
CALCUL DU GRADIENT DES FONCTIONS COÛTS PAR LA METHODE DE L'ETAT ADJOINT – EXTENSION AU FORGEAGE MULTI-PASSES

III.1. INTRODUCTION

Une manière de résoudre efficacement un problème d'optimisation consiste à utiliser le gradient des fonctions coûts par rapport aux paramètres à optimiser et à recourir à des algorithmes à base de gradient. Cependant, le calcul du gradient est complexe dans le cadre de la simulation du forgeage 3D par des logiciels éléments finis, notamment celui par rapport aux paramètres de forme. Les verrous principaux de ce problème ont été levés dans [Laroussi 2003]. Dans son travail, Laroussi a implémenté une première version de la méthode de l'état adjoint qui permet de calculer le gradient de certaines fonctions coûts par rapport aux paramètres de type "forme du lopin initial".

Cependant, ce type de paramètres est lié par la contrainte de volume constant, qui n'est pas facile à traiter. De plus, le calcul du gradient des fonctions coûts par rapport au paramètre "forme du lopin initial" est "spécifique" pour chaque problème. Nous ne pouvons pas utiliser ce calcul d'un cas à l'autre. Eviter la contrainte de volume constant et s'affranchir les spécificités du calcul du gradient est donc une tâche envisagée dans ce contexte. Etendre ce calcul aux problèmes de forgeage multi-passes est donc une bonne solution pour achever cette tâche. Cette extension constitue une contribution importante de cette thèse et fait l'objet de ce chapitre.

Dans un premier temps (paragraphe III.2), ce chapitre décrit le problème mécanique du forgeage résolu dans le code éléments finis FORGE3® (développé au CEMEF). Il présente différentes fonctions coûts retenues dans le cadre de ce travail, *avec une amélioration apportée sur la fonction coût pour mieux détecter le défaut de repli*. Il rappelle aussi la méthode de l'Etat Adjoint implémentée dans ce code pour calculer le gradient des fonctions coûts.

Dans un second temps (paragraphe III.3), il expose *le cœur de la contribution apportée avec une extension du travail de Laroussi [Laroussi 2003] sur le calcul du gradient pour le forgeage multi-passes pour l'optimisation de formes des outils*. Il présente aussi *la validation de ce calcul du gradient*, en comparant les valeurs obtenues par la méthode de l'Etat Adjoint avec celles obtenues par la méthode des différences finies.

Enfin, ce chapitre se termine par un bilan des améliorations apportées.

III.2. CALCUL DU GRADIENT DE LA FONCTION COUT PAR LA METHODE DE L'ETAT ADJOINT DANS LE LOGICIEL FORGE3®

III.2.1. Expression du problème mécanique de forgeage et des équations implémentées dans le logiciel FORGE3®

Cette première partie présente la modélisation des procédés de mise en forme par forgeage utilisée dans le code éléments finis FORGE3®. Nous formulerons tout d'abord le problème mécanique de forgeage comme un problème continu avec les équations de conservation, les lois de comportement et les conditions aux limites. Ensuite, nous décrirons le problème discrétisé issu du problème continu précédent, en introduisant la formulation éléments finis. Puis, nous présenterons la gestion du contact et les méthodes de résolution utilisées. Enfin, quelques remarques seront faites sur le remaillage et le transport des variables.

III.2.1.1. Formulation du problème continu

Le problème mécanique de la mise en forme par forgeage est classique, c'est pourquoi nous nous limiterons ici à exposer les équations de base dans un cadre isotherme. Le lecteur intéressé pourra se référer à [Cescutti 1989] pour plus de détails sur la formulation mécanique et à [Soyris 1990] pour l'aspect anisotherme.

Mettons nous à l'échelle macroscopique et considérons le cas du forgeage à chaud. Le lopin à forger est considéré comme un domaine continu et homogène, Ω de frontière $\partial\Omega$. Pour la modélisation du problème, on choisit d'exprimer les équations en tout point de Ω en fonction du champ de vitesse v et du champ de pression p . Ceci conduit à l'utilisation d'une formulation à deux champs (vitesse et pression). Afin de ne pas alourdir les équations, dans le cadre du forgeage des métaux à chaud, les hypothèses suivantes sont posées :

- les actions de l'inertie et de la gravité sont négligeables devant celles des déformations plastiques.
- la composante élastique de la déformation est négligeable par rapport à celle plastique, le matériau est donc supposé purement viscoplastique.
- le matériau forgé est donc incompressible.

III.2.1.1.1. Equations d'équilibre

L'écoulement de la matière lors de la mise en forme doit vérifier les principes fondamentaux de la mécanique des milieux continus. Avec les hypothèses précédentes, ces principes s'écrivent mathématiquement comme suit :

$$\text{- l'équilibre dynamique} \quad \text{div} \sigma = 0 \quad (3.1)$$

$$\text{- l'incompressibilité} \quad \text{div} v = 0 \quad (3.2)$$

où σ est le tenseur des contraintes de Cauchy décomposé en : $\sigma = s - pI$

s est le déviateur des contraintes, I le tenseur identité et p la pression hydrostatique définie par $p = -\frac{1}{3}Trace(\sigma)$

III.2.1.1.2. Conditions aux limites

Pour considérer les conditions aux limites, on décompose la surface de la pièce forgée (ou du domaine Ω) $\partial\Omega$ en deux parties distinctes (voir *Figure 3.1* pour une représentation en 2D) :

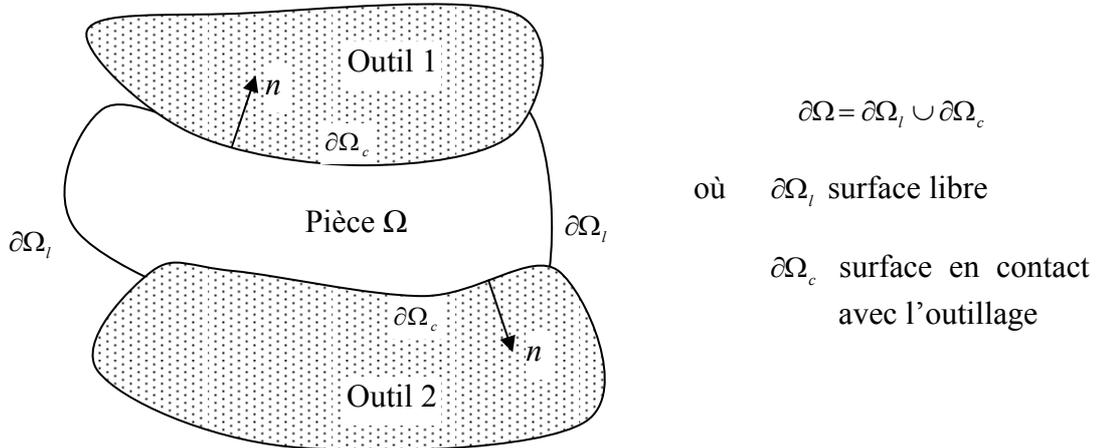


Figure 3.1. Définition des surfaces et de la normale à la pièce

Sur la surface libre $\partial\Omega_l$, la pièce n'est soumise à aucune contrainte, ce qui est traduit par :

$$\sigma n = 0 \quad (3.3)$$

avec n la normale sortante à la pièce (voir *Figure 3.1*)

Sur la surface en contact avec l'outillage $\partial\Omega_c$, deux types de conditions sont imposés:

- une condition tangentielle (condition de frottement) qui permet d'exprimer la valeur de la cission de frottement τ définie par :

$$\tau = \sigma n - (\sigma n \cdot n)n \quad (3.4)$$

Cette condition est associée à une loi tribologique qui sera présentée plus loin.

- une condition de contact unilatéral ou de Signorini, qui exprime la non pénétration de la pièce dans l'outillage. Elle est décrite par les équations suivantes :

$$\begin{cases} (v - v_{outil}) \cdot n \leq 0 \\ \sigma_n \leq 0 \\ (v - v_{outil}) \cdot n \sigma_n = 0 \end{cases} \quad (3.5)$$

où v_{outil} est la vitesse de l'outil et $\sigma_n = \sigma n \cdot n$ est la pression normale de contact. La condition de contact est portée par la normale à la surface de la pièce.

III.2.1.1.3. Lois de comportement

↳ Loi rhéologique

Pour modéliser l'écoulement du matériau, on utilise une loi de comportement de type Norton-Hoff. Cette loi s'écrit :

$$s = 2K \left(\sqrt{3} \dot{\varepsilon} \right)^{m-1} \dot{\varepsilon} \quad (3.6)$$

où K est la consistance du matériau, m la sensibilité à la vitesse de déformation, $\dot{\varepsilon}$ le tenseur des vitesses de déformation calculé par : $\dot{\varepsilon} = \frac{1}{2} (\nabla v + {}^T \nabla v)$, $\dot{\varepsilon}$ la vitesse de déformation

généralisée donnée par : $\dot{\varepsilon} = \sqrt{\frac{2}{3} \dot{\varepsilon} : \dot{\varepsilon}}$

La consistance K du matériau peut-être fonction de la déformation généralisée $\bar{\varepsilon}$ et de la température. Cependant, on se place ici dans le cadre isotherme et on considère que le matériau ne présente pas d'écrouissage. Pour plus de détails sur les équations de comportement, le lecteur peut se référer à [Rappaz et al. 1998] et [Aliaga 2000].

↳ Loi de frottement

La loi tribologique de type Norton est utilisée pour traduire l'action du frottement. Cette loi est bien adaptée aux fortes pressions de contact caractérisant les procédés de forgeage. Elle s'écrit comme suit :

$$\tau = -\alpha_f K \left\| \Delta v_g \right\|^{p_f-1} \Delta v_g \quad (3.7)$$

où Δv_g est la vitesse de glissement de l'outil : $\Delta v_g = v - v_{outil} - [(v - v_{outil}) \cdot n] n$ (3.8)

α_f est le coefficient de frottement et p_f le coefficient de sensibilité à la vitesse de glissement. Souvent, on choisit $p_f = m$.

III.2.1.1.4. Système d'équations à résoudre

En rassemblant les équations précédentes avec la décomposition de σ en fonction de s et p , on aboutit au système d'équations suivant :

$$\begin{cases} \operatorname{div} s - \operatorname{grad} p = 0 & \text{sur } \Omega \\ \operatorname{div}(v) = 0 & \text{sur } \Omega \\ \tau = -\alpha_f K \left\| \Delta v_g \right\|^{m-1} \Delta v_g & \text{sur } \partial\Omega_c \\ (v - v_{outil}) \cdot n \leq 0 \quad \text{sur } \partial\Omega_c \quad \text{et} \quad \sigma n = 0 \quad \text{sur } \partial\Omega_i \end{cases} \quad (3.9)$$

III.2.1.1.5. Forme faible du problème continu:

Pour obtenir la formulation faible du problème, on multiplie les équations locales (3.9) par des fonctions tests et les intègre. Pour l'instant, afin de ne pas alourdir la présentation avec les termes de contact, on introduit les contraintes que doivent vérifier le champ de vitesse dans l'espace des fonctions tests. Enfin, les fonctions tests qu'on utilise dans les formulations qui suivent évoluent dans les espaces V_0^{ca} et P définis par :

$$\begin{aligned} V^{ca} &= \left\{ v \in (H^1(\Omega))^3; \operatorname{div} v = 0 \text{ sur } \Omega; (v - v_{outil}) \cdot n \leq 0 \text{ sur } \partial\Omega_c \right\} \\ V_0^{ca} &= \left\{ v \in (H^1(\Omega))^3; \operatorname{div} v = 0 \text{ sur } \Omega; v \cdot n \leq 0 \text{ sur } \partial\Omega_c \right\} \\ P &= L^2(\Omega) \end{aligned}$$

Le Principe des Puissances Virtuelles nous permet d'écrire la formulation faible à partir du problème fort (3.9) comme suit :

Trouver $(v, p) \in (V^{ca}, P)$ tel que $\forall (v^*, p^*) \in (V_0^{ca}, P)$

$$\begin{cases} \int_{\Omega} s : \dot{\varepsilon}(v^*) d\Omega - \int_{\Omega} p \operatorname{div}(v^*) d\Omega - \int_{\partial\Omega_c} \tau \cdot v^* dS = 0 \\ \int_{\Omega} p^* \operatorname{div}(v) d\Omega = 0 \end{cases} \quad (3.10)$$

Une formulation plus détaillée avec les démonstrations de l'existence et de l'unicité de la solution est présentée dans [Traoré 2001] par exemple.

En vue d'une résolution par la méthode des éléments finis, il est indispensable d'écrire le système (3.10) sous sa forme discrétisée. Cela sera l'objectif du prochain paragraphe.

III.2.1.2. Formulation du problème discrétisé

La discrétisation du problème précédent est réalisé en temps et en espace.

III.2.1.2.1. Discrétisation en temps

L'intervalle de temps du procédé $[t_0, t_{fin}]$, où t_0 et t_{fin} désignent respectivement le temps initial et final de la simulation de forgeage, est discrétisé en plusieurs pas de temps. Le logiciel FORGE3® utilise le schéma d'intégration temporel de type Euler explicite comme suit:

$$X^{t+\Delta t} = X^t + \Delta t V^t \quad (3.11)$$

L'utilisation d'un tel schéma revient à faire l'hypothèse que la vitesse est constante sur l'incrément temps Δt .

La discrétisation temporelle est adoptée. Il faut maintenant choisir un type d'éléments finis adéquat pour la discrétisation spatiale.

III.2.1.2.2. Discrétisation en espace

La discrétisation spatiale consiste à approximer les espaces d'admissibilité de la vitesse V^{ca} et de la pression P par des espaces discrets de dimensions finies V_h^{ca} et P_h . Nous souhaitons bien évidemment que les fonctions (v_h, p_h) convergent vers les solutions du problème continu (v, p) . Pour obtenir cette convergence, le problème doit être consistant, c'est-à-dire que l'espace d'approximation doit tend vers l'espace continu quand la taille de maille h tend vers zéro. Ce sera le cas puisque les espaces d'approximation avec lesquels nous travaillerons sont des espaces de polynômes.

