

Outils numériques

L'implantation numérique ainsi que la validation des modèles présentés au chapitre précédent est détaillée ici. Le logiciel ABAQUS sera l'outil d'analyse employé dans ces travaux. Outre les modèles disponibles par défaut, il offre la possibilité à l'utilisateur de mettre en oeuvre des lois de comportement particulières au travers de routines spécifiques. La réponse (thermo)mécanique est déterminée par ces routines utilisateurs au point d'intégration ce qui permet de s'affranchir de tout développement spécifiquement lié à la méthode éléments finis.

En fin de chapitre, les méthodes de génération de microstructures numériques seront explicitées.

3.1 Implantation du modèle de plasticité cristalline

Le cadre de modélisation viscoplastique a été retenu dans ces travaux pour deux raisons : d'une part, comme révélé par la campagne expérimentale de caractérisation du chapitre 4, les matériaux étudiés présentent une sensibilité visqueuse qui, tout en étant faible n'est pas négligeable. D'autre part, la plasticité cristalline non visqueuse conduit à une indétermination concernant les systèmes de glissements activés au cours de la déformation. Les procédures mentionnées dans la littérature pour lever cette ambiguïté s'accompagnent d'un coût numérique parfois considérable qui se rajoute au temps de

calcul (déjà significatif) nécessaire pour déterminer l'état de contraintes aux points d'intégration. Dans l'optique d'une utilisation des outils développés sur des cas industriels, la viscoplasticité représente alors un bon compromis entre l'efficacité numérique et la précision des calculs.

Des algorithmes explicites sont retenus pour l'intégration de la loi de comportement. En effet, les algorithmes implicites, basés sur une variable primaire (taux de glissement, gradient de déformation plastique...) dont l'estimation initiale est corrigée par des méthodes itératives de type Newton-Raphson, présentent un coût numérique plus élevé que celui des algorithmes explicites pour une précision comparable ((Ling *et al.*, 2005), (Dumoulin *et al.*, 2009)). De plus, dans le cadre d'applications à des procédés de mise en forme, les fortes non-linéarités géométriques et les conditions de contact fluctuantes entre les outils et la tôle nécessitent forcément, indépendamment de la loi de comportement employée, de petits incréments de chargement. Cette caractéristique confère de facto un avantage certain aux algorithmes explicites en termes de compromis performance - précision - robustesse.

3.1.1 Implantation dans ABAQUS/Explicit

3.1.1.1 ABAQUS/Explicit et VUMAT

Le solveur dynamique explicite ABAQUS/Explicit a été conçu pour résoudre des problèmes dynamiques. Cependant, il est appliqué avec succès pour des problèmes quasi-statiques et l'expérience montre qu'il se révèle souvent plus efficace que le solveur implicite ABAQUS/Standard. En effet, ABAQUS/Standard doit s'assurer que l'incrément de déplacement calculé vérifie au mieux l'équilibre mécanique global ; pour cela il effectue des corrections itératives sur la solution par un algorithme de type Newton-Raphson jusqu'à atteindre une précision satisfaisante. Il passe en outre par la formulation et l'inversion d'une matrice tangente, opération également très coûteuse en temps de calcul lorsque le modèle contient un grand nombre de noeuds. De plus, la convergence de l'algorithme de correction n'est pas assurée pour des conditions de contact complexes.

Par contre, ABAQUS/Explicit détermine l'incrément de déplacement après intégration temporelle explicite de la relation d'équilibre dynamique :

$$\ddot{\underline{u}} = \mathbf{M}^{-1} \cdot (\underline{F} - \underline{I}) \quad (3.1)$$

avec \underline{u} le vecteur des déplacements nodaux, \mathbf{M} la matrice de masse, \underline{F} le vecteur des efforts externes et \underline{I} celui des efforts internes.

L'équilibre mécanique n'est pas vérifié et aucune itération n'est faite ; les erreurs s'accroissent à chaque intégration, la stabilité est donc conditionnelle et il faut utiliser des pas de temps suffisamment réduits. Le pas de temps doit être inférieur au temps nécessaire à une onde de compression pour se propager à travers le plus petit élément du maillage, soit :

$$\Delta t \leq \left(\frac{L^e}{c_d} \right) \quad \text{avec } c_d = \sqrt{\frac{\lambda + 2\mu}{\rho}} \quad (3.2)$$

où L^e est une longueur caractéristique d'élément, c_d est la vitesse de propagation d'une onde de compression, λ et μ sont les coefficients de Lamé du matériau et ρ sa densité.

La possibilité fournie aux utilisateurs par ABAQUS/Explicit pour définir une loi de comportement est la routine VUMAT (Vectorized User MATerial). A partir des variables fournies en entrée par ABAQUS/Explicit à chaque incrément, elle a pour objectif de calculer le tenseur de contraintes de Cauchy au point d'intégration. Aussi, bien que la VUMAT oeuvre au point d'intégration, les réponses d'un certain nombre de points (NBLOCK) sont évaluées simultanément lors d'un seul appel de la routine. Cette structure vectorisée permet d'accroître significativement l'efficacité numérique en répartissant le nombre de points évalués sur différents processeurs via les procédures de calcul parallèle. Le formalisme de la VUMAT comporte toutefois certaines particularités dont il faut tenir compte lors de l'implantation d'un modèle.

En effet, dans le souci de garantir l'objectivité incrémentale des lois de comportement, le repère de travail employé par ABAQUS/Explicit est le référentiel tournant associé à la dérivée de Green-Naghdi dénommé référentiel en rotation propre. Comme détaillé au chapitre 2, ce référentiel est généré par la rotation pure \mathbf{R} issue de la décomposition polaire du gradient de la transformation $\mathbf{F} = \mathbf{R} \cdot \mathbf{U}$ où \mathbf{U} est un tenseur d'élongation pure. Pour l'implantation d'une loi de plasticité cristalline au travers d'une VUMAT, on peut donc distinguer trois repères :

- le référentiel global qui est fixe dans l'espace ;
- le référentiel en rotation propre qui suit la rotation moyenne de la matière au point matériel (au sens de Green-Naghdi), dans lequel le code de calcul travaille et exprime les tenseurs. Nous le désignerons par l'anglicisme MECS (Material Element Coordinate System) et un tenseur $\{\bullet\}$ exprimé dans ce référentiel s'écrira $\{\hat{\bullet}\}$;

- le référentiel cristallographique défini par trois axes cristallographiques et qui suit la rotation du réseau cristallin. Il sera dénoté CACS (Crystal Associated Coordinate System). Un tenseur $\{\bullet\}$ exprimé dans le CACS sera noté $\{\tilde{\bullet}\}$.

3.1.1.2 Algorithme en repère global

La loi de comportement telle que formulée en 2.3.1.3 est exprimée dans le repère global. Le tenseur des contraintes ainsi déterminé dans le repère global doit être retourné, au code, exprimé dans le MECS. Pour ce faire, le tenseur de rotation \mathbf{R} qui transforme une base orthonormée directe du repère fixe en une base orthonormée directe du MECS est employé. ABAQUS/Explicit 6.11 fournit comme entrée de la routine VUMAT le gradient de la transformation en début (à t) et en fin d'incrément (à $t + \Delta t$), notés respectivement \mathbf{F}_n et \mathbf{F}_{n+1} . De même, le tenseur d'élongation pure à droite \mathbf{U} est fourni en début et en fin d'incrément, soit \mathbf{U}_n et \mathbf{U}_{n+1} . Soulignons ici que d'après la documentation ([Hibbitt et al., 1992](#)), le tenseur \mathbf{U} fourni est bien exprimé dans le MECS. En effet, \mathbf{F} et \mathbf{R} sont des tenseurs mixtes, qui ont un pied dans les deux bases à savoir celle du repère global et celle du MECS. On écrit en terme de composantes :

$$F_{iJ} = R_{iK} U_{KJ} \quad (3.3)$$

où les indices en majuscule sont relatifs à la base de Green-Naghdi et ceux en minuscule à la base globale. Ainsi, les tenseurs de rotation en début d'incrément \mathbf{R}_n et en fin d'incrément \mathbf{R}_{n+1} sont obtenus par la relation :

$$\mathbf{R} = \mathbf{F} \cdot \mathbf{U}^{-1} \quad (3.4)$$

Les variables d'état sont stockées pour chaque point d'intégration en fin d'incrément dans un vecteur et récupérées au début de l'incrément suivant. Pour l'algorithme en repère global, on compte comme variables d'états :

- \mathbf{F}^e le gradient élastique ;
- \mathbf{F}^p le gradient plastique ;
- $\tau_c^{s=1..12}$ les cissions critiques d'activation des systèmes de glissement ;

- $\rho^{s=1..12}$ les densités sur les systèmes de glissement si le modèle d'écroissage de (Tabourot, 2001) est employé ;
- γ le glissement total sur tous les systèmes ;
- \mathbf{Q} la matrice d'orientation.

Pour garder une certaine lisibilité, les variables d'état intégrées numériquement (exception faite de \mathbf{Q}) seront regroupées dans un pseudo-vecteur $\mathbf{H} = (\mathbf{F}^p, \tau_c^s, \rho^s, \gamma)$. De manière similaire, nous introduisons l'ensemble $\mathbf{G}_n = (\dot{\gamma}^s)_n$ des taux de glissement calculés à l'instant n . Précisons que dans le schéma ici adopté, les taux de glissement à l'instant n sont estimés explicitement à partir de données à l'instant $n - 1$.

