delta = \frac{1}{2} (C - I_d)$  :

$$\frac{\partial S}{\partial \Delta} = \frac{\partial S}{\partial C} : \frac{\partial C}{\partial \Delta} = 2 \frac{\partial S}{\partial C} = \bar{C}^{ep}$$

L'expression de cet opérateur tangent ainsi que le tenseur des contraintes, tous deux en configuration Lagrangienne, permettent, pour le calcul des forces internes, d'utiliser une formulation variationnelle en configuration initiale, comme dans [R5.03.20] par exemple.

On écrit l'équilibre sous forme variationnelle sur la configuration initiale.

$$\delta W_{int} \cdot \delta v + SW_{ext} \cdot \delta v = 0, \quad \forall \delta v \text{ cinématiquement admissible}$$

Sous l'hypothèse que le chargement ne dépend pas de la transformation géométrique, le travail virtuel des efforts extérieurs s'écrit comme une forme linéaire :

$$\delta W_{ext} \cdot \delta v = \int_{\Omega_o} \rho_o F_i \delta v_i d\Omega_o + \int_{\partial_F \Omega_o} T_i^d \delta v_i dS_o$$

$F$  : chargement volumique

$T^d$  : chargement surfacique s'exerçant sur le bord  $\partial_F \Omega_o$

Là encore, nous choisissons la configuration initiale comme configuration de référence, pour exprimer le travail des efforts intérieurs [R5,03,20].[R7.02.03].

$$SW_{int} \cdot \delta v = - \int_{\Omega_o} F_{ik} S_{kl} \delta v_{i,l} d\Omega_o \text{ avec : } \delta v_{i,l} = \frac{\partial dv_i}{\partial X_l}$$

Dans l'optique d'une résolution par une méthode de Newton, il importe d'exprimer également la variation seconde du travail virtuel des efforts intérieurs, à savoir :

$$d^2 W_{int} \cdot \delta u \cdot \delta v = - \int_{\Omega_o} \delta u_{i,k} S_{kl} \delta v_{i,l} d\Omega_o \quad \text{Rigidité géométrique}$$

$$\dots - \int_{\Omega_o} \delta u_{i,q} F_{ip} \left( \frac{\partial \mathbf{S}}{\partial \Delta} \right)_{pqkl} F_{jk} \delta v_{j,l} d\Omega_o$$

Rigidité élastique

## 10.4 Calcul effectif des déformations logarithmiques

Elles sont définies par :  $\mathbf{E} = \frac{1}{2} \log(\mathbf{C}) = \frac{1}{2} \log(\mathbf{F}^T \cdot \mathbf{F})$  (en toute rigueur, il faudrait ajouter le tenseur métrique dans le cas d'une configuration initiale définie dans un espace différent de l'espace euclidien (cas des coques par exemple)). Pour simplifier les écritures, nous nous placerons dans le cas d'une configuration initiale euclidienne, les composantes des vecteurs et tenseurs étant écrites dans un repère orthonormé 3D. La restriction au 2D est immédiate.

Le calcul de la déformation logarithmique ne peut se faire que dans le repère propre.

Il faut donc déterminer les 3 valeurs propres  $\lambda^{(i)}$  et vecteurs propres  $\mathbf{N}^{(i)}$  solutions du problème aux valeurs propres  $\mathbf{C} \mathbf{N}^{(i)} = \lambda^{(i)} \mathbf{N}^{(i)}$

On peut alors calculer les 3 valeurs :  $e^{(i)} = \frac{1}{2} \log(\lambda^{(i)})$

Les déformations logarithmiques sont alors transportées dans l'espace d'origine par :

$$\mathbf{E}_{ij} = \frac{1}{2} \sum_{k=1,3} \log(\lambda^{(k)}) \mathbf{N}_i^{(k)} \otimes \mathbf{N}_j^{(k)} \text{ car la fonction log est une fonction isotrope du tenseur } \mathbf{C} \text{ [16].}$$