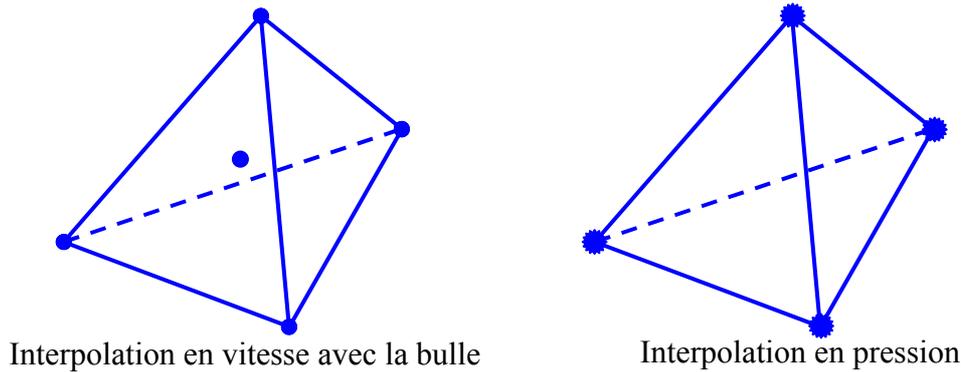


Figure 3.2. Elément fini mixte P1+/P1

Pour coupler les champs de vitesse et de pression, la condition de consistence ne suffit plus, il faut lui adjoindre une condition de stabilité spatiale appelée condition de Brezzi-Babūška [Brezzi et al. 1991]. Cette condition traduit le fait que les espaces V_h^{ca} et P_h ne peuvent pas être pris indépendamment. Compte tenu cette condition, l'élément fini mixte tétraédrique P1+/P1 [Arnold et al. 1984] a été utilisé et implémenté dans FORGE3® [Coupez 1995], pour une formulation viscoplastique incompressible. L'idée est d'enrichir l'espace des vitesses par rapport à la pression en introduisant une "bulle" au centre de l'élément. Le champ de vitesse est décomposé en une partie linéaire continue et en une partie linéaire bulle discontinue sur le tétraèdre de base. La contribution "bulle" étant linéaire et continue sur les quatre sous-tétraèdres définis par les 4 sommets et le centre de gravité du tétraèdre (voir Figure 3.2). La pression, quant à elle, est linéaire sur le tétraèdre.

$$\begin{aligned}
 P_h &= \left\{ p_h \in C^0(\Omega_h); \quad \forall \omega_h \in \Gamma_h, \quad p_h|_{\omega_h} \in P^1(\omega_h) \right\} \\
 V_h &= L_h \oplus B_h \\
 L_h &= \left\{ v_h \in (C^0(\Omega_h))^3; \quad \forall \omega_h \in \Gamma_h, \quad v_h|_{\omega_h} \in (P^1(\omega_h))^3 \right\} \\
 B_h &= \left\{ b_h \in (C^0(\Omega_h))^3; \quad \forall \omega_h \in \Gamma_h, \quad b_h|_{\omega_h^i} \in (P^1(\omega_h^i))^3, \quad i=1, \dots, 4 \text{ et } b_h = 0 \text{ sur } \partial\omega_h \right\}
 \end{aligned}$$

où les $(\omega_h^i)_{i=1, \dots, 4}$ désignent les quatre sous-tétraèdres de ω_h ; Γ_h est l'ensemble des éléments finis du maillage

Notons N^l et N^b respectivement les fonctions de base linéaires et bulle. Les champs de vitesse et de pression discrets s'écrivent :

$$\begin{aligned} \mathbf{v}_h &= \mathbf{v}_h^l + \mathbf{v}_h^b = \sum_{k=1}^{nbnoe} V_k N_k^l + \sum_{j=1}^{nbelt} V_j N_j^b \\ p_h &= \sum_{k=1}^{nbnoe} P_k N_k^l \end{aligned} \quad (12)$$

où $nbnoe$ et $nbelt$ représentent respectivement le nombre de nœuds et le nombre d'éléments du maillage.

$$\left\{ \begin{array}{l} \forall k = 1, \dots, nbnoe; \forall i = 1, \dots, 3 \\ \int_{\Omega_h} 2K(\sqrt{3}\dot{\varepsilon}_h)^{m-1} \dot{\varepsilon}_h(\mathbf{v}_h^l + \mathbf{v}_h^b) : \dot{\varepsilon}_h(N_k^l e_i) d\Omega_h + \\ \int_{\partial\Omega_{ch}} \alpha_f K \|\Delta v_{gh}\|^{m-1} (\Delta v_{gh} \cdot e_i) N_k dS_h - \int_{\Omega_h} p_h \operatorname{div}(N_k^l e_i) d\Omega_h = 0 \\ \forall e = 1, \dots, nbelt; \forall i = 1, \dots, 3 \\ \int_{\Omega_h^e} 2K(\sqrt{3}\dot{\varepsilon}_h)^{m-1} \dot{\varepsilon}_h(\mathbf{v}_h^l + \mathbf{v}_h^b) : \dot{\varepsilon}_h(N_e^b e_i) d\Omega_h - \int_{\Omega_h^e} p_h \operatorname{div}(N_e^b e_i) d\Omega_h = 0 \\ \forall k = 1, \dots, nbnoe \quad \int_{\Omega_h} N_k^l \operatorname{div}(\mathbf{v}_h^l + \mathbf{v}_h^b) d\Omega_h = 0 \end{array} \right. \quad (3.13)$$

où e_i est le vecteur unitaire dans la direction i de l'espace.

En remplaçant les fonctions tests dans (3.10) par les fonctions de base éléments finis, nous obtenons la formulation faible discrétisée (3.13) de notre problème (à laquelle manque les équations de contact). Cette formulation est donc valable pour un champ cinématiquement admissible qui vérifie les équations de contact. Les équations de contact sont présentées dans le paragraphe suivant.

III.2.1.3. Gestion du contact

En ce qui concerne la gestion du contact, nous nous limiterons à une description très générale, un développement plus complet se trouvant dans [Mocellin 1999].

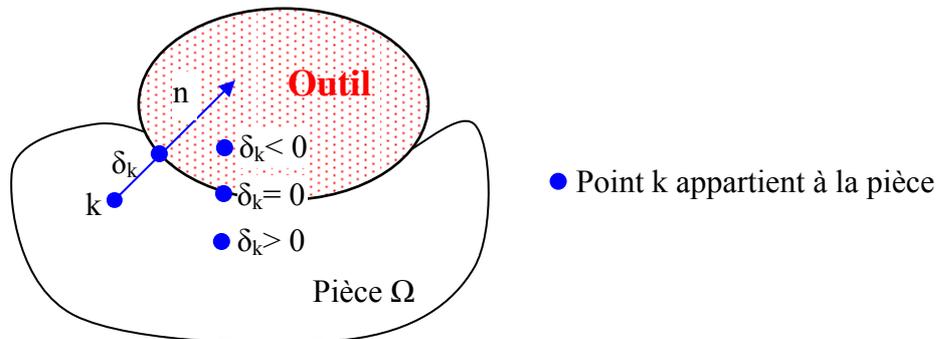


Figure 3.3. Définition de la distance de contact δ

En tout point du domaine Ω décrivant la pièce, notons δ la distance de ce point à la surface de l'outil. La *Figure 3.3* décrit la convention de distance qui définit δ par :

- $\delta_k = 0$ si le point k appartient à la surface de l'outil
- $\delta_k < 0$ si le point k a pénétré dans l'outil
- $\delta_k > 0$ si k est à l'extérieur de l'outil

La condition de non-pénétration que nous voulons imposer sur la surface revient à écrire :

$$\delta \geq 0 \quad \text{sur } \partial\Omega \quad (3.14)$$

Au cours des incréments de temps, les points de $\partial\Omega$ potentiellement en contact avec l'outil ne doivent pas pénétrer dans celui-ci. On autorise ces points, soit à rester en contact avec l'outil, soit à quitter le contact dans le sens des $\delta > 0$. Pour cela, avec un développement limité à l'ordre 1 en temps, la condition suivante est imposée :

$$\delta^{t+\Delta t} = \delta^t + (v_{\text{outil}} - v) \cdot n^t \Delta t + O(\Delta t^2) \geq 0 \quad (3.15)$$

Cette condition de contact unilatéral peut encore être exprimée à l'ordre 1 en temps par :

$$h(v) = (v - v_{\text{outil}}) \cdot n^t - \frac{\delta^t}{\Delta t} \leq 0 \quad (3.16)$$

Au niveau discret, l'équation (3.16) s'écrit :

$$\forall k \in \partial\Omega_h, \quad h(V_k) = (V_k - v_{\text{outil}}) \cdot n_k - \frac{\delta_k}{\Delta t} \leq 0 \quad (3.17)$$

Dans le logiciel FORGE3®, le contact est traité en chaque nœud de la surface du domaine par une méthode de pénalisation décrite de manière détaillée dans [Mocellin 1999]. En prenant en compte le terme de contact pénalisé, nous obtenons la formulation discrétisée finale (3.18) de notre problème de forgeage :

$$\left\{ \begin{array}{l} \forall k = 1, \dots, nbnoe; \forall i = 1, \dots, 3 \\ \int_{\Omega_h} 2K \left(\sqrt{3} \dot{\bar{\epsilon}}_h \right)^{m-1} \dot{\bar{\epsilon}}_h (v_h^l + v_h^b) : \dot{\bar{\epsilon}}_h (N_k^l e_i) d\Omega_h + \int_{\partial\Omega_h} 1_{\partial\Omega_f} \alpha_f K \|\Delta v_{gh}\|^{m-1} (\Delta v_{gh} \cdot e_i) N_k dS_h - \\ \int_{\Omega_h} p_h \operatorname{div}(N_k^l e_i) d\Omega_h + \rho 1_{\partial\Omega_c}(k) \left[(V_k - v_{\text{outil}}) \cdot n_k - \frac{\delta_k}{\Delta t} \right] n_k^l S_k = 0 \\ \forall e = 1, \dots, nbelt; \forall i = 1, \dots, 3 \\ \int_{\Omega_h^e} 2K \left(\sqrt{3} \dot{\bar{\epsilon}}_h \right)^{m-1} \dot{\bar{\epsilon}}_h (v_h^l + v_h^b) : \dot{\bar{\epsilon}}_h (N_e^b e_i) d\Omega_h - \int_{\Omega_h^e} p_h \operatorname{div}(N_e^b e_i) d\Omega_h = 0 \\ \forall k = 1, \dots, nbnoe \quad \int_{\Omega_h} N_k^l \operatorname{div}(v_h^l + v_h^b) d\Omega_h = 0 \end{array} \right. \quad (3.18)$$

où ρ est un coefficient de pénalisation pris suffisamment grand, $S_k = \int_{\tilde{\partial\Omega}_h} N_k^l dS_h$ la surface associée au nœud k , $1_{\partial\Omega_c}(k)$ l'indicatrice du contact au nœud k , $\partial\Omega_f$ surface de frottement et $1_{\partial\Omega_f}(x)$ l'indicatrice de frottement au nœud x , avec :

$$\begin{aligned} \partial\Omega_f &= \{x \in \partial\Omega \text{ tel que } \delta \leq 0\} \\ \forall x \in \partial\Omega, &\begin{cases} 1_{\partial\Omega_f}(x) = 1 & \text{si } \delta(x) \leq 0 \\ 1_{\partial\Omega_f}(x) = 0 & \text{sinon} \end{cases} \end{aligned} \quad (3.19)$$

$$\begin{cases} 1_{\partial\Omega_c}(k) = 1 & \text{si } (V_k - v_{outil}) \cdot n_k - \frac{\delta_k}{\Delta t} > 0 \\ 1_{\partial\Omega_c}(k) = 0 & \text{sinon} \end{cases} \quad (3.20)$$

Nous pouvons écrire le système (3.18) sous la forme d'un système vectoriel non linéaire :

$$R(v^l, v^b, p) = 0 \quad (3.21)$$

III.2.1.4. Méthode de résolution implémentée dans FORGE3®

Dans FORGE3®, on utilise l'algorithme de Newton-Raphson pour résoudre itérativement le système (3.21) à chaque incrément de temps. Etant donné le triplet initial $(v_{init}^l, v_{init}^b, p_{init})$, on cherche une correction $(\Delta v^l, \Delta v^b, \Delta p)$ à apporter telle que :

$$R(v_{init}^l + \Delta v^l, v_{init}^b + \Delta v^b, p_{init} + \Delta p) = 0 \quad (3.22)$$

A l'aide d'un développement de Taylor à l'ordre un, on obtient :

$$R(v_{init}^l, v_{init}^b, p_{init}) + \frac{\partial R}{\partial v^l} \Delta v^l + \frac{\partial R}{\partial v^b} \Delta v^b + \frac{\partial R}{\partial p} \Delta p = 0 \quad (3.23)$$

Ce qui conduit à la résolution du système linéaire :

$$\begin{pmatrix} \frac{\partial R_l}{\partial v^l} & \frac{\partial R_l}{\partial v^b} & \frac{\partial R_l}{\partial p} \\ \frac{\partial R_b}{\partial v^l} & \frac{\partial R_b}{\partial v^b} & \frac{\partial R_b}{\partial p} \\ \frac{\partial R_p}{\partial v^l} & \frac{\partial R_p}{\partial v^b} & \frac{\partial R_p}{\partial p} \end{pmatrix} \begin{pmatrix} \Delta v^l \\ \Delta v^b \\ \Delta p \end{pmatrix} = - \begin{pmatrix} R_l \\ R_b \\ R_p \end{pmatrix} \quad (3.24)$$

Il faut remarquer que le terme $\frac{\partial R_p}{\partial p} = 0$ [Mocellin 1999]. De plus, une propriété du champ « bulle » dans le cas linéaire ($m = 1$), qui s'étend au cas non linéaire [Perchat 2000] donne :

$$\frac{\partial R_l}{\partial v^b} = \frac{\partial R_b}{\partial v^l} = 0 \quad (3.25)$$

Nous pouvons maintenant simplifier le système (3.24) comme suit :

$$\begin{pmatrix} \frac{\partial R_l}{\partial v^l} & 0 & \frac{\partial R_l}{\partial p} \\ 0 & \frac{\partial R_b}{\partial v^b} & \frac{\partial R_b}{\partial p} \\ \frac{\partial R_p}{\partial v^l} & \frac{\partial R_p}{\partial v^b} & 0 \end{pmatrix} \begin{pmatrix} \Delta v^l \\ \Delta v^b \\ \Delta p \end{pmatrix} = - \begin{pmatrix} R_l \\ R_b \\ R_p \end{pmatrix} \quad (3.26)$$

Si on extrait la seconde ligne du système (3.26), le champ de vitesse « bulle » peut s'exprimer en fonction du champ de pression. Finalement, le système à résoudre se met sous la forme :

$$H \begin{pmatrix} \Delta v^l \\ \Delta p \end{pmatrix} = -R_{\text{modif}} \quad (3.27)$$

où H est la matrice hessienne symétrique et R_{modif} le second membre modifié obtenus après l'élimination du terme « bulle ».