On peut donc écrire sous une forme globale :

$$\dot{\mathbf{H}} = \mathbf{h}(t, \mathbf{H}, \mathbf{G}) \quad (3.5)$$

où \mathbf{h} représente les différentes équations d'évolution des variables internes.

L'intégration numérique des relations est effectuée par des schémas explicites de Runge-Kutta d'ordre 1 (méthode d'Euler explicite), 2 (méthode du point milieu) ou 4. On considère que l'indice n correspond à l'instant t_n et que l'indice $n + 1$ renvoie à l'instant $t_n + \Delta t$ où Δt est l'intervalle de temps entre deux instants consécutifs.

En employant la forme unifiée des trois méthodes de Runge-Kutta évoquées, on obtient pour un schéma d'ordre N :

$$\begin{aligned} \Delta \mathbf{H}_i &= \mathbf{h}(t_n + a_i \Delta t, \mathbf{H}_n + a_i \Delta \mathbf{H}_{i-1}, \mathbf{G}_i) \Delta t \\ \mathbf{H}_{n+1} &= \mathbf{H}_n + \sum_{i=1}^N b_i \Delta \mathbf{H}_i \end{aligned} \quad (3.6)$$

avec les coefficients de pondération suivants

$$\begin{array}{lll} N = 1 & a_i = \{0\} & b_i = \{1\} \\ N = 2 & a_i = \left\{0, \frac{1}{2}\right\} & b_i = \left\{\frac{1}{2}, \frac{1}{2}\right\} \\ N = 4 & a_i = \left\{0, \frac{1}{2}, \frac{1}{2}, 1\right\} & b_i = \left\{\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6}\right\} \end{array} \quad (3.7)$$

Des relations similaires à celles de (3.6) sont établies pour l'intégration du tenseur de contraintes et l'algorithme en repère global est décrit dans le Tableau 3.1.

L'actualisation de la matrice d'orientation se fait par une intégration avec la formule de Rodrigues ((Raphanel *et al.*, 2004), (Dai, 2015)) pour le calcul du terme exponentiel; cette méthodologie permet de garantir l'orthogonalité des matrices d'orientation. L'orientation initiale donne la matrice de rotation \mathbf{Q}_0 fonction des angles d'Euler φ_1 , ϕ , φ_2 (en convention de Bunge) et a pour expression :

$$\mathbf{Q}_0 = \begin{bmatrix} c(\varphi_1)c(\varphi_2) - s(\varphi_1)c(\phi)s(\varphi_2) & -c(\varphi_1)s(\varphi_2) - s(\varphi_1)c(\phi)c(\varphi_2) & s(\phi_1)s(\phi) \\ s(\varphi_1)c(\varphi_2) + c(\varphi_1)c(\phi)c(\varphi_2) & -s(\varphi_1)s(\varphi_2) + c(\varphi_1)c(\phi)c(\varphi_2) & -c(\varphi_1)s(\phi) \\ s(\varphi_2)s(\phi) & s(\phi)c(\varphi_2) & c(\phi) \end{bmatrix} \quad (3.8)$$

où $c(\varphi) = \cos(\varphi)$ et $s(\varphi) = \sin(\varphi)$.

3.1.1.3 Validation de l'implantation

Des simulations d'essais de traction et de cisaillement sur monocristal (un seul élément fini) effectuées par (Raphanel *et al.*, 2004) ont été reproduites avec la VUMAT développée. (Raphanel *et al.*, 2004) mettent en oeuvre, dans un code EF implicite (SARPP), un algorithme explicite de plasticité cristalline avec un schéma numérique de Runge-Kutta d'ordre 2. Une maille cristalline de type CFC est choisie et le modèle d'érouissage PAN, dont les paramètres sont listés dans le Tableau 3.2, est employé. La vitesse de chargement imposée en traction comme en cisaillement est de $0,001 \text{ s}^{-1}$.

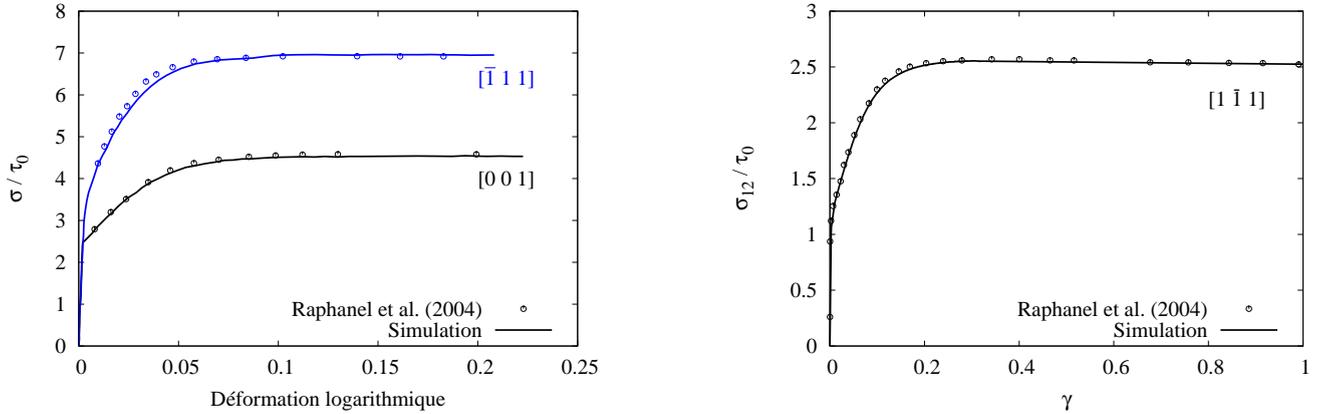
En outre, différentes orientations initiales sont attribuées aux monocristaux; l'axe cristallographique initialement colinéaire à la direction du chargement imposé est spécifié sur la Figure 3.1 qui illustre la comparaison entre les résultats de la présente approche et ceux de (Raphanel *et al.*, 2004).

- Initialisation : $\tilde{\mathbf{C}}; \tilde{\mathbf{S}}_0^{s=1..12}; \mathbf{F}^e = \mathbf{F}^p = \mathbf{I}; \mathbf{Q}_0; \tau_{c_0}^{s=1..12}, \rho_0^{s=1..12}, \gamma = 0$
- Données d'entrée : $\hat{\boldsymbol{\sigma}}_n, \mathbf{F}_{n+1}, \mathbf{F}_n, \hat{\mathbf{U}}_{n+1}, \hat{\mathbf{U}}_n$
 1. Récupération des variables d'état
 - $\mathbf{F}_n^e, \mathbf{Q}_n, \mathbf{H}_n = (\mathbf{F}^p, \tau_c^s, \rho^s, \gamma)_n$
 2. Transport de $\hat{\boldsymbol{\sigma}}_n$ du MECS vers le repère fixe
 - $\mathbf{R}_n = \mathbf{F}_n \cdot \hat{\mathbf{U}}_n^{-1}$
 - $\boldsymbol{\sigma}_n = \mathbf{R}_n \cdot \hat{\boldsymbol{\sigma}}_n \cdot \mathbf{R}_n^T$
 3. Transport de $\tilde{\mathbf{C}}$ et des $\tilde{\mathbf{S}}_0^s$ du CACS vers le repère fixe
 - $\mathbf{C} = \mathbf{Q}_n \cdot \mathbf{Q}_n \cdot \tilde{\mathbf{C}} \cdot \mathbf{Q}_n^T \cdot \mathbf{Q}_n^T$
 - Faire de $s=1$ à 12 : $\mathbf{S}_0^s = \mathbf{Q}_0 \cdot \tilde{\mathbf{S}}_0^s \cdot \mathbf{Q}_0^T$
 4. Transport de \mathbf{S}_0^s en début d'incrément
 - Faire de $s=1$ à 12 : $\mathbf{S}_n^s = \mathbf{F}_n^e \cdot \mathbf{S}_0^s \cdot \mathbf{F}_n^{e-1}$; $\mathbf{M}_n^s = \text{sym} \{ \mathbf{S}_n^s \}$
 5. Calcul des taux de déformation et de rotation (cinématiques)
 - $\mathbf{D}_{n+1} = \text{sym} \{ \dot{\mathbf{F}} \cdot \mathbf{F}_{n+1}^{-1} \}$; $\mathbf{W}_{n+1} = \text{asym} \{ \dot{\mathbf{F}} \cdot \mathbf{F}_{n+1}^{-1} \}$
 6. Boucle de Runge-Kutta : faire de $i=1$ à N
 - $\boldsymbol{\sigma}_i = \boldsymbol{\sigma}_n + a_i \Delta \boldsymbol{\sigma}_{i-1}$; $\mathbf{H}_i = \mathbf{H}_n + a_i \Delta \mathbf{H}_{i-1}$
 - Faire de $s=1$ à 12 : $\tau^s = \boldsymbol{\sigma}_i : \mathbf{M}_n^s$; $\dot{\gamma}^s = \dot{\gamma}_0 \text{signe} \{ \tau^s \} \left| \frac{\tau^s}{\tau_{c_i}^s} \right|$
 - Calcul de \mathbf{D}^p et \mathbf{W}^p (2.55)
 - Calcul de $\Delta \boldsymbol{\sigma}_i$ et $\Delta \mathbf{H}_i$
 7. Mise à jour des variables
 - $\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_n + \sum_{i=1}^N b_i \Delta \boldsymbol{\sigma}_i$
 - $\mathbf{H}_{n+1} = \mathbf{H}_n + \sum_{i=1}^N b_i \Delta \mathbf{H}_i$
 - $\mathbf{F}_{n+1}^e = \mathbf{F}_{n+1} \cdot (\mathbf{F}_{n+1}^p)^{-1}$
 8. Actualisation de la matrice d'orientation (calcul de l'exponentielle par la formule de Rodrigues)
 - $\mathbf{Q}_{n+1} = \exp \left(\Delta t (\mathbf{W} - \mathbf{W}^p)_{n+1} \right) \cdot \mathbf{Q}_n$
 9. Transport de $\boldsymbol{\sigma}_{n+1}$ du repère fixe vers le MECS
 - $\mathbf{R}_{n+1} = \mathbf{F}_{n+1} \cdot \hat{\mathbf{U}}_{n+1}^{-1}$
 - $\hat{\boldsymbol{\sigma}}_{n+1} = \mathbf{R}_{n+1}^T \cdot \boldsymbol{\sigma}_{n+1} \cdot \mathbf{R}_{n+1}$
- Stocker les variables d'état et retourner $\hat{\boldsymbol{\sigma}}_{n+1}$ au solveur

TABLEAU 3.1: Algorithme de la routine VUMAT en repère global.