Pour le post-traitement, c'est à dire le calcul du tenseur des contraintes et de l'opérateur tangent, il faut calculer les quantités  $\bar{\mathbf{P}} = 2 \frac{\partial \mathbf{E}}{\partial \mathbf{C}}$  et  $\mathbf{T} : \bar{\mathbf{L}}$

$$\bar{\mathbf{P}} = \sum_{i=1,3} \frac{1}{2} d^{(i)} \mathbf{N}^{(i)} \otimes \mathbf{N}^{(i)} \otimes \mathbf{M}^{(ii)} + \sum_{i=1,3} \sum_{j \neq i}^3 \theta_{ij} \mathbf{N}^{(i)} \otimes \mathbf{N}^{(j)} \otimes \mathbf{M}^{(ij)}$$

$$\begin{aligned} \mathbf{T} : \bar{\mathbf{L}} &= \sum_i^3 \frac{1}{4} f^{(i)} \zeta^{(ii)} \mathbf{M}^{(ii)} \otimes \mathbf{M}^{(ii)} + \sum_i^3 \sum_{j \neq i}^3 \sum_{k \neq i, k \neq j}^3 2\eta \zeta^{(ij)} \mathbf{M}^{(ik)} \otimes \mathbf{M}^{(ij)} \\ &+ \sum_i^3 \sum_{j \neq i}^3 2\xi^{(ij)} [\zeta^{(ij)} (\mathbf{M}^{(ij)} \otimes \mathbf{M}^{(jj)} + \mathbf{M}^{(jj)} \otimes \mathbf{M}^{(ij)}) + \zeta^{(jj)} \mathbf{M}^{(ij)} \otimes \mathbf{M}^{(ij)}] \end{aligned}$$

$$\text{avec } d^{(i)} = \frac{1}{\lambda^{(i)}} \quad f^{(i)} = \frac{-2}{(\lambda^{(i)})^2} \quad \zeta^{(ij)} = \mathbf{T} : \mathbf{N}^{(i)} \otimes \mathbf{N}^{(j)} \quad \mathbf{M}_{ab}^{(ij)} = \mathbf{N}_a^{(i)} \mathbf{N}_b^{(j)} + \mathbf{N}_a^{(j)} \mathbf{N}_b^{(i)}$$

et  $\theta^{(ij)}$ ,  $\xi^{(ij)}$  et  $\eta^{(ij)}$  définis par :

• si toutes les valeurs propres sont différentes :  $\theta^{(ij)} = \frac{e^{(i)} - e^{(j)}}{(\lambda^{(i)} - \lambda^{(j)})}$ ,  $\xi^{(ij)} = \frac{(\theta^{(ij)} - \frac{1}{2} d^{(j)})}{(\lambda^{(i)} - \lambda^{(j)})}$

$$\eta = \sum_i^3 \sum_{j \neq i}^3 \sum_{k \neq i, k \neq j}^3 \frac{e^{(i)}}{2(\lambda^{(i)} - \lambda^{(j)})(\lambda^{(i)} - \lambda^{(k)})}$$

• si deux valeurs propres sont égales :

$$\lambda^{(i)} = \lambda^{(j)} \neq \lambda^{(k)} : \theta^{(ij)} = \theta^{(ji)} = \frac{1}{2} d^{(j)} \quad \xi^{(ij)} = \xi^{(ji)} = \frac{1}{8} f^{(j)} \quad \eta = \xi^{(ki)}$$
$$\theta^{(mn)} = \frac{e^{(m)} - e^{(n)}}{\lambda^{(m)} - \lambda^{(n)}} \quad \xi^{(mn)} = \frac{\left(\theta^{(mn)} - \frac{1}{2} d^{(n)}\right)}{\left(\lambda^{(m)} - \lambda^{(n)}\right)} \quad n = k, m \in \{i, j\} \quad \text{ou} \quad m = k, n \in \{i, j\}$$

•si les trois valeurs propres sont égales :  $\lambda^{(i)} = \lambda^{(j)} = \lambda^{(k)} : \theta^{(ij)} = \frac{1}{2} d^{(j)} \quad \xi^{(ij)} = \frac{1}{8} f^{(j)} \quad \eta = \frac{1}{8} f^{(j)}$