Dans FORGE3®, la résolution de (3.27) se fait de manière itérative par la méthode du résidu minimal préconditionné. L'intérêt d'une méthode itérative vis à vis d'une méthode directe est énorme, non seulement en terme de temps de calcul mais aussi en terme de coût de stockage [Marie 1997].

III.2.1.5. Remaillage et transport des variables dans FORGE3®

FORGE3® a pour but de simuler des procédés de mise en forme en mettant en jeu de grandes déformations. Sous l'action des outils, la matière se déforme et par conséquent le maillage aussi, puisque FORGE3® adopte une description Lagrangienne. Or, les résultats d'estimation de l'erreur a priori de la méthode éléments finis montrent que les éléments du maillage doivent être de forme aussi proche que celle d'éléments équilatéraux. La précision de méthode éléments finis dégénère avec la qualité du maillage. Donc, pour réserver une bonne qualité de maillage durant la simulation, Coupez [Coupez 1991] a développé un algorithme de remaillage automatique dans FORGE3®. Dès que le maillage est de mauvaise qualité, ou à intervalles fixes, un nouveau maillage de meilleure qualité est généré. Il est nécessaire de transporter les variables d'un maillage à l'autre. Pour passer de l'ancienne à la nouvelle configuration, nous définissons un opérateur linéaire de transport noté J par la relation (3.28) :

$$X' = JX \quad (3.28)$$

L'utilisation d'une opération de remaillage nécessite le développement d'une méthode de transport des variables d'histoire depuis l'ancienne configuration Ω jusqu'à la nouvelle Ω' . Il existe deux types de variables :

- Les variables P0 (constantes sur chaque élément) comme par exemple la déformation équivalente $\bar{\varepsilon}$, la partie déviatorique s du tenseur des contraintes, ... calculée aux points d'intégration (centre de l'élément).
- Les variables P1 (variables nodales) comme la vitesse, la pression ...

FORGE3® utilise la technique d'interpolation inverse pour le transport des variables, qui est décrite dans le paragraphe qui suit.

III.2.1.5.1. Transport d'une variable nodale P1

Le transport des variables nodales s'effectue en deux étapes :

- recherche de l'élément e de l'ancien maillage d'appartenance de chaque nœud k' du nouveau maillage

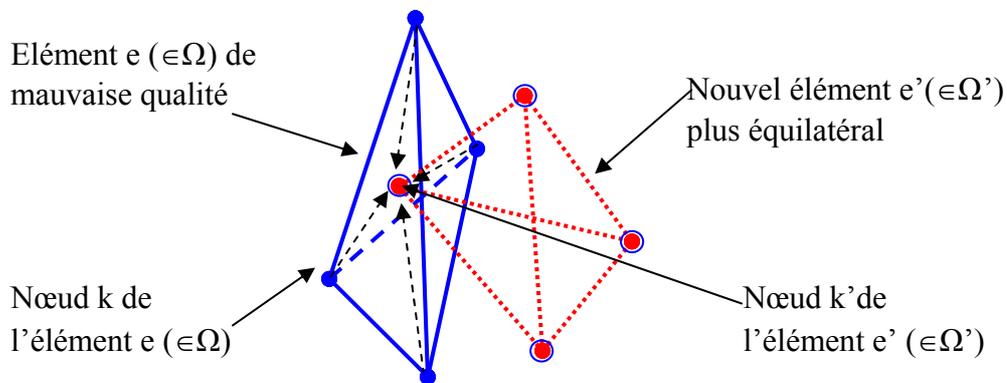


Figure 3.4. Transport par interpolation inverse

- interpolation des valeurs nodales depuis les sommets de l'élément e d'appartenance de l'ancien maillage Ω jusqu'au nœud k' du nouveau maillage Ω' (Figure 3.4). Cette interpolation est effectuée en utilisant les fonctions de forme de l'ancien maillage.

III.2.1.5.2. Transport d'une variable élémentaire P0

Le transport des variables P0 par élément est un peu plus délicat. Il est réalisé en deux étapes. Soit f la quantité P0 à transporter.

- dans un premier temps, nous ramenons le transport P0 à un transport P1 (Figure 3.5). Pour cela, nous calculons une interpolation nodale f_n à partir des valeurs aux éléments f_{elt} par la méthode des moindres carrés, en minimisant la fonctionnelle :

$$\Pi_{L^2}(f_n) = \int_{\Omega} (f - \bar{f})^2 dw = \left\| \sum_{n=1}^{Nbnoe} f_n N_n - \sum_{elt=1}^{Nbelt} f_{elt} 1_{elt} \right\|_{L^2}^2 \quad (3.29)$$

où N_n désigne les fonctions de base éléments finis et 1_{elt} la fonction indicatrice valant un dans l'élément considéré et zéro en dehors.

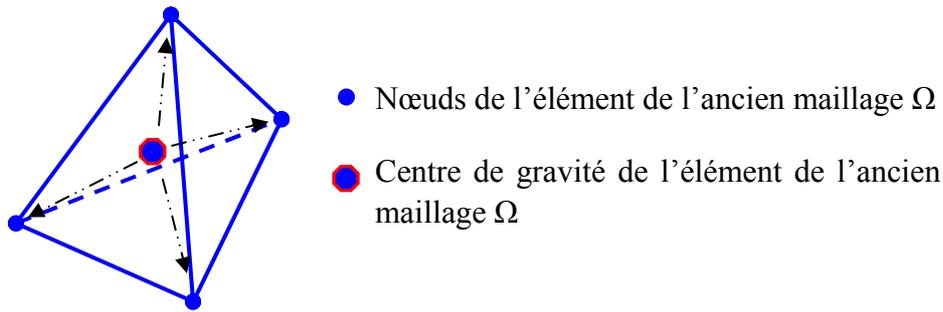


Figure 3.5. Passage d'une interpolation P0 à une interpolation P1

- dans une deuxième étape, les quantités nodales f_n sont transportées par la procédure de transport de P1 décrite précédemment (Figure 3.6). Ensuite, il faut ramener les variables au centre de l'élément nouveau

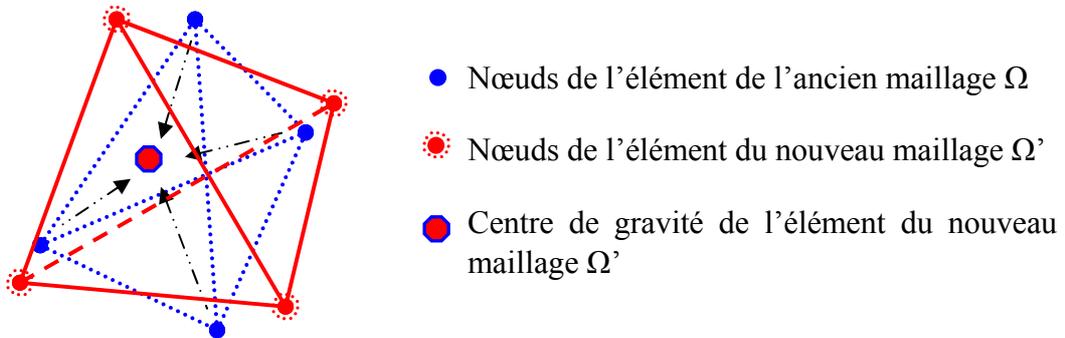


Figure 3.6. Transport P1 des variables au centre de gravité de l'élément du nouveau maillage Ω'

Nous venons de décrire le problème mécanique de forgeage et sa résolution au sein de FORGE3®. Dans notre travail, ce logiciel est utilisé pour calculer la valeur des fonctions coûts (ou critères d'optimisation) ainsi que ses gradients, qui seront ensuite utilisés dans nos problèmes d'optimisation. Ces fonctions coûts sont présentées dans le paragraphe qui suit.

III.2.2. Fonctions coûts

Différentes fonctions coûts ont été implémentées dans le logiciel FORGE3® dans le cadre du travail de Laroussi [Laroussi 2003] comme l'énergie totale de mise en forme, la mesure non-qualité de la surface, le remplissage de la matrice finale. Dans ce travail, nous allons présenter seulement deux fonctions coûts utilisées dans nos problèmes d'optimisation respectivement l'énergie totale et la mesure non-qualité de la surface ou de "repli".

III.2.2.1. Fonction coût énergie

La fonction coût énergie totale de forgeage a été introduite dans [Fourment et al. 1996]. Elle représente l'énergie dépensée pour déformer la matière de la configuration Ω_{t_0} à la configuration $\Omega_{t_{fin}}$. L'expression de cette fonction coût est la suivante :

$$\Phi_{ene} = \int_{t=t_0}^{t_{fin}} \left[\underbrace{\int_{\Omega_t} \sigma : \dot{\varepsilon} dw}_{\text{déformation}} + \underbrace{\int_{\partial\Omega_{ct}} \tau \cdot v ds}_{\text{frottement}} \right] dt \quad (3.30)$$

Dans le cadre d'une loi de comportement de Norton-Hoff et d'une loi de frottement de Norton, cette fonction s'écrit :

$$\Phi_{en} = \int_{t_0}^{t_{fin}} \left(\int_{\Omega_t} K (\sqrt{3} \dot{\varepsilon})^{m+1} dw + \int_{\partial\Omega_{ft}} \alpha_f K \|\Delta v_g\|^{p_f+1} ds \right) dt \quad (3.31)$$

où la quantité $\int_{\Omega_t} K (\sqrt{3} \dot{\varepsilon})^{m+1} dw$ est la puissance de déformation et la quantité $\int_{\partial\Omega_{ft}} \alpha_f K \|\Delta v_g\|^{p_f+1} ds$ la puissance de frottement.

III.2.2.2. Fonction coût repli

Le repli est un défaut majeur du forgeage. Un repli apparaît lorsque les frontières Ω_t de la pièce se chevauchent. A la fin du procédé, la fissure créée représente une zone de faiblesse inacceptable. Ce défaut est donc synonyme de rebut de la pièce. L'élimination de ce défaut est un problème clé de la conception du procédé. Deux questions se posent : Comment savoir quand le repli apparaît ? Et comment s'en éloigner ?

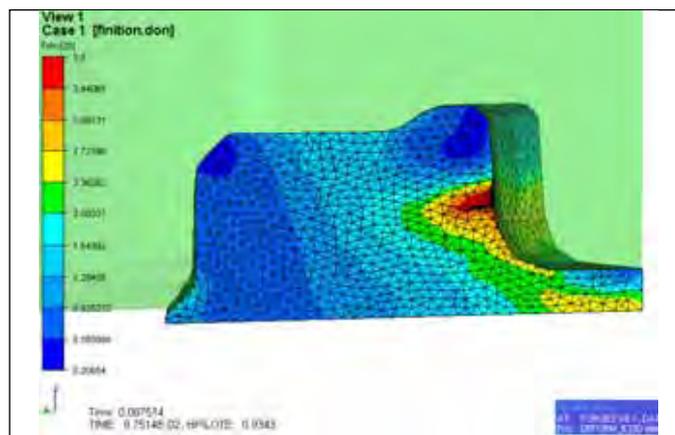


Figure 3.7. Isovaleurs de $\dot{\varepsilon}$ lors de la formation d'un repli de matière

Pour répondre à la première question, une fonction coût Φ_{repli} est utilisée, mais sa définition est délicate. On constate que dans la zone de repli, la matière est soumise à de forts cisaillements. Des valeurs anormalement élevées de $\dot{\varepsilon}$ sont obtenues, alors même que le repli

n'est pas encore formé, ni détecté (parfois plus de 10 fois supérieures à la valeur moyenne sur le reste de la surface libre – comme le montre la *Figure 3.7*).

Grâce à cette propriété, des fonctions coût Φ_{repli} basées sur la vitesse de déformation généralisée ont été proposées. Vieilledent [Vieilledent 1999] a proposé une fonction de la forme :

$$\Phi_{repli1} = \int_{t_0}^{t_{fin}} \left(\int_{\partial\Omega_t} \dot{\bar{\varepsilon}} ds \right) dt \quad (3.32)$$

Pourtant, cette fonction ne pouvait être utilisée que localement avec des boîtes prédéfinies supposant de connaître la zone de formation du repli. Dans son travail, M. Laroussi [Laroussi 2003] a modifié cette formule en ajoutant un coefficient λ afin de mieux détecter des replis. La fonction coût repli prend la forme :

$$\Phi_{repli2} = \int_{t_0}^{t_{fin}} \left(\int_{\partial\Omega_t} (\dot{\bar{\varepsilon}})^\lambda ds \right) dt \quad (3.33)$$

Le coefficient λ va nous permettre, en lui attribuant des valeurs supérieures à 1, d'amplifier les valeurs élevées de $\dot{\bar{\varepsilon}}$ dans les zones de repli.