E	ν	n	q	$\dot{\gamma}_0$	τ_0	h_0/τ_0	τ^*/τ_0
1	0,3	50	1	0,001	0,001	8,9	1,8

TABLEAU 3.2: Paramètres matériau utilisés pour les simulations.



(a) Traction uniaxiale sur un monocristal

(b) Cisaillement sur un monocristal

FIGURE 3.1: Comparaison des prédictions de la VUMAT (repère global) par rapport à (Raphanel *et al.*, 2004). σ est la contrainte dans la direction de traction et σ_{12} est la contrainte de cisaillement dans le plan (12).

Un bon accord entre les deux approches est obtenu pour les essais de traction [Figure 3.1\(a\)](#) en utilisant un schéma d'intégration numérique d'Euler. Pour ce qui est du test en cisaillement qui induit des rotations importantes des systèmes de glissement, le schéma de Runge-Kutta d'ordre 4 fut employé afin d'imposer des pas de temps relativement importants ($\Delta t = 0.1$ comme dans (Raphanel *et al.*, 2004)), ce qui conduit à des résultats satisfaisants sur la [Figure 3.1\(b\)](#).

3.1.1.4 Algorithme en repère cristallographique (CACS)

L'implantation réalisée au paragraphe précédent implique de multiples transferts de tenseurs entre trois référentiels pour chaque point d'intégration et à chaque incrément. Dans le cas de simulations de procédés nécessitant un grand nombre d'éléments finis et d'incrément de calcul, on peut intuitivement envisager des difficultés portant sur :

- l’efficacité : les transferts entre les différents référentiels étant coûteux en temps de calcul ;
- la précision numérique : la multiplication des opérations au point matériel induisant inéluctablement une accumulation d’erreurs numériques.

Ces différentes raisons ont poussé à modifier l’algorithme présenté dans le [Tableau 3.1](#). Plusieurs auteurs (([Guan et al., 2006](#)), ([Zamiri et al., 2007](#)), ([Li et al., 2008](#)), ([Rousselier et al., 2009](#)), ([Segurado et al., 2012](#)), ([Zhang et al., 2012](#))) se sont intéressés à l’implantation de la plasticité cristalline en repères tournants. L’approche mise en oeuvre ici est relative aux travaux de ([Amirkhizi et Nemat-Nasser, 2007](#)).

Dans le MECS, la loi de comportement du monocristal est incrémentalement objective et s’écrit :

$$\dot{\boldsymbol{\sigma}} = \hat{\mathbf{C}} : (\hat{\mathbf{D}} - \hat{\mathbf{D}}^p) - \hat{\boldsymbol{\sigma}} \operatorname{tr}(\hat{\mathbf{D}}) \quad (3.9)$$

Cette relation exprimée dans le CACS garde sa forme compacte :

$$\dot{\boldsymbol{\sigma}} = \tilde{\mathbf{C}} : (\tilde{\mathbf{D}} - \tilde{\mathbf{D}}^p) - \tilde{\boldsymbol{\sigma}} \operatorname{tr}(\tilde{\mathbf{D}}) \quad (3.10)$$

ce qui présente certains avantages. Le tenseur des constantes élastiques d’un cristal anisotrope s’exprime par défaut dans ce repère ; aucun transport tensoriel n’est donc requis pour tenir compte de l’anisotropie élastique. Aussi, étant donné que le CACS tourne avec le réseau cristallin, les vecteurs caractéristiques des systèmes de glissement ($\tilde{\underline{m}}^{(s)}$ et $\tilde{\underline{n}}^{(s)}$) sont invariants (si on les considère continuellement orthonormés) ou simplement étirés élastiquement. En choisissant le CACS comme repère de travail, on s’affranchit donc d’un certain nombre d’opérations numériques sur les tenseurs de Schmid. Néanmoins, cette approche impose de suivre l’évolution de l’orientation du CACS par rapport à celle du MECS (matrice \mathbf{O}) au cours de la déformation. Pour ce faire, on se base sur une distinction claire entre la rotation du réseau cristallin (dite élastique) \mathbf{R}^* et la rotation de la matière (dite plastique) par rapport au réseau cristallin \mathbf{R}^{**} ([Nemat-Nasser, 2004](#)). On peut alors définir la rotation totale de l’élément matériel comme une composition de ces deux tenseurs de rotation. Dans le contexte de la décomposition polaire du gradient de la transformation \mathbf{F} en tenseurs de rotation \mathbf{R} et d’élongation pure \mathbf{U} , on peut considérer l’égalité suivante :

$$\mathbf{R} = \mathbf{R}^* \cdot \mathbf{R}^{**} \quad (3.11)$$

Ainsi, vu que le MECS est généré par \mathbf{R} , la position du CACS se déduira de celle du MECS à une rotation plastique près. La relation d'évolution de la matrice d'orientation \mathbf{O} du cristal (donc du CACS) s'écrira

$$\dot{\mathbf{O}} = \widetilde{\mathbf{W}}^p \cdot \mathbf{O} \quad (3.12)$$

où $\widetilde{\mathbf{W}}^p$ est le spin plastique. La matrice \mathbf{O} permet de passer du CACS au MECS et inversement (voir Figure 3.2).

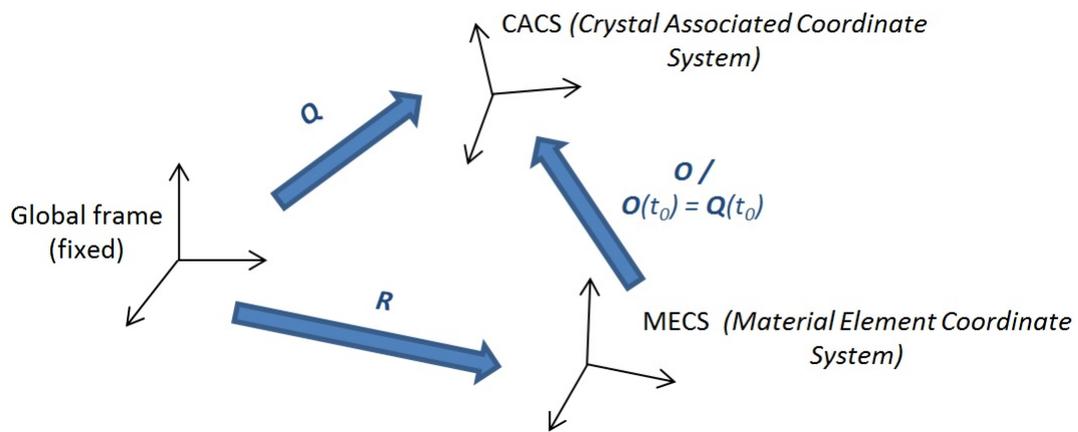


FIGURE 3.2: Différents systèmes de coordonnées mis en jeu (Adzima *et al.*, 2016).

Il est clair qu'à l'instant initial ($t = 0$), le MECS et le repère global coïncident par défaut car $\mathbf{R} = \mathbf{I}$.

Par ailleurs, le taux de déformation $\hat{\mathbf{D}}$ dans le MECS est fonction uniquement du tenseur d'élongation \mathbf{U} fourni par ABAQUS/Explicit directement exprimé dans ce repère. En partant de la décomposition polaire de \mathbf{F} et en exploitant la propriété d'orthogonalité du tenseur de rotation ($\mathbf{R} \cdot \mathbf{R}^T = \mathbf{I} \Rightarrow \dot{\mathbf{R}} \cdot \mathbf{R}^T + \mathbf{R} \dot{\mathbf{R}}^T = \mathbf{0}$) on obtient en effet :

$$\begin{aligned} \hat{\mathbf{D}} &= \mathbf{R}^T \cdot \mathbf{D}\mathbf{R} = \mathbf{R}^T \cdot \text{sym} \{ \mathbf{L} \} \mathbf{R} \\ &= \mathbf{R}^T \cdot \text{sym} \left\{ \left(\dot{\mathbf{R}} \cdot \hat{\mathbf{U}} + \mathbf{R} \cdot \dot{\hat{\mathbf{U}}} \right) \left(\mathbf{U}^{-1} \mathbf{R}^T \cdot \right) \right\} \mathbf{R} \\ &= \mathbf{R}^T \text{sym} \left\{ \mathbf{R} \dot{\hat{\mathbf{U}}} \hat{\mathbf{U}}^{-1} \mathbf{R}^T \right\} \mathbf{R} \\ &= \text{sym} \left\{ \dot{\hat{\mathbf{U}}} \hat{\mathbf{U}}^{-1} \right\} \end{aligned} \quad (3.13)$$

- Initialisation : $\tilde{\mathbf{C}}; \tilde{\mathbf{S}}_0^{s=1..12}; \tilde{\mathbf{F}}^e = \tilde{\mathbf{F}}^p = \mathbf{I}; \mathbf{Q}_0; \tau_{c_0}^{s=1..12}, \rho_0^{s=1..12}, \gamma = 0$
- Données d'entrée : $\hat{\boldsymbol{\sigma}}_n, \hat{\mathbf{U}}_{n+1}, \hat{\mathbf{U}}_n$
 1. Récupération des variables d'état
 - $\tilde{\mathbf{F}}_n^e, \mathbf{Q}_n, \tilde{\mathbf{H}}_n = \left(\tilde{\mathbf{F}}^p, \tau_c^s, \rho^s, \gamma \right)_n$
 2. Calcul des taux de déformation dans le MECS
 - $\hat{\mathbf{D}}_{n+1} = \text{sym} \left\{ \hat{\mathbf{U}} \cdot \hat{\mathbf{U}}_{n+1}^{-1} \right\}$
 3. Transport de $\hat{\boldsymbol{\sigma}}_n, \hat{\mathbf{D}}_{n+1}$ et $\hat{\mathbf{U}}_{n+1}$ du MECS vers le CACS
 - $\tilde{\boldsymbol{\sigma}}_t = \mathbf{Q}_n^T \cdot \hat{\boldsymbol{\sigma}}_t \cdot \mathbf{Q}_n$
 - $\tilde{\mathbf{D}}_{n+1} = \mathbf{Q}_n^T \cdot \hat{\mathbf{D}}_{n+1} \cdot \mathbf{Q}_n$
 - $\tilde{\mathbf{U}}_{n+1} = \mathbf{Q}_n^T \cdot \hat{\mathbf{U}}_{n+1} \cdot \mathbf{Q}_n$
 4. Actualisation de $\tilde{\mathbf{S}}^s$ (élongation) / Calcul de sa partie symétrique
 - Faire de $s=1$ à 12 $\tilde{\mathbf{S}}_n^s = \tilde{\mathbf{F}}_n^e \cdot \tilde{\mathbf{S}}_c^s \cdot \tilde{\mathbf{F}}_n^{e^{-1}}$ / $\tilde{\mathbf{M}}_n^s = \text{sym} \left\{ \tilde{\mathbf{S}}_n^s \right\}$
 5. Boucle de Runge-Kutta : faire de $i=1$ à N
 - $\tilde{\boldsymbol{\sigma}}_i = \tilde{\boldsymbol{\sigma}}_n + a_i \Delta \tilde{\boldsymbol{\sigma}}_{i-1}; \tilde{\mathbf{H}}_i = \tilde{\mathbf{H}}_n + a_i \Delta \tilde{\mathbf{H}}_{i-1}$
 - Faire de $s=1$ à 12 $\tau^s = \tilde{\boldsymbol{\sigma}}_i : \tilde{\mathbf{M}}_n^s$ / $\dot{\gamma}^s = \dot{\gamma}_0 \text{signe} \left\{ \tau^s \right\} \left| \frac{\tau^s}{\tau_{c_i}^s} \right|$
 - Calcul de $\Delta \tilde{\boldsymbol{\sigma}}_i$ et $\Delta \tilde{\mathbf{H}}_i$
 6. Mise à jour des variables
 - $\tilde{\boldsymbol{\sigma}}_{n+1} = \tilde{\boldsymbol{\sigma}}_n + \sum_{i=1}^N b_i \Delta \tilde{\boldsymbol{\sigma}}_i$
 - $\tilde{\mathbf{H}}_{n+1} = \tilde{\mathbf{H}}_n + \sum_{i=1}^N b_i \Delta \tilde{\mathbf{H}}_i$
 - $\tilde{\mathbf{F}}_{n+1}^e = \tilde{\mathbf{U}}_{n+1} \cdot \tilde{\mathbf{F}}_{n+1}^p$
 7. Actualisation de la matrice d'orientation
 - $\mathbf{O}_{n+1} = \exp \left(\Delta t \left(\tilde{\mathbf{W}}_{n+1}^p \right) \right) \cdot \mathbf{O}_n$
 8. Transport de $\tilde{\boldsymbol{\sigma}}_{n+1}$ du CACS vers le MECS
 - $\hat{\boldsymbol{\sigma}}_{n+1} = \mathbf{Q}_{n+1} \cdot \tilde{\boldsymbol{\sigma}}_{n+1} \cdot \mathbf{Q}_{n+1}^T$
- Stocker les variables d'état et retourner $\hat{\boldsymbol{\sigma}}_{n+1}$ au solveur

TABLEAU 3.3: Algorithme de la routine VUMAT en repère cristallographique (CACS).

ce qui est cohérent avec le fait que dans le MECS, seule la composante d'élongation \mathbf{U} du gradient de la transformation \mathbf{F} est prise en compte. Par conséquent, il est possible de se passer du repère global et d'effectuer une implantation en ne considérant que le MECS et le CACS.

L'algorithme proposé pour l'implantation dans le CACS est résumé dans le [Tableau 3.3](#).

3.1.1.5 Analyse comparative des implantations en repères fixe et cristallographique

Nous établissons ici une comparaison des deux algorithmes sur des simulations de trajets de chargement sur un monocristal. Il est supposé être de type CFC et provenir du cuivre. Son orientation initiale est du type Brass caractérisé par le triplet d'angles d'Euler (en convention de Bunge) suivant $(35^\circ, 45^\circ, 0^\circ)$. La vitesse de déplacement imposée est de 10^{-2} s^{-1} et les paramètres utilisés pour les simulations sont renseignés dans le [Tableau 3.4](#). La méthode de Runge-Kutta d'ordre 4 est employée dans toutes les simulations.

Les deux algorithmes sont d'abord comparés sur la réponse mécanique du monocristal au cours d'essais de traction uniaxiale et de cisaillement. Le monocristal est maillé avec un élément cubique à huit noeuds et intégration réduite (un point de Gauss) dénommé C3D8R dans ABAQUS.

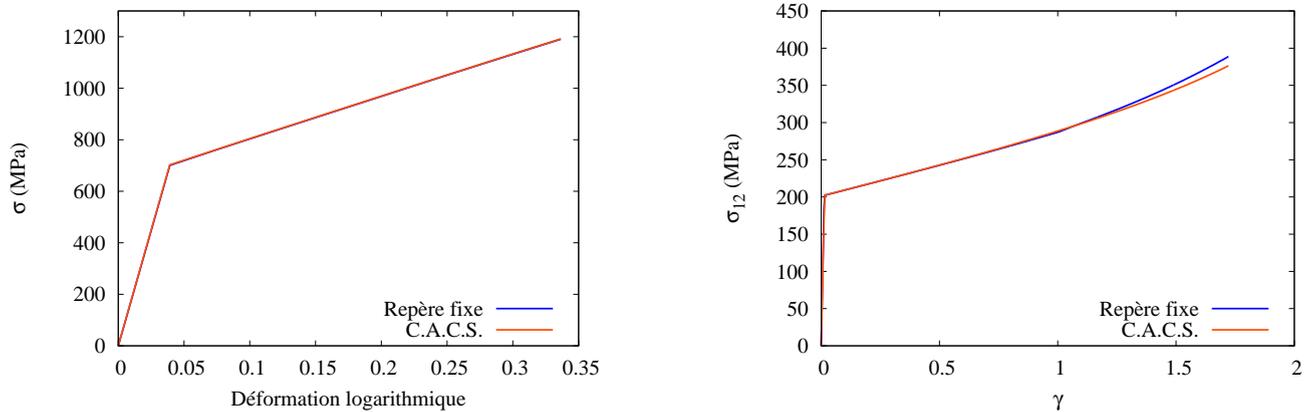
c_{11}	c_{12}	c_{44}	n	q	$\dot{\gamma}_0$	h_0	τ_0	τ^*
166 GPa	120 GPa	76 GPa	50	1,4	0,001	100	175 MPa	700 MPa

TABLEAU 3.4: Paramètres matériau utilisés dans les simulations.

Les résultats des simulations d'essais de traction et de cisaillement avec les deux algorithmes sont illustrés sur la [Figure 3.3](#).

Les deux algorithmes donnent des résultats très proches. Les divergences qui apparaissent à environ 125% de déformation en cisaillement sont inhérentes aux différences entre les cinématiques des repères objectifs de Jaumann (algorithme en repère de fixe) et de Green-Naghdi (algorithme en repère cristallographique).