Deux tests d'écrasement entre tas plats d'un cube entaillé (*Figure 3.8.a*) et d'un cube normal (*Figure 3.8.b*) ont été effectués. La dimension des cubes est de 50x50x50mm, la hauteur d'écrasement est de 20% de la hauteur initiale. On s'attend à ce que la valeur de la fonction coût repli soit beaucoup plus élevée dans le cas du cube entaillé qu'avec le cube plein.

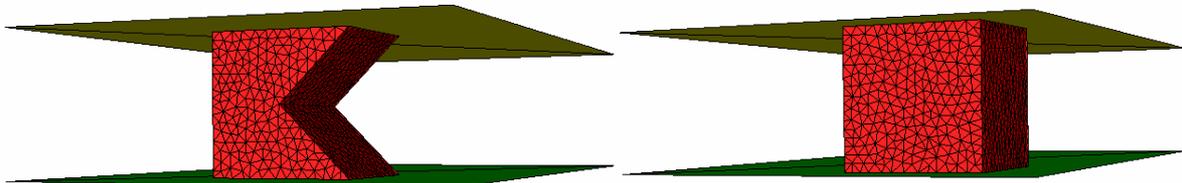


Figure 3.8. Tests d'écrasement d'un cube entaillé (a) et d'un cube plein (b)

En effet, dans le cas du cube entaillé, la zone de repli est petite donc la surface d'intégration des valeurs de $\dot{\bar{\varepsilon}}$ est faible. Le *Tableau 3.1* présente la comparaison des valeurs de Φ_{repli} obtenues. Le critère a une signification globale uniquement pour $\lambda \geq 4$. Autrement dit, le repli est détecté seulement pour des valeurs de λ supérieures à 4.

λ	1	2	3	4
Φ_{repli} CUBE PLEIN	2106	4687	10476	23510
Φ_{repli} CUBE ENTAILLE	1490	3419	9674	35427

Tableau 3.1 : Valeur de la fonction coût « repli » non-adimensionnée pour différentes valeurs de λ

Avec cette fonction coût, nous voyons bien que plus λ est grand, mieux la potentialité de repli est identifiée. Cependant, on ne peut utiliser une valeur de λ aussi grande que l'on veut, et on s'aperçoit que cette valeur dépend du problème.

Pour s'affranchir de ce problème, on adimensionnalise la fonction coût afin de pouvoir utiliser des valeurs plus grandes de λ (si nécessaire) et pour essayer d'avoir un critère quantitatif d'apparition de repli. La fonction proposée prend la forme suivante :

$$\Phi_{repli}^{a\ dim} = \frac{1}{t_{fin} - t_0} \int_{t_0}^{t_{fin}} \left(\frac{1}{S_{libref}} \int_{\partial\Omega_{lib}} \left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_{car}} \right)^\lambda ds \right)^{1/\lambda} dt \quad (3.34)$$

où $\dot{\varepsilon}_{car} = \frac{v_{ref}}{h_{ref}}$ est vitesse de déformation généralisée caractéristique

v_{ref} est la vitesse initiale de l'outil de forgeage, h_{ref} la course de forgeage, S_{libref} la surface libre référence (= surface libre initiale)

Avec la nouvelle formule, nous avons effectué les mêmes tests d'écrasement de cubes. Le *Tableau 3.2* regroupe les valeurs de $\Phi_{repli}^{a\ dim}$ pour différentes valeurs de λ .

λ	1	2	3	4	5	10
$\Phi_{repli}^{a\ dim}$ CUBE PLEIN	1.040	1.103	1.154	1.209	1.272	1.666
$\Phi_{repli}^{a\ dim}$ CUBE ENTAILLE	0.511	1.027	1.633	2.306	2.959	5.336

Tableau 3.2 : Valeur de la fonction coût « repli » adimensionnalisée pour différentes valeurs de λ

On constate que dans le cas du cube normal (sans repli) la valeur de la fonction coût de repli ne dépend pas beaucoup de la valeur de λ . Elle est proche de 1. La vitesse de déformation généralisée est relativement homogène sur la surface libre. Il n'y a pas une grande influence de λ dans ce cas. Au contraire, dans le cas du cube entaillé, λ joue un rôle important pour amplifier des valeurs anormalement grandes de $\dot{\varepsilon}$ dans les zones du repli. On constate que, à partir de $\lambda = 3$, on obtient déjà une différence notable entre les deux cas. Ces résultats nous permettent de dire que la nouvelle fonction coût repli (3.34) donne des résultats plus globaux. Elle fonctionne bien en détectant le repli avec des valeurs de $\lambda \geq 3$. Cela donne une idée pour savoir s'il y a repli ou pas : on fait le rapport des fonctions replis obtenues avec $\lambda = 1$ et $\lambda = 10$ (ou 5). Si ce rapport est grand, il y a probablement un repli.

Dans les parties qui suivent, nous allons utiliser cette nouvelle formule avec une valeur de $\lambda = 10$.

Une fois que le repli est détecté, il nous faut encore répondre à la deuxième question « Comment faire pour l'éliminer ? ». Cela est un des objectifs principaux de ce travail de thèse. Comme nous l'avons mentionné dès le début de ce chapitre, l'utilisation du gradient des

fonctions coûts par rapport aux paramètres à optimiser pour utiliser des algorithmes à base de gradient est une manière de résoudre efficacement ce problème d'optimisation. Le calcul du gradient est complexe mais les verrous principaux de ce problème ont été levés dans [Laroussi 2003] en utilisant la méthode de l'Etat Adjoint. Le prochain paragraphe va décrire cette méthode de l'Etat Adjoint implémentée dans FORGE3® pour le calcul du gradient des fonctions coûts.

III.2.3. Calcul du gradient par la méthode de l'Etat Adjoint

III.2.3.1. Equations de la méthode de l'Etat Adjoint

Nous avons différentes fonctions coûts $\Phi(\mu)$ qui représentent différents critères du problème d'optimisation. L'objectif de ce paragraphe est de décrire la méthode de l'Etat Adjoint utilisée pour calculer le gradient de ces fonctions coût Φ par rapport aux paramètres d'optimisation μ , qui peuvent être de deux types :

- paramètres du procédé (vitesse de forgeage, coefficient de frottement, ...)
- paramètres de forme (forme du lopin, forme des outils, dimension, ...)

Pour un procédé de forgeage instationnaire composé de N incréments de temps, la loi de comportement étant sans "histoire", les fonctions coût $\Phi(\mu)$ et le résidu $R(\mu)$ du problème de forgeage, après discrétisation par éléments finis, sont des fonctions dépendant uniquement de μ , de la position X des nœuds du maillage et de V et P . A un incrément donné i , nous pouvons écrire:

$$\begin{aligned} R_i(\mu) &= R_i(\mu, X_i(\mu), V_i(\mu), P_i(\mu)) \\ \Phi(\mu) &= \Phi(\mu, X(\mu), V(\mu), P(\mu)) \end{aligned} \quad (3.35)$$

Avec

$$\begin{aligned} X(\mu) &= (X_i(\mu))_{i=0, \dots, N} \\ V(\mu) &= (V_i(\mu))_{i=0, \dots, N-1} \\ P(\mu) &= (P_i(\mu))_{i=0, \dots, N-1} \end{aligned}$$

La méthode de l'Etat Adjoint consiste à définir un Lagrangien Λ pour chaque fonction coût $\Phi(\mu)$ en introduisant le vecteur des états adjoints $(\lambda_i)_{i=0, N-1}$ de la manière suivante :

$$\Lambda(\mu, \lambda) = \Phi(\mu, X(\mu), V(\mu), P(\mu)) + \sum_{i=0}^{N-1} \lambda_i R_i(\mu, X_i(\mu), V_i(\mu), P_i(\mu)) \quad (3.36)$$

Si $(V_i(\mu), P_i(\mu))$ vérifie les équation du problème $R_i(\mu, X_i, V_i, P_i) = 0$ à chaque incrément de temps ($i = 0, \dots, N-1$), nous avons :

$$\Lambda(\mu) = \Phi(\mu) \quad \text{et donc} \quad \frac{d\Lambda}{d\mu}(\mu) = \frac{d\Phi}{d\mu}(\mu) \quad \forall \mu \quad (3.37)$$

Le calcul du gradient de $\Phi(\mu)$ est donc équivalent à celui de Λ . On omet les (μ) pour simplifier les notations. Aussi :

$$\frac{d\Lambda}{d\mu} = \frac{d\Phi}{d\mu} + \sum_{i=0}^{N-1} {}^T \lambda_i \frac{dR_i}{d\mu} \quad (3.38)$$

avec
$$\frac{d\Phi}{d\mu} = \frac{\partial\Phi}{\partial\mu} + \sum_{i=0}^N \frac{\partial\Phi}{\partial X_i} \frac{dX_i}{d\mu} + \sum_{i=0}^{N-1} \left(\frac{\partial\Phi}{\partial V_i} \frac{dV_i}{d\mu} + \frac{\partial\Phi}{\partial P_i} \frac{dP_i}{d\mu} \right) \quad (3.39)$$

et
$$\forall i=0, N-1 \quad \frac{dR_i}{d\mu} = \frac{\partial R_i}{\partial\mu} + \frac{\partial R_i}{\partial X_i} \frac{dX_i}{d\mu} + \frac{\partial R_i}{\partial V_i} \frac{dV_i}{d\mu} + \frac{\partial R_i}{\partial P_i} \frac{dP_i}{d\mu} \quad (3.40)$$

Considérons d'abord le cas où aucun remaillage n'est opéré, l'équation (3.11) peut être réécrite :

$$\forall i=0, N \quad X_i = X_0 + \sum_{j=0}^{i-1} V_j \Delta t_j \quad (3.41)$$

La différentiation de cette équation nous donne :

$$\forall i=0, N \quad \frac{dX_i}{d\mu} = \frac{dX_0}{d\mu} + \sum_{j=0}^{i-1} \frac{dV_j}{d\mu} \Delta t_j + \overbrace{\sum_{j=0}^{i-1} V_j \frac{d\Delta t_j}{d\mu}}^{\approx 0} \quad (3.42)$$

Nous supposons que le vecteur des paramètres n'a aucune influence sur le pas de temps $\left(\frac{d\Delta t_j}{d\mu} \approx 0 \right)$, ce qui est le cas lorsque le pas de temps est constant, et qui est une bonne approximation autrement [Balan 1996] [Vielledent 1999].

L'égalité (3.42) est injectée dans les équations (3.39) et (3.40) dans le but d'éliminer les dérivées $\left(\frac{dX_i}{d\mu} \right)_{i=1, \dots, N}$. Après quelques transformations, nous obtenons :

$$\begin{aligned} \frac{d\Lambda}{d\mu} &= \frac{\partial\Phi}{\partial\mu} + \sum_{i=0}^N \frac{\partial\Phi}{\partial X_i} \frac{dX_0}{d\mu} + \sum_{i=0}^{N-1} {}^T \lambda_i \left(\frac{\partial R_i}{\partial\mu} + \frac{\partial R_i}{\partial X_i} \frac{dX_0}{d\mu} \right) + \sum_{i=0}^{N-1} \left(\frac{\partial\Phi}{\partial P_i} + {}^T \lambda_i \frac{\partial R_i}{\partial P_i} \right) \frac{dP_i}{d\mu} \\ &+ \sum_{i=0}^{N-2} \left(\frac{\partial\Phi}{\partial V_i} + {}^T \lambda_i \frac{\partial R_i}{\partial V_i} + \Delta t_i \left(\sum_{j=i+1}^N \frac{\partial\Phi}{\partial X_j} + \sum_{j=i+1}^{N-1} {}^T \lambda_j \frac{\partial R_j}{\partial X_j} \right) \right) \frac{dV_i}{d\mu} \\ &+ \left(\frac{\partial\Phi}{\partial V_{N-1}} + \Delta t_{N-1} \frac{\partial\Phi}{\partial X_N} + {}^T \lambda_{N-1} \frac{\partial R_{N-1}}{\partial V_{N-1}} \right) \frac{dV_{N-1}}{d\mu} \end{aligned} \quad (3.43)$$

Le coût de calcul des termes $\frac{dV_i}{d\mu}$ et $\frac{dP_i}{d\mu}$ dépend du nombre de paramètres μ . La méthode de l'état adjoint a pour but d'éviter cette dépendance au nombre de paramètres. A chaque

incrément de temps i , on choisit donc l'Etat Adjoint λ_i permettant d'éliminer ces deux termes. Pour cela, le vecteur des états adjoints $(\lambda_i)_{i=0,\dots,N-1}$ doit vérifier les équations suivantes :

- élimination de $\frac{dV_i}{d\mu}$

$$\begin{cases} {}^T\lambda_{N-1} \frac{\partial R_{N-1}}{\partial V_{N-1}} = -\left(\frac{\partial \Phi}{\partial V_{N-1}} + \Delta t_{N-1} \frac{\partial \Phi}{\partial X_N} \right) \\ \forall i = N-2, \dots, 0 \quad {}^T\lambda_i \frac{\partial R_i}{\partial V_i} = -\left(\frac{\partial \Phi}{\partial V_i} + \Delta t_i \sum_{j=i+1}^N \frac{\partial \Phi}{\partial X_j} + \Delta t_i \left(\sum_{j=i+1}^{N-1} {}^T\lambda_j \frac{\partial R_j}{\partial X_j} \right) \right) \end{cases} \quad (3.44)$$

- élimination de $\frac{dP_i}{d\mu}$

$$\forall i = N-1, \dots, 0 \quad {}^T\lambda_i \frac{\partial R_i}{\partial P_i} = -\frac{\partial \Phi}{\partial P_i} \quad (3.45)$$