Pour comparer l'efficacité des deux algorithmes, des simulations sont effectuées sur le monocristal maillé avec un nombre croissant d'éléments C3D8R. Les essais de traction uniaxiale sont simulés avec une vitesse de déformation de 10^{-2} s^{-1} jusqu'à 30% de déformation ; les calculs sont exécutés sur huit coeurs en parallèle sur des stations de travail Dell 2XQuad-core, 2.93 GHZ et 24 GB de RAM. Les champs de contraintes équivalentes de von Mises obtenus pour les modèles à 1000 éléments finis sont montrés sur la [Figure 3.4](#). Bien qu'il s'agisse de simulations sur un monocristal, pour lesquelles tous les éléments ont la même orientation initiale, les champs de contraintes obtenus après

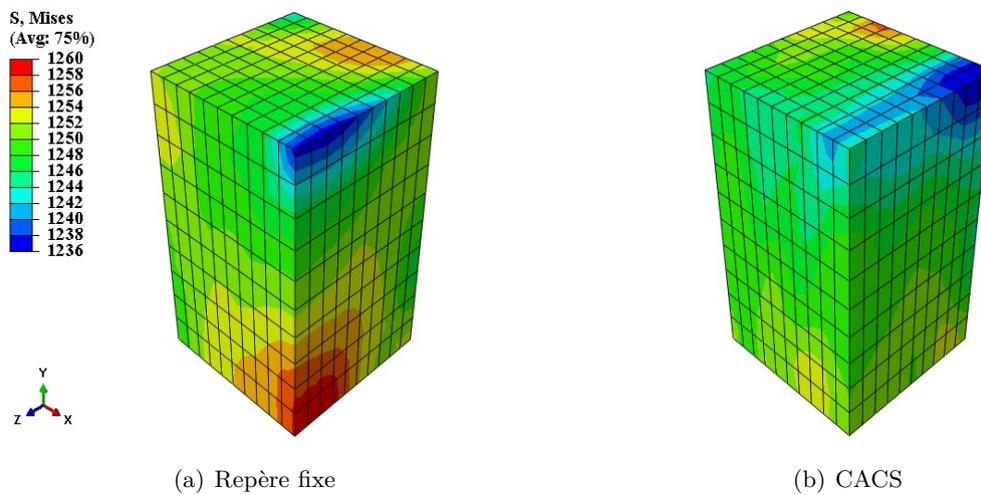


(a) Traction uniaxiale sur un monocristal

(b) Cisaillement sur un monocristal

FIGURE 3.3: Comparaison des prédictions des algorithmes en repères global et cristallographique.

déformation ne sont pas homogènes. Chaque élément se comporte comme une cristallite et l'hétérogénéité qui survient résulte de la désorientation qui se développe progressivement entre les différentes cristallites. Les écarts restent, cependant, relativement faibles avec une amplitude de 24 MPa.



(a) Repère fixe

(b) CACS

FIGURE 3.4: Champs de contraintes obtenus avec les deux algorithmes pour un essai de traction sur un monocristal maillé avec 1000 éléments C3D8R, pour une déformation logarithmique de 0.35.

On remarque que l'hétérogénéité est plus marquée sur les bords des cubes et paraît plus étendue pour l'algorithme en repère fixe.

Les temps de calcul requis par les deux algorithmes en fonction du nombre d'éléments du modèle éléments finis sont rapportés dans le [Tableau 3.5](#).

Repère d'implantation	Nombre d'éléments			
	1	1000	10642	27000
Global	5 s	2 h 14 min	6 h 42 min	24 h 33 min
CACS	5 s	2 h 15 min	3 h 08 min	5 h 57 min

TABLEAU 3.5: Temps de calcul des algorithmes pour différents maillages.

Avec l'augmentation du nombre d'éléments et donc la multiplication d'opérations au point d'intégration, l'algorithme implanté dans le CACS se révèle nettement plus efficace que celui implanté dans le repère global. Il est à noter que pour des simulations plus complexes avec du contact et un matériau polycristallin, ces temps de calcul ne font qu'augmenter. L'algorithme implanté dans le CACS constitue donc la méthode de référence utilisée dans la suite de ces travaux.

3.1.2 Implantation dans ABAQUS/Standard

Afin de pouvoir effectuer des analyses avec les solveurs dynamique-explicite et statique-implicite, une implantation de l'algorithme en repère tournant, présenté précédemment, est effectuée dans ABAQUS/Standard au travers du développement d'une UMAT. Le référentiel objectif employé dans ABAQUS/Standard étant celui de Jaumann, des adaptations ont été apportées à la cinématique et sont détaillées dans le paragraphe suivant.

3.1.2.1 ABAQUS/Standard et UMAT

Le solveur implicite ABAQUS/Standard recherche à chaque incrément de temps une solution en déplacement qui vérifie l'équilibre mécanique de la structure. En partant d'un état d'équilibre connu, une estimation des incréments de déplacement est calculée et corrigée itérativement jusqu'à ce que les conditions d'équilibre soient vérifiées (à une tolérance près). La solution vérifiant l'équilibre mécanique est donc obtenue en fin de

calcul et pour des problèmes linéaires, de grands incréments de temps sont admissibles. La routine UMAT permet de définir une loi de comportement dans ABAQUS/Standard. Pour chaque point d'intégration, elle retourne au code le tenseur de contraintes ainsi que le jacobien $\mathbf{C}_J = \frac{1}{J} \frac{\partial(\Delta\sigma)}{\partial(\Delta\epsilon)}$ où $\Delta\epsilon$ est calculé par ABAQUS à partir des déplacements nodaux.

Le repère de travail en déformations finies (MECS) est ici le référentiel corotationnel de Jaumann, généré par la rotation \mathcal{R} telle que :

$$\dot{\mathcal{R}}^T \cdot \mathcal{R} = \mathbf{W} \quad (3.14)$$

Il est possible de récupérer comme entrée de la UMAT, l'incrément de rotation $d\mathcal{R}$ du tenseur \mathcal{R} . Il est calculé par l'approximation de Hughes-Winget (Hughes et Winget, 1980) selon :

$$d\mathcal{R} = \left(\mathbf{I} - \frac{1}{2} \Delta \mathbf{W} \right)^{-1} \cdot \left(\mathbf{I} + \frac{1}{2} \Delta \mathbf{W} \right) \quad (3.15)$$

Par conséquent, pour obtenir la rotation à un instant donné, on a :

$$\mathcal{R}_{n+1} = d\mathcal{R} \cdot \mathcal{R}_n \quad (3.16)$$

Les variables cinématiques de base, à savoir le taux de déformation \mathbf{D} et le tenseur d'élongation \mathbf{U} (calculé par décomposition polaire de \mathbf{F}), peuvent donc être exprimées dans le MECS de Jaumann. En dehors de ces aspects, l'algorithme mis en place est similaire à celui employé dans ABAQUS/Explicit.

En outre, le solveur requiert la matrice jacobienne \mathbf{C}_J du modèle pour déterminer (itérativement) l'incrément de déplacement idoine. La matrice jacobienne est le module tangent de la loi de comportement cohérent avec l'algorithme d'intégration de celle-ci. Dans le cadre de l'algorithme mis en place, elle s'écrit :

$$\mathbf{C}_J = \frac{1}{J_{n+1}} \left(\mathbf{O}_{n+1} \cdot \mathbf{O}_{n+1} \cdot \frac{\partial \Delta \tilde{\sigma}}{\partial \Delta \tilde{\epsilon}} \cdot \mathbf{O}_{n+1}^T \cdot \mathbf{O}_{n+1}^T \right) \quad (3.17)$$

A partir de (3.10), on obtient en termes de composantes :

$$\frac{\partial \Delta \tilde{\sigma}_{ij}}{\partial \Delta \tilde{\epsilon}_{kl}} = \tilde{C}_{ijkl} - \tilde{\sigma}_{ij} \delta_{kl} \quad (3.18)$$

et donc l'expression finale du jacobien telle que :

$$\mathbf{C}_J = \frac{1}{J_{n+1}} \left(\mathbf{O}_{n+1} \cdot \mathbf{O}_{n+1} \cdot \left(\tilde{\mathbf{C}} - \tilde{\boldsymbol{\sigma}}_n \otimes \mathbf{I} \right) \cdot \mathbf{O}_{n+1}^T \cdot \mathbf{O}_{n+1}^T \right) \quad (3.19)$$

Le [Tableau 3.6](#) présente l'algorithme en repère cristallographique tel que implanté dans ABAQUS/Standard. Cette implantation dans la version statique-implicite du solveur sera exploitée dans le chapitre 5, notamment sur la prédiction du retour élastique.

3.2 Implantation du modèle phénoménologique

Le logiciel SiDoLo, développé par Philippe Pilvin ([Cailletaud et Pilvin, 1993](#)), est un outil dédié à l'analyse de relations de comportements de systèmes physiques. Il permet de programmer des lois de comportement, de simuler leur réponse sous diverses sollicitations et surtout de déterminer les paramètres adéquats des modèles en confrontant les résultats de simulation à des données expérimentales. SiDoLo comporte un module d'intégration de lois de comportement et un module d'identification paramétrique. Les deux modules peuvent s'utiliser ensemble ou indépendamment. Le module identification nécessite des résultats de simulation d'essais et les données expérimentales de ces essais. Les simulations peuvent alors être effectuées à l'aide de codes externes ou avec le module d'intégration (de lois de comportement) propre à SiDoLo. Dans ce dernier cas de figure qui est retenu ici, le modèle est programmé dans l'environnement de SiDoLo. Le « mode différentiel », notamment, permet d'écrire les équations différentielles de la loi de comportement sous une forme incrémentale. L'intégration numérique est alors assurée par SiDoLo par des méthodes explicites de Runge-Kutta. De plus, le dialogue entre les modules d'identification et de simulation est direct et ne nécessite aucune interface spécifique.

On donne ici une description simplifiée de l'implantation du modèle élasto-viscoplastique phénoménologique déjà présenté au chapitre 2 dans SiDoLo. Rappelons que nous considérons dans ces travaux un modèle élasto-viscoplastique anisotrope avec un écrouissage isotrope R et un écrouissage cinématique \mathbf{X} . Par rapport à la présentation générale du modèle, le terme d'écrouissage cinématique est écrit ici comme la somme de composantes de type Prager et Armstrong-Frederick soit :

- Initialisation : $\tilde{\mathbf{C}}; \tilde{\mathbf{S}}_0^{s=1..12}; \tilde{\mathbf{F}}^e = \tilde{\mathbf{F}}^p = \mathbf{I}; \mathbf{Q}_0; \tau_{c_0}^{s=1..12}, \rho_0^{s=1..12}, \gamma = 0$
- Données d'entrée : $\tilde{\boldsymbol{\sigma}}_n, \mathbf{F}_{n+1}, \mathbf{F}_n, d\mathcal{R}_n$
 1. Récupération des variables d'état
 - $\tilde{\mathbf{F}}_n^e, \mathbf{Q}_n, \tilde{\mathbf{H}}_n = \left(\tilde{\mathbf{F}}^p, \tau_c^s, \rho^s, \gamma \right)_n, \mathcal{R}_n$
 2. Actualisation de la rotation générant le MECS de Jaumann
 - $\mathcal{R}_{n+1} = d\mathcal{R}_{n+1} \cdot \mathcal{R}_n$
 3. Calcul du taux de déformation dans le repère global et décomposition polaire de \mathbf{F}_{n+1}
 - $\mathbf{D}_{n+1} = \text{sym} \left\{ \dot{\mathbf{F}} \cdot \mathbf{F}_{n+1}^{-1} \right\}$
 - $\mathbf{F}_{n+1} = \mathbf{R}_{n+1} \cdot \mathbf{U}_{n+1}$
 4. Transport du taux de déformation et du tenseur d'élongation vers le MECS
 - $\hat{\mathbf{D}}_{n+1} = \mathcal{R}_{n+1} \cdot \mathbf{D}_{n+1} \cdot \mathcal{R}_{n+1}^T$
 - $\hat{\mathbf{U}}_{n+1} = \mathcal{R}_{n+1} \cdot \mathbf{U}_{n+1} \cdot \mathcal{R}_{n+1}^T$
 5. Transport de $\tilde{\boldsymbol{\sigma}}_n, \hat{\mathbf{D}}_{n+1}$ et $\hat{\mathbf{U}}_{n+1}$ du MECS vers le CACS
 - $\tilde{\boldsymbol{\sigma}}_t = \mathbf{O}_n^T \cdot \tilde{\boldsymbol{\sigma}}_t \cdot \mathbf{O}_n$
 - $\tilde{\mathbf{D}}_{n+1} = \mathbf{O}_n^T \cdot \hat{\mathbf{D}}_{n+1} \cdot \mathbf{O}_n$
 - $\tilde{\mathbf{U}}_{n+1} = \mathbf{O}_n^T \cdot \hat{\mathbf{U}}_{n+1} \cdot \mathbf{O}_n$
 6. Actualisation de $\tilde{\mathbf{S}}^s$ (élongation) / Calcul de sa partie symétrique
 - Faire de $s=1$ à 12 $\tilde{\mathbf{S}}_n^s = \tilde{\mathbf{F}}_n^e \cdot \tilde{\mathbf{S}}_c^s \cdot \tilde{\mathbf{F}}_n^{e^{-1}} / \tilde{\mathbf{M}}_n^s = \text{sym} \left\{ \tilde{\mathbf{S}}_n^s \right\}$
 7. Boucle de Runge-Kutta : faire de $i=1$ à N
 - $\tilde{\boldsymbol{\sigma}}_i = \tilde{\boldsymbol{\sigma}}_n + a_i \Delta \tilde{\boldsymbol{\sigma}}_{i-1}; \tilde{\mathbf{H}}_i = \tilde{\mathbf{H}}_n + a_i \Delta \tilde{\mathbf{H}}_{i-1}$
 - Faire de $s=1$ à 12 $\tau^s = \tilde{\boldsymbol{\sigma}}_i : \tilde{\mathbf{M}}_n^s / \dot{\gamma}^s = \dot{\gamma}_0 \text{signe} \left\{ \tau^s \right\} \left| \frac{\tau^s}{\tau_{c_i}^s} \right|$
 - Calcul de $\Delta \tilde{\boldsymbol{\sigma}}_i$ et $\Delta \tilde{\mathbf{H}}_i$
 8. Mise à jour des variables
 - $\tilde{\boldsymbol{\sigma}}_{n+1} = \tilde{\boldsymbol{\sigma}}_n + \sum_{i=1}^N b_i \Delta \tilde{\boldsymbol{\sigma}}_i$
 - $\tilde{\mathbf{H}}_{n+1} = \tilde{\mathbf{H}}_n + \sum_{i=1}^N b_i \Delta \tilde{\mathbf{H}}_i$
 - $\tilde{\mathbf{F}}_{n+1}^e = \tilde{\mathbf{U}}_{n+1} \cdot \tilde{\mathbf{F}}_{n+1}^p$
 9. Actualisation de la matrice d'orientation (calcul de l'exponentielle par la formule de Rodrigues)
 - $\mathbf{O}_{n+1} = \exp \left(\Delta t \left(\tilde{\mathbf{W}}^p \right) \right) \cdot \mathbf{O}_n$
 10. Calcul de la matrice jacobienne \mathbf{C}_J par (3.19)
 11. Transport de $\tilde{\boldsymbol{\sigma}}_{n+1}$ du CACS vers le MECS
 - $\tilde{\boldsymbol{\sigma}}_{n+1} = \mathbf{O}_{n+1} \cdot \tilde{\boldsymbol{\sigma}}_{n+1} \cdot \mathbf{O}_{n+1}^T$
- Stocker les variables d'état et retourner $\hat{\boldsymbol{\sigma}}_{n+1}$ et le jacobien \mathbf{C}_J au solveur

TABLEAU 3.6: Algorithme de la routine UMAT en repère cristallographique (CACS).

$$\begin{aligned} \mathbf{X} &= \mathbf{X}_1 + \mathbf{X}_2 \\ \dot{\mathbf{X}}_1 &= \frac{2}{3} C_X \mathbf{D}^p - D_X \mathbf{X} \dot{p} \quad \text{et} \quad \dot{\mathbf{X}}_2 = H_X \mathbf{D}^p \end{aligned} \quad (3.20)$$

La combinaison de ces deux termes permet d'obtenir une bonne description du comportement, après inversion de la charge, aussi bien à la transition élasto-plastique qu'à de grands taux de déformations.

On désigne par \mathbf{H} le pseudo-vecteur des variables internes. Il comporte la déformation plastique $\boldsymbol{\epsilon}^p = \int_0^t \mathbf{D}^p dt$ et les composantes d'écrouissage cinématique \mathbf{X}_1 et \mathbf{X}_2 .

Les algorithmes explicites se révèlent bien adaptés à la structure modulaire de SiDoLo ; celui implanté en « mode différentiel » est présenté dans le [Tableau 3.7](#). Le repère d'expression des relations incrémentales est le référentiel corotationnel de Jaumann.

- o Données d'entrée : $\boldsymbol{\epsilon}_{n+1}^e$, $\mathbf{H}_n = (\boldsymbol{\epsilon}^p, \mathbf{X}_1, \mathbf{X}_2)_n$
 1. Calcul de $\boldsymbol{\sigma}_{n+1} = \mathbf{C} : \boldsymbol{\epsilon}_{n+1}^e$ (loi de Hooke généralisée)
 2. Calcul des variables d'écrouissage $\mathbf{X}_{n+1}(\mathbf{H}_n)$ et de $R_{n+1}(\mathbf{H}_n)$
 3. Calcul de la contrainte équivalente $\bar{\sigma}(\boldsymbol{\sigma}, \mathbf{X})_{n+1}$ et de la contrainte seuil $\sigma_y(R_{n+1})$
 4. Formulation du critère de plasticité $f = \bar{\sigma} - \sigma_y$
 - si $f \leq 0$, incrément élastique, $\Delta \mathbf{H} = 0$
 - sinon incrément élasto-plastique, calcul de $\Delta \mathbf{H}$
 5. Actualisation de la position du référentiel corotationnel et calcul de l'incrément de déformation élastique $\Delta \boldsymbol{\epsilon}^e$

TABLEAU 3.7: Algorithme du modèle phénoménologique dans SiDoLo.

Le modèle ainsi implanté dans SiDoLo peut être utilisé pour la simulation de trajets de chargement et l'identification de paramètres.

De plus, une interface entre SiDoLo et ABAQUS/Standard permet de convertir le fichier SiDoLo en UMAT apte à l'analyse par éléments finis.