Afin de simplifier les notations, nous introduisons une variable intermédiaire Γ , définie par :

$$\forall i = -1, \dots, N-1 \quad \Gamma_i = \frac{\partial \Phi}{\partial X_N} + \sum_{j=i+1}^{N-1} \left(\frac{\partial \Phi}{\partial X_i} + {}^T\lambda_j \frac{\partial R_j}{\partial X_i} \right)^T \quad (3.46)$$

ou encore :

$$\begin{cases} \Gamma_{N-1} = \frac{\partial \Phi}{\partial X_N} \\ \forall i = N-1, \dots, 0 \quad \Gamma_{i-1} = \Gamma_i + \frac{\partial \Phi}{\partial X_i} + {}^T\lambda_i \frac{\partial R_i}{\partial X_i} \end{cases} \quad (3.47)$$

En utilisant des variables compactées définies par $W_i = \begin{pmatrix} V_i \\ P_i \end{pmatrix}$ et $\bar{\Gamma}_i = \begin{pmatrix} \Gamma_i \\ 0 \end{pmatrix}$, les équations (3.44) et (3.45) peuvent être condensées en le système suivant :

$$\lambda_i = -\left(\frac{\partial R_i}{\partial W_i} \right)^{-1T} \left(\frac{\partial \Phi}{\partial W_i} + \Delta t_i \bar{\Gamma}_i \right) \quad \forall i = N-1, \dots, 0 \quad (3.48)$$

L'équation (3.47) montre que la variable $\bar{\Gamma}_i$ est déterminée à partir de la variable $\bar{\Gamma}_{i+1}$. Il en résulte que $\bar{\Gamma}_i$ doit être calculée incrémentalement dans le sens inverse du temps, c'est-à-dire de $i = N-1$ à $i = 0$. Le vecteur des états adjoints λ_i doit donc être calculé de la même manière (équation (3.48)). Ceci représente la difficulté majeure pour implémenter la méthode de l'état adjoint dans le cadre de problèmes instationnaires [Fourment et al. 2001] [Chung et al. 2003]. Une fois le vecteur des états adjoints $(\lambda_i)_{i=0,\dots,N-1}$ calculé, on obtient :

$$\frac{d\Phi}{d\mu}(\mu) = \frac{\partial \mathcal{A}}{\partial \mu}(\mu, \lambda) = \frac{\partial \Phi}{\partial \mu} + \sum_{i=0}^{N-1} {}^T\lambda_i \frac{\partial R_i}{\partial \mu} + \Gamma_{-1} \frac{dX_0}{d\mu} \quad (3.49)$$

où X_0 représente les coordonnées des nœuds du maillage initial de la simulation FORGE3®

Nous constatons que la méthode de l'état adjoint est caractérisée par :

- le fait que le vecteur des états adjoints est déterminé dans le sens inverse du temps, en commençant par le dernier incrément.
- le fait que l'analyse de sensibilité doit donc être réalisée après la simulation du procédé de forgeage. Elle nécessite donc le stockage des données de chaque incrément de calcul pour le calcul inverse.
- le fait qu'à chaque incrément de temps, elle nécessite la résolution d'autant de systèmes linéaires qu'il y a de fonctions coût. Le coût de la méthode est donc bien indépendant du nombre de paramètres.

III.2.3.2. Remaillage et transport des variables adjointes

Considérons maintenant une simulation du forgeage avec des remaillages.

III.2.3.2.1. Influences du remaillage sur les équations de l'Etat Adjoint

Supposons qu'un remaillage intervienne à l'incrément r ($0 \leq r \leq N-1$) pour générer un nouveau maillage de meilleure qualité Ω'_r . Notons J_r l'opérateur linéaire de transport permettant de passer de l'ancienne à la nouvelle configuration :

$$X'_r = J_r X_r \quad (3.50)$$

On doit transporter le terme $\frac{dX}{d\mu}$ de l'ancien maillage vers le nouveau comme pour la méthode directe [Balan 1996]. La manière la plus simple est d'utiliser l'opérateur de transport par interpolation inverse J déjà utilisé dans FORGE3® :

$$\frac{dX'_r}{d\mu} = \frac{dX'_r}{\underbrace{dX_r}_{J_r}} \frac{dX_r}{d\mu} \quad (3.51)$$

Pour $r \leq i$, l'équation (3.42) doit être remplacée par :

$$\forall i = r, \dots, N-1 \quad \frac{dX_i}{d\mu} = J_r \left(\frac{dX_0}{d\mu} + \sum_{j=0}^{r-1} \frac{dV_j}{d\mu} \Delta t_j \right) + \sum_{j=r}^{i-1} \frac{dV_j}{d\mu} \Delta t_j \quad (3.52)$$

Cette modification transforme l'équation (3.43) en la suivante :

$$\begin{aligned}
\frac{d\Lambda}{d\mu}(\mu) &= \frac{\partial\Phi}{\partial\mu} + \sum_{i=0}^{N-1} {}^T\lambda_i \frac{\partial R_i}{\partial\mu} + [\Gamma_{-1} - \Gamma_{r-1}(1 - J_r)] \frac{dX_0}{d\mu} \\
&+ \sum_{i=0}^{r-1} \left(\frac{\partial\Phi}{\partial V_i} + {}^T\lambda_i \frac{\partial R_i}{\partial V_i} + \Delta t_i [\Gamma_i - \Gamma_{r-1}(1 - J_r)] \right) \frac{dV_i}{d\mu} \\
&+ \sum_{i=r}^{N-1} \left(\frac{\partial\Phi}{\partial V_i} + {}^T\lambda_i \frac{\partial R_i}{\partial V_i} + \Delta t_i \Gamma_i \right) \frac{dV_i}{d\mu} + \sum_{i=0}^{N-1} \left(\frac{\partial\Phi}{\partial P_i} + {}^T\lambda_i \frac{\partial R_i}{\partial P_i} \right) \frac{dP_i}{d\mu}
\end{aligned} \tag{3.53}$$

Nous constatons que l'opérateur de transport s'applique uniquement à la variable Γ . S'il y a un remaillage à l'incrément i , il suffit de modifier le calcul de Γ comme suit pour que l'équation (3.47) reste valable :

$$\forall i = N-1, \dots, 0 \quad \begin{cases} \Gamma_{i-1} = \Gamma_i + \frac{\partial\Phi}{\partial X_i} + {}^T\lambda_i \frac{\partial R_i}{\partial X_i} & \text{[pas de remaillage à l'incrément } i \text{]} \\ \Gamma_{i-1} = \left(\Gamma_i + \frac{\partial\Phi}{\partial X_i} + {}^T\lambda_i \frac{\partial R_i}{\partial X_i} \right) J_i & \text{[remaillage à l'incrément } i \text{]} \end{cases} \tag{3.54}$$

D'un point de vue pratique, les remarques suivantes sont faites :

- Premièrement, la dérivée $\frac{dX_0}{d\mu}$ n'est utilisée qu'à l'incrément 0, donc elle n'a pas besoin d'être transportée au cours des remaillages comme dans la méthode de différentiation directe.
- Deuxièmement, on constate que l'opérateur de transport intervient sous sa forme transposée ${}^T J$, pour calculer la variable ${}^T \Gamma$ (Γ intervient lui aussi sous sa forme transposée dans le calcul de la variable adjointe λ - voir l'équation (3.48)). S'il y a un remaillage à l'incrément r , nous avons :

$$\begin{aligned}
\Gamma_{r-1} &= \Gamma_{r-1}' J_r \\
\text{et donc } {}^T \Gamma_{r-1} &= {}^T J_r {}^T \Gamma_{r-1}'
\end{aligned} \tag{3.55}$$

Il faut remarquer que le calcul des variables Γ_i et λ_i se fait dans le sens inverse du temps, alors que l'opérateur de transport J_r va de l'ancienne configuration Ω_h vers la nouvelle Ω_h' .

Remarque : Lorsque des remaillages sont effectués, il faut stocker toutes les données concernant l'ancien et le nouveau maillage (connectivités volumiques et surfaciques, nombre de nœuds, d'éléments, de faces, ...). Ce stockage est impératif car ces informations sont utilisées lors de la résolution "adjointe".

III.2.3.2.2. Transport des variables adjointes

Il faut noter d'abord que le transport des données est réalisé par interpolation inverse. L'équation (3.50) est donc réécrite sous sa forme scalaire :

$$\forall k' \in \Omega', \quad X^{k'} = \sum_{l \in \mathcal{G}(k')} J^{kl} X^l \quad (3.56)$$

où $\mathcal{G}(k')$ représente l'ensemble des nœuds l de l'élément e de l'ancien maillage Ω qui contient le nœud k' du nouveau maillage Ω' .

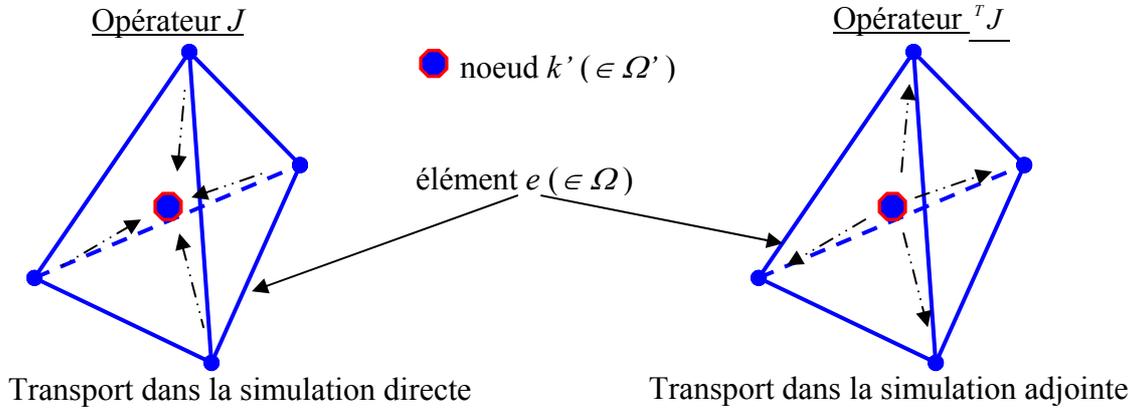


Figure 3.9. Opérateur de transport pour la simulation directe et le calcul de l'Etat Adjointe

Dans le cadre de la méthode de l'état adjoint, nous voulons transporter la variable nodale ${}^T \Gamma$ (variable P1) du nouveau maillage Ω' vers l'ancien maillage Ω . L'équation (3.55) s'écrit :

$$\forall l \in \Omega, \quad {}^T \Gamma^l = \sum_{\substack{k' \in \Omega' \\ \text{tel que: } l \in \mathcal{G}(k')}} J^{klT} \Gamma^{k'} \quad (3.57)$$

La valeur $\Gamma^{k'}$ au nœud k' est connue car lors du calcul de l'adjoint, Ω' est le maillage courant (Figure 3.9). On peut trouver l'algorithme pour calculer la valeur de ${}^T \Gamma$ sur le maillage Ω dans [Laroussi 2003].

Le coût de calcul, ainsi que la précision de cet opérateur ${}^T J$, sont tout à fait similaires à ceux de l'opérateur J de la simulation directe.

III.2.3.3. Technique semi-analytique pour calculer les dérivées partielles

Pour calculer les différentes dérivées partielles qui apparaissent dans les équations de la méthode de l'état adjoint ($\frac{\partial \Phi}{\partial \mu}$, $\frac{\partial R_i}{\partial \mu}$, $\frac{\partial R_i}{\partial X_i}$, $\frac{\partial \Phi}{\partial X_i}$ et $\frac{\partial \Phi}{\partial W_i}$), nous retenons une technique semi-analytique. Elle présente l'avantage d'être flexible, facile à implémenter et précise. Nous utilisons des quotients différentiels décentrés à droite, comme par exemple :

$$\forall j = 1, \dots, nbpar \quad \frac{\partial \Phi}{\partial \mu_j} = \lim_{\|\Delta \mu_j\| \rightarrow 0} \frac{\Phi(\mu + \Delta \mu_j, X, W) - \Phi(\mu, X, W)}{\|\Delta \mu_j\|} \quad (3.58)$$

où

$$\forall j = 1, \dots, nbpar \quad \Delta \mu_j = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \varepsilon \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} 1 \\ 2 \\ \vdots \\ j \\ \vdots \\ nbpar \end{matrix} \quad (3.59)$$

où ε est une petite perturbation prise en général égale à 10^{-6} en valeur relative.

Le coût de calcul des dérivées $\frac{\partial \Phi}{\partial \mu}$ et $\frac{\partial R_i}{\partial \mu}$ n'est pas grand car il est proportionnel au nombre de paramètres (qui est petit devant le nombre d'éléments du maillage). En revanche, pour le calcul des dérivées $\frac{\partial \Phi}{\partial X_i}$, $\frac{\partial \Phi}{\partial W_i}$ et $\frac{\partial R_i}{\partial X_i}$, le coût est proportionnel au nombre de nœuds du maillage si la programmation est effectuée de la même manière. Il résulte que pour un maillage moyen ce coût devient rapidement exorbitant. Pour ce problème, une méthode de programmation basée sur la décomposition de Φ et R_i au niveau des éléments et des faces du maillage a été retenue. Le point clé consiste à remarquer que lorsqu'on perturbe un nœud k du maillage, l'influence de cette perturbation se limite aux éléments et aux faces contenant ce nœud k . La perturbation est locale et l'assemblage n'est fait que pour les éléments appartenant au nœud k .

Le surcoût du calcul des dérivées par la méthode semi-analytique est ainsi indépendant du nombre de nœuds. Ce coût est extrêmement raisonnable puisqu'il ne traduit que par un nombre limité d'assemblages de résidu et d'évaluations de la fonction coût. Il est très largement inférieur au coût de calcul nécessaire à la résolution d'un système linéaire 3D.

Notons enfin que la dérivée $\frac{\partial R_i}{\partial X_i}$ est une matrice dont le nombre de coefficients est proportionnel à $(nbnoe)^2$. Stocker une telle matrice est très vite problématique, dès que le nombre de nœuds devient grand. Or, seul le produit de cette dérivée avec le vecteur des états adjoint λ_i apparaît dans les équations (voir équation (3.47)). Ainsi, on calcule directement la quantité ${}^t \lambda_i \frac{\partial R_i}{\partial X_i}$ au niveau d'un élément ou d'une face, comme expliqué précédemment.