3.3 Génération de microstructure numérique

Le comportement du matériau étant désormais décrit (au point d'intégration) à l'échelle du monocristal, on s'intéresse maintenant au passage à l'échelle des polycristaux afin d'envisager in fine le calcul de structures à l'échelle macroscopique avec de la plasticité cristalline. Cette transition d'échelles est assurée ici par une méthode à champs complets dite d'homogénéisation par éléments finis. Cette technique nécessite une représentation numérique de la microstructure c'est-à-dire une modélisation géométrique de la forme des grains et de leur arrangement spatial ainsi que le maillage éléments finis associé. On distingue généralement trois classes de méthodes de représentation des grains qui sont détaillées par la suite.

3.3.1 L'utilisation de formes simplifiées

Les grains sont modélisés par des formes géométriques régulières à même de paver l'espace. On emploie à cet effet des volumes simples tels que le cube, le tétraèdre, le dodécaèdre et l'octaèdre tronqué. Ces volumes peuvent tous être maillés avec des éléments hexaédriques ([Figure 3.5\(a\)](#)). Si les formes cubiques et tétraédriques offrent des facilités de manipulation géométrique, elles conduisent généralement à des modélisations peu réalistes de la forme des grains. D'un autre côté, les volumes dodécaédriques et octaédriques permettent de bien représenter des grains plus ou moins sphériques; des difficultés surviennent néanmoins sur les agrégats ainsi générés en raison des formes complexes que prennent les frontières extérieures comme illustré sur la [Figure 3.5\(b\)](#). Ce type d'empilement ne peut remplir parfaitement un volume parallélépipédique en raison des surfaces non planes obtenues.

L'inconvénient principal de ce type de modélisation reste l'uniformité de morphologie des grains obtenus, ce qui constitue une approximation forte de la microstructure réelle des matériaux métalliques.

3.3.2 Les tessellations de Voronoï

Employée par de nombreux auteurs (entre autres ([Barbe et al., 2001](#)), ([Wang et al., 2009](#)), ([Li et al., 2014](#))), cette technique consiste à partitionner l'espace en polyèdres

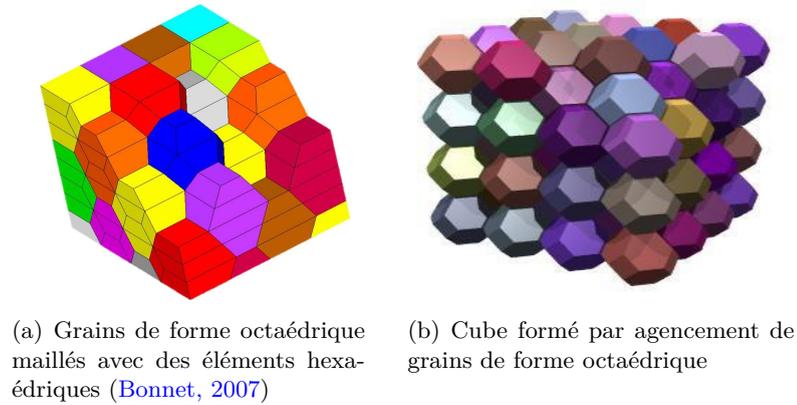


FIGURE 3.5: Représentation de grains par des octaédres.

obtenus par tessellation de Voronoï qui représentent les grains . La tessellation de Voronoï consiste à répartir des germes (ou générateurs) dans un domaine et à construire des cellules basées sur la distance minimum à chaque germe. Synthétiquement, une cellule de Voronoï représente l'ensemble des points les plus proches d'un germe. Les frontières de la cellule sont les médiatrices des segments reliant deux germes. De manière formelle, si on considère un domaine de l'espace Ω dans lequel sont répartis des germes $\{g_i\}_{i=1..n}$, les cellules de Voronoï sont les domaines $\{V_i\}_{i=1..n}$ construits selon :

$$V_i = \{x \in \Omega \mid \|x - g_i\| \leq \|x - g_j\|\} \text{ pour } j = 1..n \text{ et } j \neq i \quad (3.21)$$

et qui satisfont les conditions :

$$V_i \cap V_j = \{0\} \text{ et } \cup V_i = \Omega. \quad (3.22)$$

La Figure 3.6 illustre les étapes de l'algorithme permettant d'obtenir les cellules de Voronoï en deux dimensions.

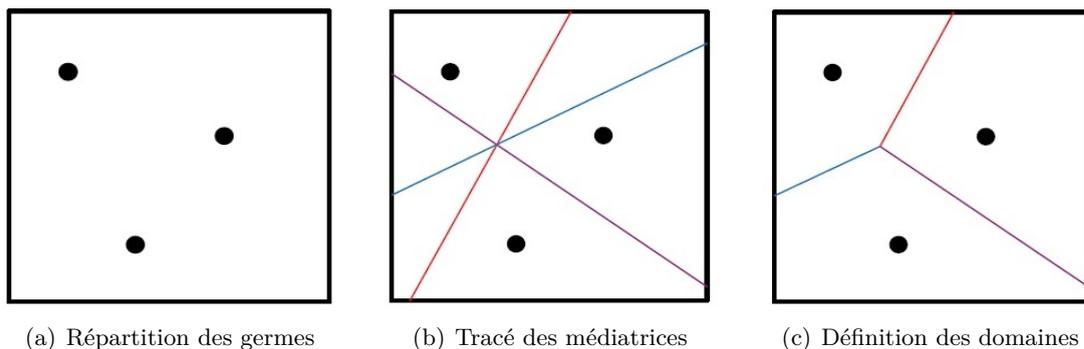


FIGURE 3.6: Principe de fonctionnement en 2D d'une tessellation de Voronoï.

Les grains ainsi obtenus sont ensuite maillés par des techniques spécifiques. Un maillage régulier permet d'utiliser des éléments hexaédriques tandis qu'un maillage libre effectué avec des éléments tétraédriques conduit à une meilleure représentation des interfaces des grains (voir [Figure 3.7](#)).

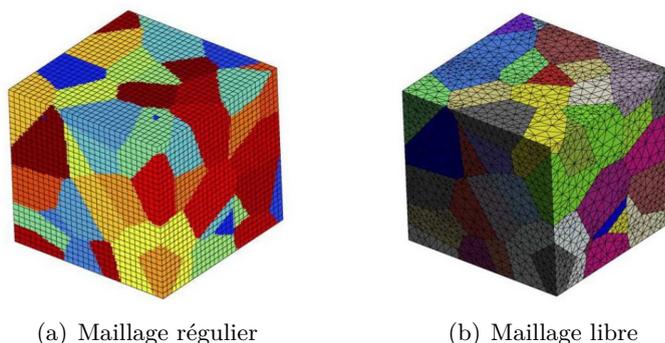


FIGURE 3.7: Techniques de maillage d'un polycristal.

3.3.3 La modélisation par croissance de sphère

Le principe consiste à répartir des germes dans un maillage éléments finis pré-existant et d'y faire croître des sphères (([Anderson et al., 1989](#)), ([Humphreys, 1997](#))). L'affectation des éléments du maillage pré-existant aux sphères se fait au fur et à mesure qu'un point d'intégration de l'élément est englobé dans la sphère. L'opération se poursuit jusqu'à ce que tous les éléments du maillage aient été affectés. Les regroupements d'éléments ainsi formés représentent les grains. Ce genre de modèle est inspiré du processus physique de nucléation qui conduit à la formation de polycristaux. Le logiciel *Germination* développé par ([Bonnet, 2007](#)) utilise ce type de modèles. Son principe de fonctionnement est illustré en deux dimensions sur la [Figure 3.8](#).

Dans cette thèse, les deux techniques précédentes sont utilisées de façon complémentaire à savoir les tessellations de Voronoï au travers du logiciel *Neper* ([Quey et al., 2011](#)) et la croissance de sphères avec le logiciel *Germination*. Les deux logiciels sont présentés dans les paragraphes suivants.

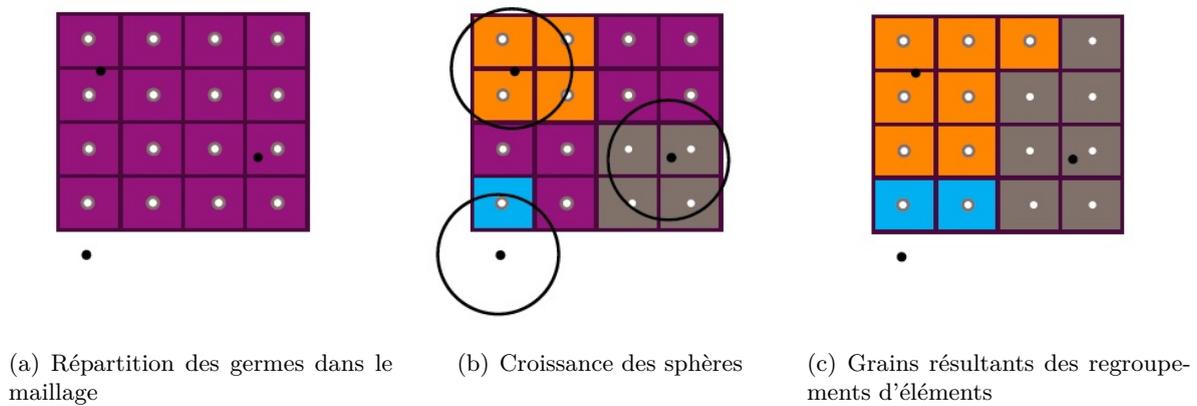


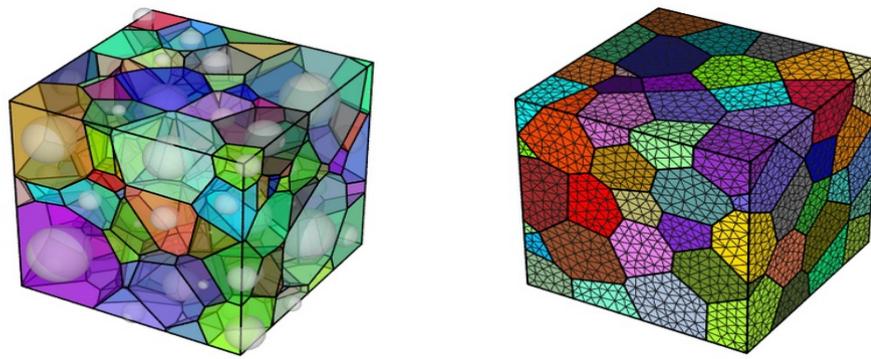
FIGURE 3.8: Principe de fonctionnement de *Germination*. Les carrés représentent des éléments finis avec en leur centre un point d'intégration. Les points noirs sont les germes ou générateurs à partir desquels croissent les sphères qui vont regrouper des éléments et former des grains représentés par des couleurs différentes.