III.2.3.4. Traitement des termes de contact et de frottement

Considérons le calcul du terme $\frac{\partial R}{\partial X}$ par la méthode semi-analytique. On décompose le résidu en vitesse en différentes contributions comme suit :

$$R = R_{rheo} + R_{frott} + R_{incomp} + R_{contact} \quad (3.60)$$

avec : $R_{rheo} = \int_{\Omega} s : \dot{\varepsilon}^* dw$ est le terme volumique contenant la loi de comportement du matériau,

$R_{incomp} = - \int_{\Omega} p \operatorname{div} v^* dw$ est le terme volumique lié à la condition d'incompressibilité

$R_{frott} = - \int_{\partial\Omega} 1_{\partial\Omega_f} \tau \cdot v^* ds$ est le terme surfacique contenant la loi de frottement τ

$R_{contact}^k$ est la contribution du contact au nœud k , donnée par

$$R_{contact}^k = \rho 1_{\partial\Omega_c}(k) \left[(V_k - v_{outil}) \cdot n_k - \frac{\delta_k}{\Delta t} \right] n_k S_k \quad \forall k \in \partial\Omega_h \quad (3.61)$$

Pour calculer les dérivées $\frac{\partial R_{contact}}{\partial X}$ et $\frac{\partial R_{frott}}{\partial X}$, nous devons différencier l'inégalité de contact unilatéral par rapport aux coordonnées du maillage. Ce problème a été abordé par de nombreux auteurs [Balan 1996] [Chung et al. 1998] [Vieilledent 1999] [Zabaras et al. 2000] [Stupkiewicz et al. 2002] [Chung et al. 2002] [Stupkiewicz et al. 2003]. Les fonctions indicatrices du contact et du frottement apparaissent dans les termes de la condition de contact unilatéral sont des fonctions de type « partie positive ». La particularité de ces fonctions est qu'elles peuvent présenter des points de non-dérivabilité, comme le montre la Figure 3.10 avec la fonction $[x]^+$ non dérivable en 0 :

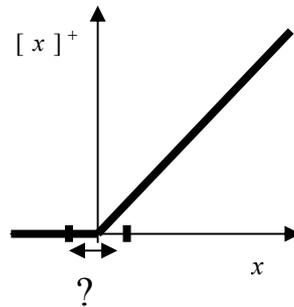


Figure 3.10. Fonction $[x]^+$ non dérivable en 0 et pour laquelle la dérivée à droite est choisie

On voit bien que $[x]^+$ accepte une dérivée à droite et une dérivée à gauche. On choisit la dérivée à droite pour $1_{\partial\Omega_c}$ et $1_{\partial\Omega_f}$ avec les raisonnements suivants:

- physiquement, après perturbation, le nœud a plus de raisons de rester en contact que de quitter systématiquement le contact. On suppose donc que la surface de contact $\partial\Omega_c$ reste inchangée ainsi on retient le choix des travaux antérieurs de plusieurs auteurs : [Balan 1996] [Chung et al. 1998] [Vieilledent 1999] [Zabaras et al. 2000] [Stupkiewicz et al. 2002] [Chung et al. 2002] [Stupkiewicz et al. 2003] et [Laroussi 2003]. Cela revient à considérer la condition de contact unilatéral

comme étant bilatérale au moment de la différentiation. L'inégalité devient égalité et on a :

$$\frac{\partial \mathbf{1}_{\partial\Omega_c}}{\partial X} = \frac{\partial \mathbf{1}_{\partial\Omega_f}}{\partial X} = 0 \quad (3.62)$$

- ce choix est cohérent avec celui du terme $\frac{\partial [h]^+}{\partial v}$, effectué pour calculer la contribution du contact à la matrice hessienne dans FORGE3® au sein de l'algorithme de Newton-Raphson généralisé (au contact unilatéral).

On écrit maintenant le résidu en vitesse en faisant intervenir la dépendance implicite des fonctions indicatrices du contact et du frottement, ainsi que la distance de contact δ et la normale n comme suit :

$$R(\mu, X, W) = \tilde{R}(X, W, \mathbf{1}_{\partial\Omega_c}, \mathbf{1}_{\partial\Omega_f}, \delta, n, \mu) \quad (3.63)$$

$$\text{où } \mathbf{1}_{\partial\Omega_c} = \mathbf{1}_{\partial\Omega_c}(\mu, X, W), \quad \mathbf{1}_{\partial\Omega_f} = \mathbf{1}_{\partial\Omega_f}(\mu, X), \quad \delta = \delta(\mu, X), \quad n = n(\mu, X) \quad (3.64)$$

Nous sommes en mesure d'écrire la dérivée $\frac{\partial R}{\partial X}$ comme suit :

$$\frac{\partial R}{\partial X} = \frac{\partial R}{\partial X} \Big|_{\mathbf{1}_{\partial\Omega_c}, \mathbf{1}_{\partial\Omega_f}, \delta, n} + \frac{\partial R}{\partial \delta} \frac{\partial \delta}{\partial X} + \frac{\partial R}{\partial n} \frac{\partial n}{\partial X} + \underbrace{\frac{\partial R}{\partial \mathbf{1}_{\partial\Omega_c}} \frac{\partial \mathbf{1}_{\partial\Omega_c}}{\partial X}}_{=0} + \underbrace{\frac{\partial R}{\partial \mathbf{1}_{\partial\Omega_f}} \frac{\partial \mathbf{1}_{\partial\Omega_f}}{\partial X}}_{=0} \quad (3.65)$$

Ainsi, lorsque l'on calcule le terme $\frac{\partial R}{\partial X}$ par la technique semi-analytique en effectuant des perturbations locales sur X :

- la condition de contact ne change pas, donc les surfaces de contact $\partial\Omega_c$ et de frottement $\partial\Omega_f$ restent inchangées
- seules les distances de contact δ et les normales n sont perturbées, et sont à recalculer

Pour le calcul du terme $\frac{\partial R}{\partial \mu}$, le même raisonnement est appliqué:

$$\frac{\partial R}{\partial \mu} = \frac{\partial R}{\partial \mu} \Big|_{\mathbf{1}_{\partial\Omega_c}, \mathbf{1}_{\partial\Omega_f}, \delta, n} + \frac{\partial R}{\partial \delta} \frac{\partial \delta}{\partial \mu} + \frac{\partial R}{\partial n} \frac{\partial n}{\partial \mu} + \underbrace{\frac{\partial R}{\partial \mathbf{1}_{\partial\Omega_c}} \frac{\partial \mathbf{1}_{\partial\Omega_c}}{\partial \mu}}_{=0} + \underbrace{\frac{\partial R}{\partial \mathbf{1}_{\partial\Omega_f}} \frac{\partial \mathbf{1}_{\partial\Omega_f}}{\partial \mu}}_{=0} \quad (3.66)$$

Si la fonction coût Φ fait intervenir des termes de contact et/ou de frottement, pour calculer les termes $\frac{\partial \Phi}{\partial \mu}$, $\frac{\partial \Phi}{\partial X}$ et $\frac{\partial \Phi}{\partial W}$, on adoptera la même démarche que pour le calcul de $\frac{\partial R}{\partial X}$, en utilisant la décomposition suivante :

$$\Phi = \Phi_{\text{volumique}} + \Phi_{\text{frott}} + \Phi_{\text{contact}} \quad (3.67)$$

où $\Phi_{\text{volumique}}$ est la contribution volumique de Φ

Φ_{frott} est la contribution due au frottement de Φ

Φ_{contact} est la contribution due au contact de Φ

III.3. CALCUL DU GRADIENT POUR LE FORGEAGE MULTI-PASSES ET POUR L'OPTIMISATION DE FORME D'OUTILS

III.3.1. Problème lié au type de paramètres de forme du lopin initial : contrainte de volume constant à traiter et spécifique

Dans son travail de thèse, Laroussi [Laroussi 2003] a implémenté une première version de la méthode de l'état adjoint décrite précédemment dans le logiciel FORGE3® pour certaines fonctions coûts (Φ_{ener} énergie totale...). Cela lui a permis d'obtenir le gradient par rapport aux paramètres de type "forme du lopin initial" et avec une bonne précision. Cela nous permet d'utiliser les algorithmes à base de gradient (BFGS, SCIP, ...) ainsi que des algorithmes hybrides (AG-MGO, AG-MGA...) pour résoudre le problème d'optimisation d'engrenage qui va être décrit dans la partie d'application.

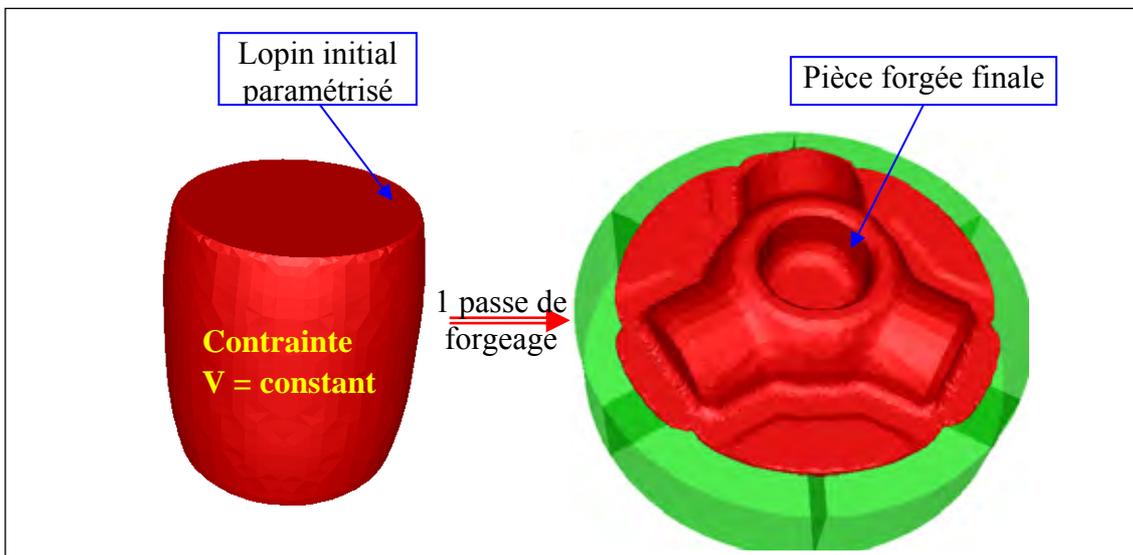


Figure 3.11. Exemple du cas forgeage à 1 passe avec le lopin initial paramétré

Cependant, ce type de paramètres est lié par la contrainte de volume constant, qui n'est pas facile à traiter. La méthode la plus simple est d'éliminer un des paramètres en utilisant la fonction analytique qui relie les paramètres. Cette fonction est facile à expliciter si la pièce est de forme simple (axisymétrique,...) et paramétrée par des polygones de bas degré. Au contraire, si la pièce est de forme compliquée (vrai 3D) ou paramétrée par des courbes de haut degré (par exemple une B-spline), la fonction analytique ne peut être explicitée. Cela n'empêche pas de la prendre en compte [Vieilledent 1999] mais le problème d'optimisation, et sa solution, dépendent du choix du paramètre éliminé, ce qui n'est pas parfaitement satisfaisant.

De plus, avec l'approche retenue dans [Laroussi 2003] le calcul du terme $\frac{dX_0}{d\mu}$ est très spécifique et dépendant de chaque cas d'optimisation. Le calcul du gradient des fonctions coûts par rapport au paramètre "forme du lopin initial" est donc "spécifique" pour chaque problème. Nous ne pouvons pas utiliser ce calcul d'un problème à l'autre.

Eviter la contrainte de volume constant et s'affranchir les spécificités du calcul du gradient est donc envisagée dans ce contexte. Cela constitue donc une contribution importante de cette thèse.

III.3.2. Paramétrisation des outils de préforme

Pour pouvoir aborder une gamme plus vaste de problèmes d'optimisation, nous nous orientons vers un autre type de paramètre de forme : la forme des outils dans le cadre du forgeage multi-passes.

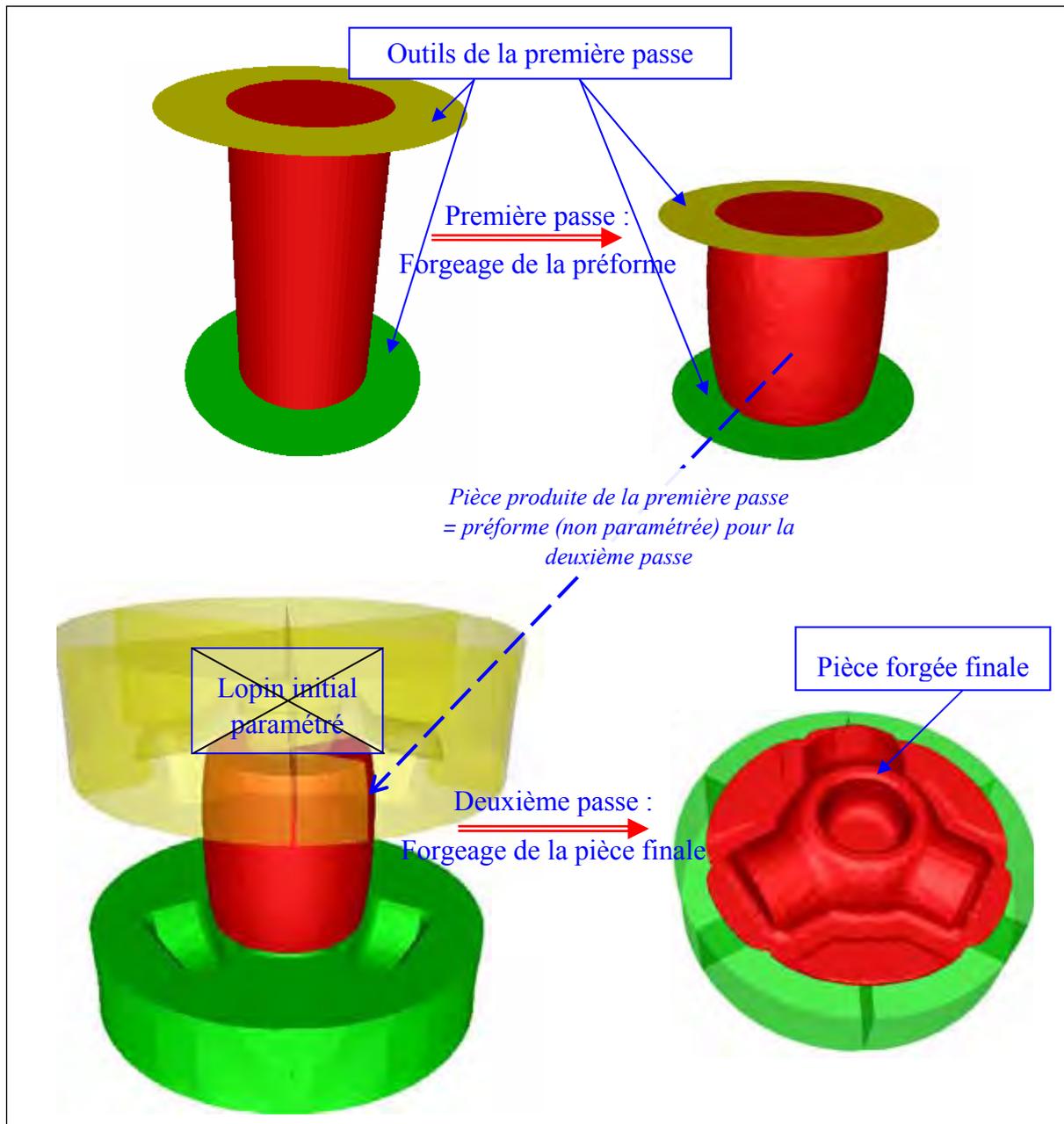


Figure 3.12. Exemple de forgeage en 2 passes avec outils de préforme paramétrés

Le forgeage multi-passes consiste à réaliser une pièce de forme compliquée en plusieurs opérations et avec des outils différents. La Figure 3.12 nous montre un exemple en deux

passes pour forger un triaxe à partir d'un lopin initial cylindrique. La pièce produite à l'étape actuelle est la préforme de l'étape suivante. La complexité des outils augmente à chaque opération de forgeage. Cette méthode de forgeage permet d'obtenir des pièces de forme compliquées à partir des lopins initiaux de formes très simples comme par exemple un lopin cylindrique (Figure 3.12).

Dans le cadre du forgeage en deux passes, la forme des outils de préforme est paramétrisée en considérant celle du lopin initial (lopin de la première opération de forgeage) comme fixée. Pour les cas avec un nombre d'opération supérieur à deux, les outils paramétrisés pourraient être ceux de la première opération ou ceux des opérations intermédiaires, selon la nature du problème.

La forme des outils de forgeage peut être paramétrée par deux approches selon sa complexité :

- par de vraies surfaces 3D, si la forme des outils est très complexe,
- par des courbes ou des polygones si l'outil est axisymétrique ou de forme régulière suivant une direction quelconque.

La paramétrisation des outils 3D est compliquée et laborieuse. De plus, dans de nombreux cas, la préforme est axisymétrique et seulement la forme finale est véritablement 3D. Dans un premier temps, nous considérons la paramétrisation des outils de forgeage de forme axisymétrique. Nous avons donc choisi les courbes Bsplines d'ordre 4 (degré 3) [Vielledent 1999] pour paramétrer ces formes.

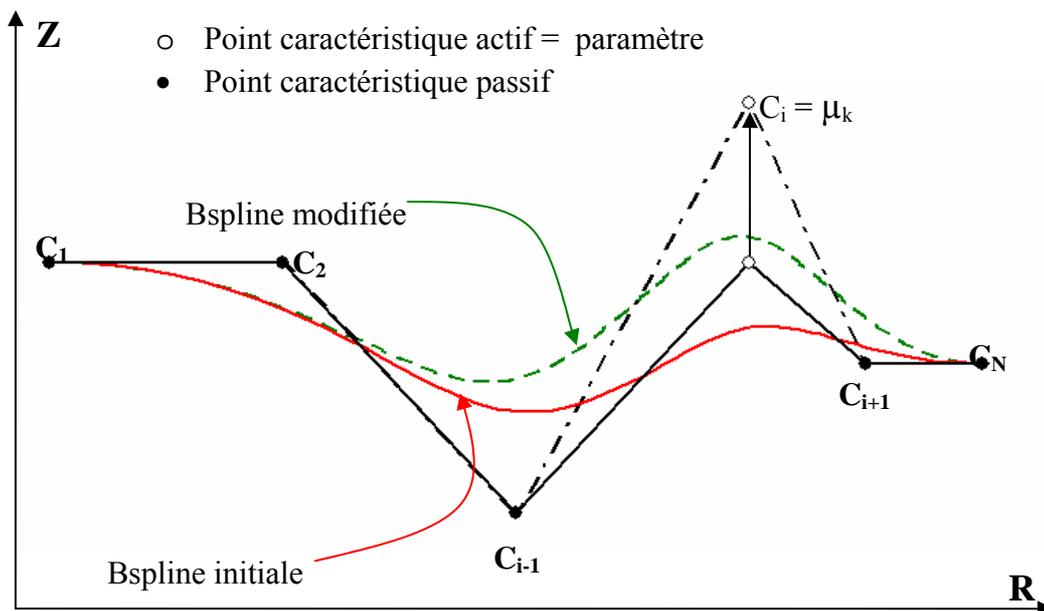


Figure 3.13. Exemple de courbe Bspline initiale et de courbe Bspline modifiée

Une courbe Bspline est définie par un contour polygonal comportant N ($N \geq 4$) sommets C_1, \dots, C_N , ou points caractéristiques ou encore points de contrôle. Ces sommets peuvent être fixes ou mobiles. Les sommets mobiles sont les paramètres d'optimisation. Le mouvement de ces sommets peut s'effectuer dans une direction quelconque, mais nous nous limitons dans le cas où les sommets ne se déplacent que dans la direction Z . La Figure 3.13 présente un

exemple de courbes Bsplines : la courbe initiale (en rouge) et sa forme modifiée (en verte) après changement de la position d'un point de contrôle. Les Bsplines sont choisies car elles présentent certains avantages : elles sont continues et souples, nécessitent peu de paramètres pour définir des géométries complexes en 2D, et elles sont faciles à contrôler.

Plusieurs étapes doivent encore être effectuées pour créer un outil depuis un contour polygonal initial. La *Figure 3.14* présente un exemple de ces étapes lors de la création d'un outil de préforme pour le forgeage d'un triaxe. Grâce à la symétrie du triaxe, nous étudions seulement un sixième de la moitié supérieur du triaxe. A partir d'un contour polygonal donné (*Figure 3.14.a*) qui contient plusieurs points de contrôle, certains étant des paramètres de forme, nous créons d'abord la Bspline correspondante (*Figure 3.14.b*). Ensuite, il faut effectuer une extrusion de la courbe Bspline suivant la direction axisymétrique et créer une matrice de connectivité pour générer le maillage de surface en 3D de l'outil (*Figure 3.14.c*). Puis, il faut lier les paramètres de contrôle à ce maillage pour générer un nouvel outil de forgeage (*Figure 3.14.d*). Enfin, cet outil sera utilisé dans le calcul du forgeage pour la valeur des fonctions coûts.

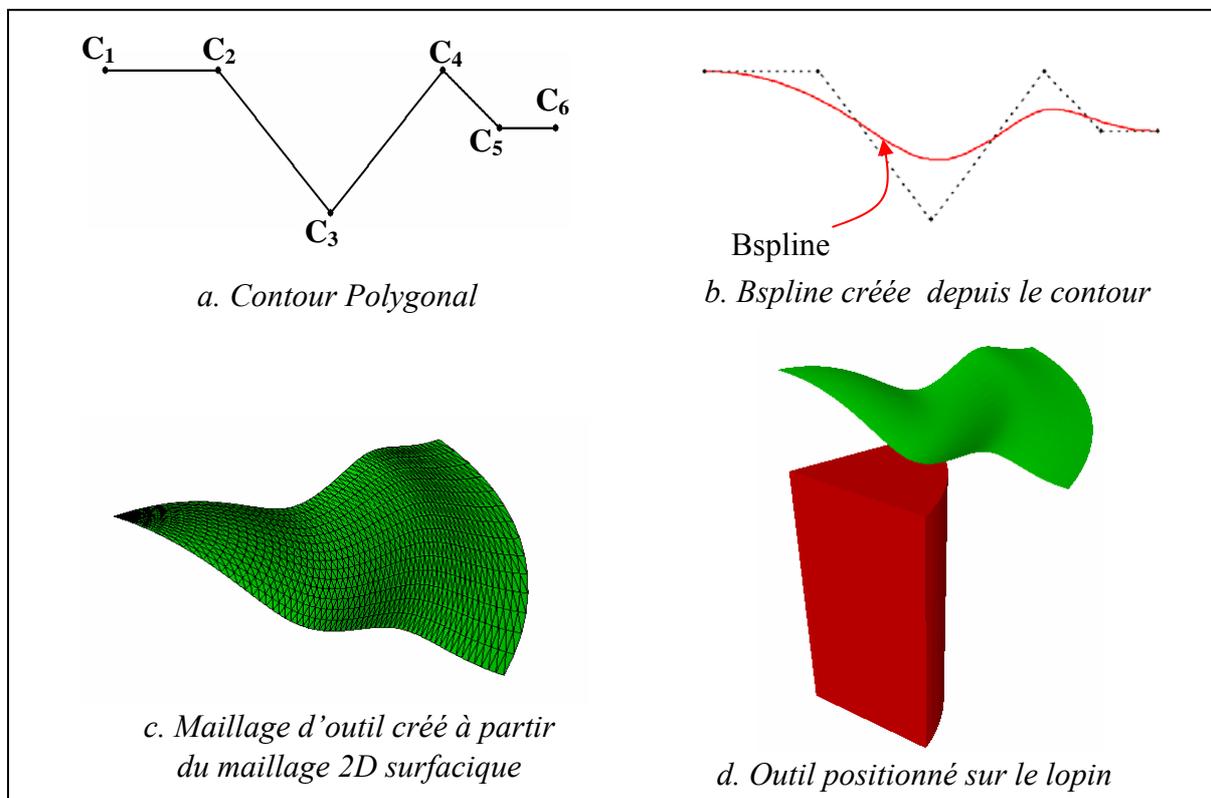


Figure 3.14. Création de l'outil de préforme pour le forgeage d'un triaxe à partir d'un contour polygonal

III.3.3. Calcul du gradient par rapport au paramètre de forme des outils

Les équations de base de la méthode de l'Etat Adjoint sont les mêmes lorsque l'on calcule le gradient des fonctions coûts par rapport au paramètre "forme des outils". Les changements interviennent dans l'expression de la dérivée totale.

Reprenons l'équation (3.49) pour calculer la dérivée totale d'une fonction coût Φ , nous avons :

$$\frac{d\Phi}{d\mu}(\mu) = \frac{\partial\Phi}{\partial\mu} + \sum_{i=0}^{N-1} \tau \lambda_i \frac{\partial R_i}{\partial\mu} + \Gamma_{-1} \frac{dX_0}{d\mu} \quad (3.68)$$

où rappelons-le X_0 représente les coordonnées des nœuds du maillage du lopin initial.

Le calcul des gradients présenté précédemment est assez spécifique aux paramètres de forme du lopin initial. La modification d'un paramètre μ quelconque dans le jeu de paramètres actuel engendre le changement de toutes les coordonnées X_0 du maillage du lopin initial. Par contre, le résidu R_i ne dépend pas explicitement de ce type de paramètres :

$$\frac{dX_0}{d\mu} \neq 0 \quad \text{et} \quad \frac{\partial R_i}{\partial\mu} = 0 \quad (3.69)$$

La dérivée totale de Φ est alors donnée par :

$$\frac{d\Phi}{d\mu}(\mu) = \frac{\partial\Phi}{\partial\mu} + \Gamma_{-1} \frac{dX_0}{d\mu} \quad (3.70)$$

Pour le forgeage multi-passes, en revanche on optimise la forme des outils de forgeage en considérant celle du lopin initial comme fixée (X_0 fixée). Cette fois, le résidu dépend explicitement des paramètres, mais pas de la préforme initiale et nous avons :

$$\frac{\partial R_i}{\partial\mu} \neq 0 \quad \text{et} \quad \frac{dX_0}{d\mu} = 0 \quad (3.71)$$

L'équation de la dérivée totale de Φ devient dans ce cas :

$$\frac{d\Phi}{d\mu}(\mu) = \frac{\partial\Phi}{\partial\mu} + \sum_{i=0}^{N-1} \tau \lambda_i \frac{\partial R_i}{\partial\mu} \quad (3.72)$$

Nous devons donc maintenant calculer $\frac{\partial R_i}{\partial\mu}$ qui intervient dans la dérivée totale de Φ . Ce terme est lui aussi calculé par différences finies. Dans (3.60), un seul terme de R_i dépend explicitement de la géométrie des outils, c'est le terme de contact (3.61). En effet, les distances de contact $\delta_k(X, \mu)$ et les normales $n_k(X, \mu)$ dépendent explicitement de la variation du paramètre μ . Ainsi, la dérivée du résidu à chaque incrément i se réduit à celle du terme de contact :

$$\frac{\partial R_i}{\partial\mu} = \frac{\partial R_{i,\text{contact}}}{\partial\mu} \quad (3.73)$$

Nous calculons donc le terme $\frac{\partial R_{i,\text{contact}}}{\partial\mu}$ à chaque incrément i par la méthode de différences finies. Pour cela, on effectue des perturbations des différentes composantes de μ d'une

quantité $\Delta\mu$ en supposant que les surfaces de contact $\partial\Omega_c$ et de frottement $\partial\Omega_f$ restent inchangées. On recalcule seulement les distances de contact $\delta_k^{pert}(X, \mu + \Delta\mu)$ et les normales $n_k^{pert}(X, \mu + \Delta\mu)$ perturbées en effectuant une nouvelle analyse de contact pour chaque configuration d'outillage perturbée. Toutefois, la perturbation des paramètres de forme des outils ainsi que l'analyse de contact à un incrément i du calcul de l'Etat Adjoint sont délicates. Elles nécessitent des précautions qui sont décrites sur la *Figure 3.15*.