3.3.4 Le logiciel *Neper*

Neper (Quey *et al.*, 2011) est un logiciel libre qui permet de générer et de mailler des polycristaux en deux ou trois dimensions. Les grains sont d'abord générés par tessellations de Voronoï et ensuite maillés à l'aide de bibliothèques dédiées (Netgen et Gmsh). *Neper* offre la possibilité d'utiliser diverses variantes de la procédure de Voronoï. Dans une tessellation centroïde de Voronoï, par exemple, les germes sont distribués de manière à ce qu'ils soient les centres de masse des grains obtenus ; cette procédure génère des grains à forme plus équiaxe que la tessellation de Voronoï standard. La tessellation de Laguerre-Voronoï (voir Figure 3.9(a)) permet d'attribuer des poids initiaux aux germes ; on peut ainsi reproduire une distribution de taille de grains en utilisant, entre autres, une distribution gaussienne de poids avec une moyenne et un écart type. La procédure de maillage est également flexible et offre de nombreuses possibilités ; on peut notamment choisir de générer un maillage régulier avec des éléments hexaédriques ou opter pour un maillage libre comme sur la Figure 3.9(b).

Neper permet également de générer des microstructures périodiques qui s'avèrent pertinentes pour la mise au point de volumes élémentaires représentatifs.

Cependant, en dépit de son efficacité et de sa flexibilité, ce logiciel présente l'inconvénient de ne pouvoir mailler que des domaines de l'espace aux formes géométriques simples à base de cubes ou des cylindres. Si ces géométries sont pertinentes pour des investigations « académiques », elles ne sont pas adaptées pour des analyses de pièces industrielles aux formes complexes. Dans ce cas de figure, nous aurons recours au logiciel *Germination*.



(a) Tessellation de Laguerre-Voronoi

(b) Maillage régulier

FIGURE 3.9: Polycristaux et maillages générés par *Neper* (Quey *et al.*, 2011).

3.3.5 Le logiciel *Germination*

Germination a été développé au cours de la thèse de Nicolas Bonnet (Bonnet, 2007). Ce logiciel utilise la méthode de croissance de sphères dans un maillage prédéfini et permet donc de représenter une microstructure sur n'importe quel type de géométrie souhaitée. Les germes sont répartis (aléatoirement ou à des positions spécifiques) dans un parallélépipède qui englobe le maillage de la pièce. Les rayons des sphères sont ensuite augmentés incrémentalement à partir des germes ; dès qu'un point d'intégration d'un élément entre dans une sphère, l'élément est affecté au grain en cours de création. A chaque incrément de calcul, le rayon r des sphères est augmenté de :

$$\Delta r = \frac{1}{V_{cr}} d \quad (3.23)$$

où V_{cr} , une entrée du logiciel, est la « vitesse » de croissance (supérieure à 1) et d , la distance minimale entre deux centres d'éléments, est calculée lors de l'importation du maillage.

La Figure 3.10 montre un exemple de microstructure générée par *Germination* sur une géométrie d'éprouvette de traction.

Nos travaux ont conduit à de nouveaux développements dans le logiciel et à l'ajout de fonctionnalités permettant de représenter des textures particulières. Par exemple, les tôles métalliques ayant subi un laminage important présentent souvent une microstructure relativement lamellaire avec des grains fortement étirés dans la direction de laminage comme illustré ci-dessous.



FIGURE 3.10: Microstructure générée par *Germination*. Les couleurs servent uniquement à distinguer les grains les uns des autres.

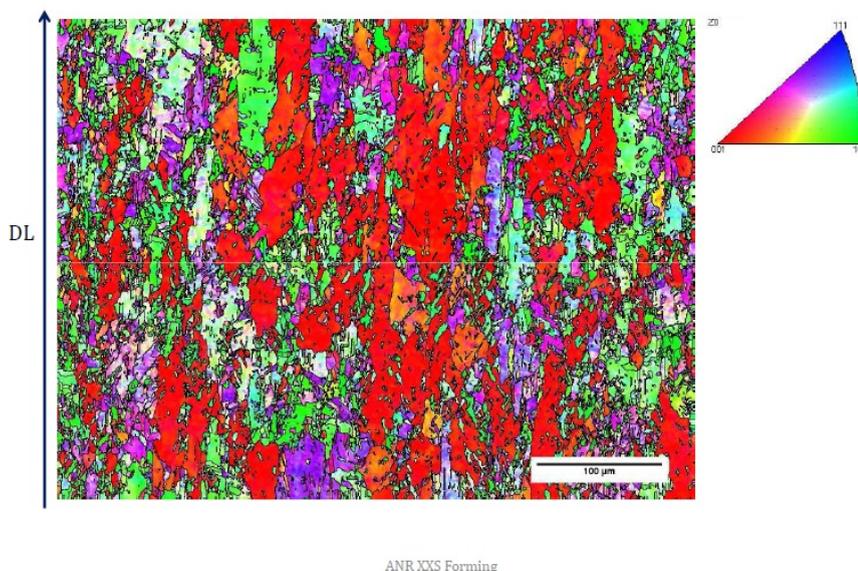


FIGURE 3.11: Analyse EBSD de l'alliage CuFe2P (Caer et Pesci, 2014).

La Figure 3.11 est une cartographie de la microstructure d'un des alliages de cuivre de l'étude dans le cadre du projet, issue d'une analyse par EBSD. Le challenge consiste à obtenir une représentation numérique de ce type de texture. Pour ce faire, une procédure a été mise en place dans *Germination* qui consiste à définir trois facteurs d'homothétie (un pour chaque dimension de l'espace) par le biais desquels on altère le maillage de la pièce lors de l'importation, avant l'étape de distribution des germes et de croissance des sphères. Les grains sont donc générés sur le maillage aux dimensions altérées, et une dernière étape est nécessaire pour ramener le maillage et les grains aux dimensions d'origine de la pièce. On peut ainsi obtenir le type de microstructure recherché comme l'illustre, pour l'exemple considéré, la Figure 3.12.

Les techniques expérimentales de caractérisation microstructurale donnent aussi accès aux tailles de grains, ce qui permet de déterminer une taille moyenne des grains ainsi que la dispersion correspondante. Afin d'en établir une représentation numérique dans *Germination*, une procédure développée sous la forme d'un script Python récupère les



FIGURE 3.12: Microstructure de grains fortement étirés générée par *Germination*.

données expérimentales et calcule la taille moyenne de grains et l'écart type par rapport à l'échantillon considéré. Une distribution lognormale des diamètres de grains d_g associée aux deux paramètres est alors générée comme fichier d'entrée à *Germination*. L'incrément sur les rayons des sphères (3.23) n'est plus uniforme mais spécifique à chaque germe g et modifié comme suit :

$$\Delta r_g = \frac{k}{V_{cr}} d_g \quad (3.24)$$

où $0 < k \leq 1$ est un facteur permettant de diminuer ou d'augmenter la taille des incréments et d'avoir un calcul plus ou moins long et fin au regard de la morphologie des grains souhaités.

Un exemple d'utilisation de cette fonctionnalité est illustré sur la Figure 3.13, qui montre une dispersion relativement importante sur les grains par rapport à la Figure 3.10 où la taille de grains est plus homogène.



FIGURE 3.13: Microstructure de tôle avec une dispersion marquée sur la taille de grains générée par *Germination*.

3.4 Conclusion

Les développements nécessaires pour la mise en oeuvre numérique des deux lois de comportement retenues dans ces travaux ont été exposés. Le modèle de plasticité cristalline

a été implanté dans ABAQUS au point d'intégration alors que le modèle phénoménologique a été implanté à la fois dans ABAQUS et dans SiDoLo. La transition d'échelle du monocristal à l'échelle de la structure est réalisée par éléments finis (CPFEM), ce qui a nécessité le recours à des techniques de génération de maillages représentatifs de la microstructure des matériaux. Le logiciel libre *Neper* a été retenu pour les géométries simples, en complément du logiciel *Germination* pour les géométries plus complexes.

Le cadre de la modélisation étant posé, le chapitre suivant est consacré à la caractérisation des matériaux. Ces données expérimentales sont essentielles pour calibrer les modèles choisis et les outils numériques d'investigation des procédés de micro-formage.