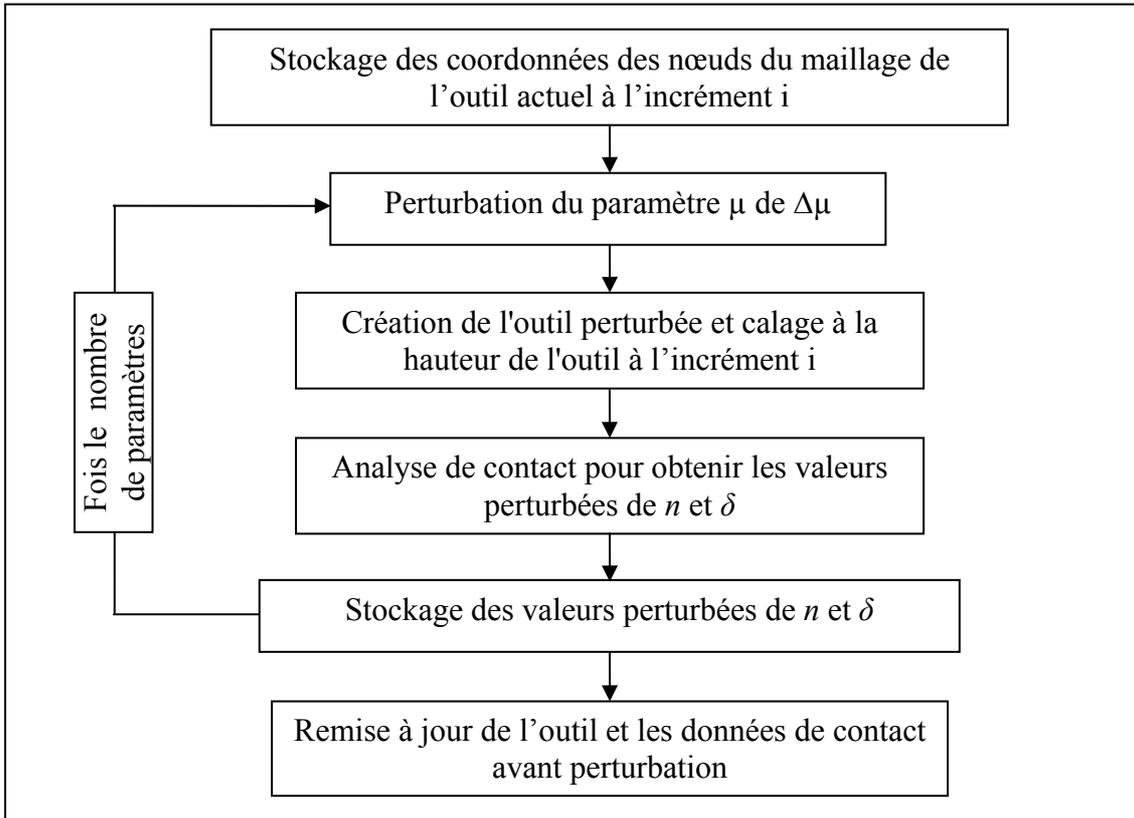


Figure 3.15. Algorithme pour calculer les valeurs perturbées de n et δ

Une fois les valeurs perturbées de n et δ obtenues, la valeur de $\frac{\partial R_i^k}{\partial \mu}$ est calculée par :

$$\forall k = 1, \dots, Nbnoe; \forall i = N - 1, \dots, 0$$

$$\frac{\partial R_i^k}{\partial \mu} = \frac{\partial R_{i\text{contact}}^k}{\partial \mu} = \frac{\rho 1_{\partial\Omega_c}(k) \left\{ \left[(V_k - V_{\text{outil}}) \cdot n_k^{pert} - \frac{\delta_k^{pert}}{\Delta t} \right] n_k^{pert} - \left[(V_k - V_{\text{outil}}) \cdot n_k - \frac{\delta_k}{\Delta t} \right] n_k \right\} S_k}{\Delta \mu} \quad (3.74)$$

En ce qui concerne $\frac{\partial \Phi}{\partial \mu}$ et les fonctions coûts utilisées, le terme Φ_{contact} de (3.67) est nul, de

sorte que $\frac{\partial \Phi}{\partial \mu} = 0$. Dans un cas plus général $\frac{\partial \Phi_{\text{contact}}}{\partial \mu}$ serait calculé de la même manière que

$$\frac{\partial R_{\text{contact}}}{\partial \mu}$$

III.3.4. Transport des variables de l'état adjoint entre deux opérations de forgeage

Comme nous l'avons mentionné précédemment, le calcul de l'état adjoint est effectuée dans l'ordre inverse du temps. Il n'est déclenché que lorsque toutes les simulations directes du FORGE3® sont terminées.

Pour le forgeage en une passe, la gestion des variables de l'Etat Adjoint est présentée dans [Laroussi 2003]. En revanche, pour le forgeage multi-passes, la dérivée totale est le résultat de plusieurs calculs séquentiels de l'état adjoint (chaque opération de forgeage est un calcul adjoint complet comme pour le forgeage en une passe). Donc, nous devons transporter les variables entre deux opérations séquentielles. Lors de l'analyse les équations, nous constatons

que seulement la variable Γ avec la valeur de la dérivée totale $\left. \frac{d\Phi}{d\mu} \right|^K$ après chaque opération K ont besoin d'être transportées. Supposons que le forgeage consiste en M passes. Après un calcul de l'état adjoint de la passe K, pour poursuivre avec celui à la passe K-1, les données sont transportées en plusieurs étapes comme présentées sur la *Figure 3.16*.

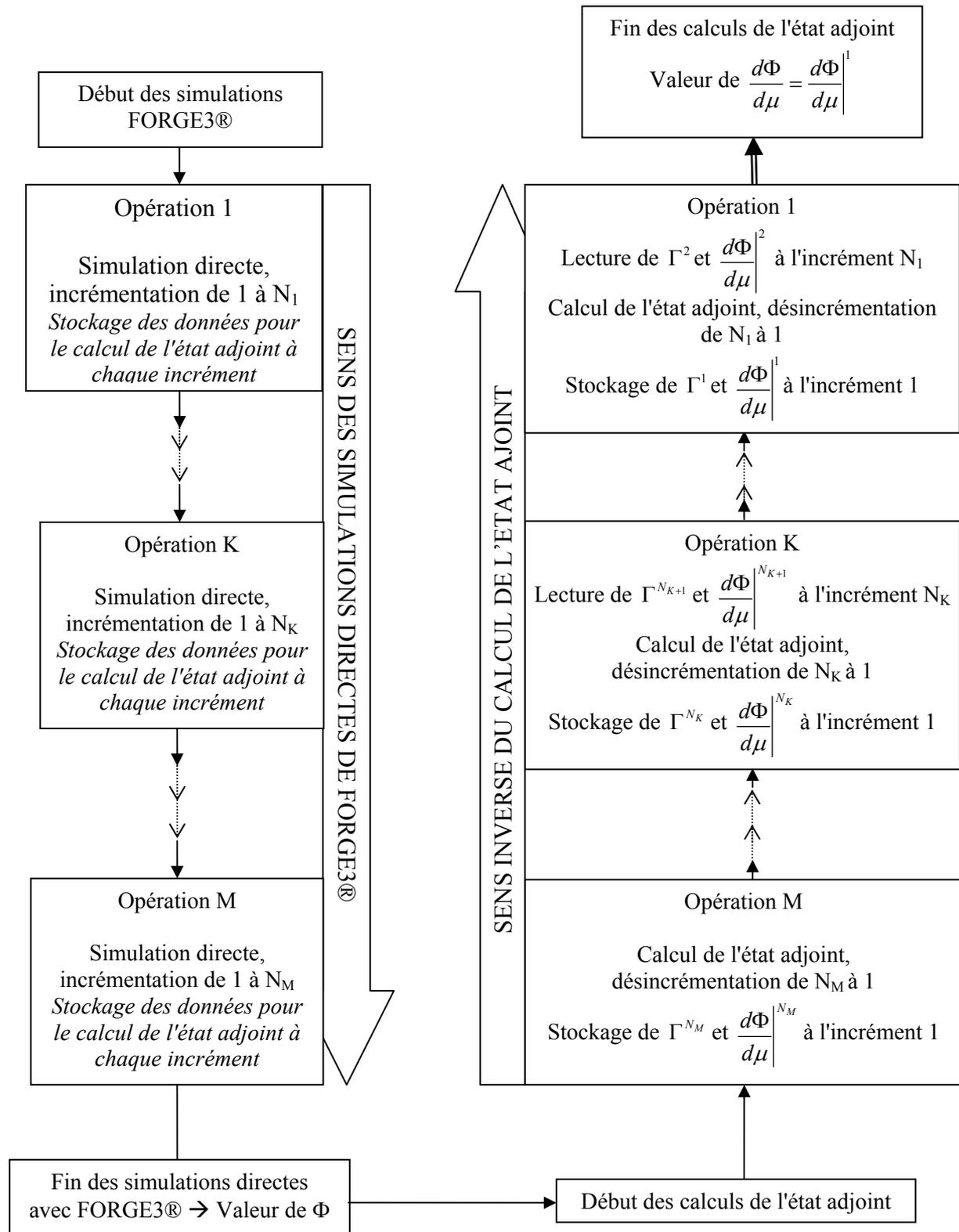


Figure 3.16. Sens des simulations directes et des calculs de l'état adjoint dans un cas de forgeage composé de M passes avec transfert des données entre deux passes.

III.3.5. Validation du calcul du gradient pour le forgeage multi-passes

La validation du calcul de ces dérivées par rapport aux paramètres de type "préforme du lopin initial" a été faite dans [Laroussi 2003]. Ici, nous présentons la validation du calcul du gradient par rapport au paramètre de la forme des outils, dans le cas du forgeage multi-passes, avec outil axisymétrique paramétré par des fonctions Bsplines.

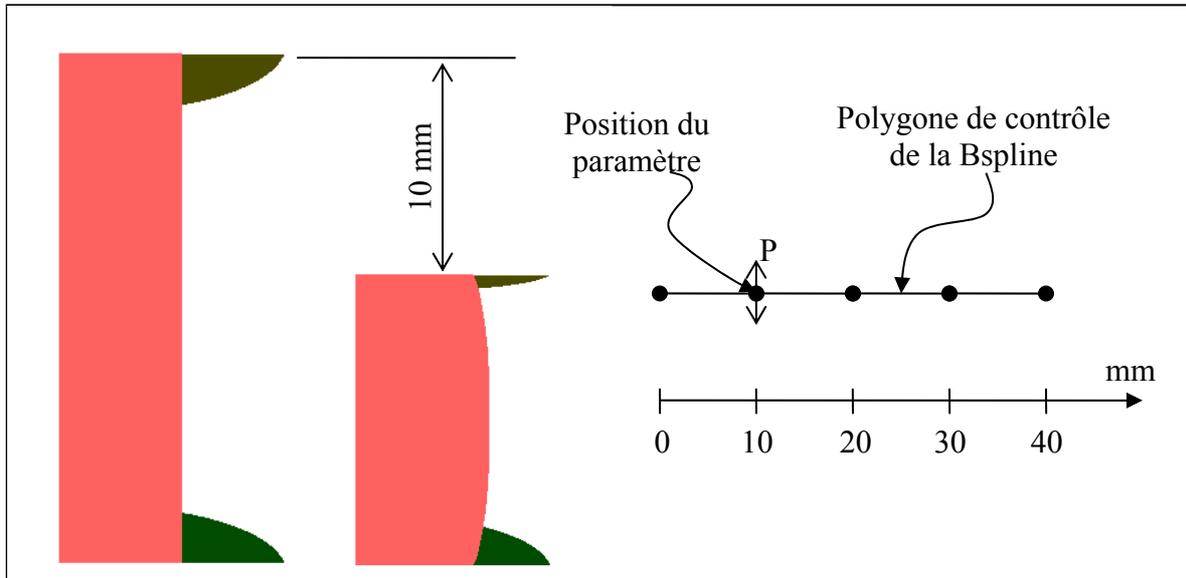


Figure 3.17. Test d'écrasement d'un lopin cylindrique et position du paramètre sur le polygone de contrôle de la Bspline

Nous validons d'abord le nouveau type de paramètre "forme d'outil" en réalisant un test d'écrasement d'un lopin cylindrique. Comme ce lopin est symétrique, nous simulons seulement l'écrasement d'un sixième de ce lopin. La paramétrisation de l'outil avec les courbes Bspline a été présentée dans la section III.3.2. La position du paramètre considéré est présentée sur la Figure 3.17 (le rayon du lopin est de 20mm). La course d'écrasement est de 10mm (la hauteur du lopin est de 90mm). La valeur du gradient obtenue est comparée à celle d'un calcul par différences finies.

Pour valider le calcul du gradient dans le cas multi-passes, nous décomposons le test d'écrasement précédent en deux opérations de forgeage, chaque passe ayant une course de forgeage de 5mm (Figure 3.18). Dans ce cas, nous paramétrons les outils des deux opérations de forgeage. Si le calcul du gradient est précis, la valeur calculée doit être la même que celle obtenue en une seule passe de 10mm. La validation va être effectuée pour les deux fonctions coûts "énergie totale" du forgeage et "défaut de repli".

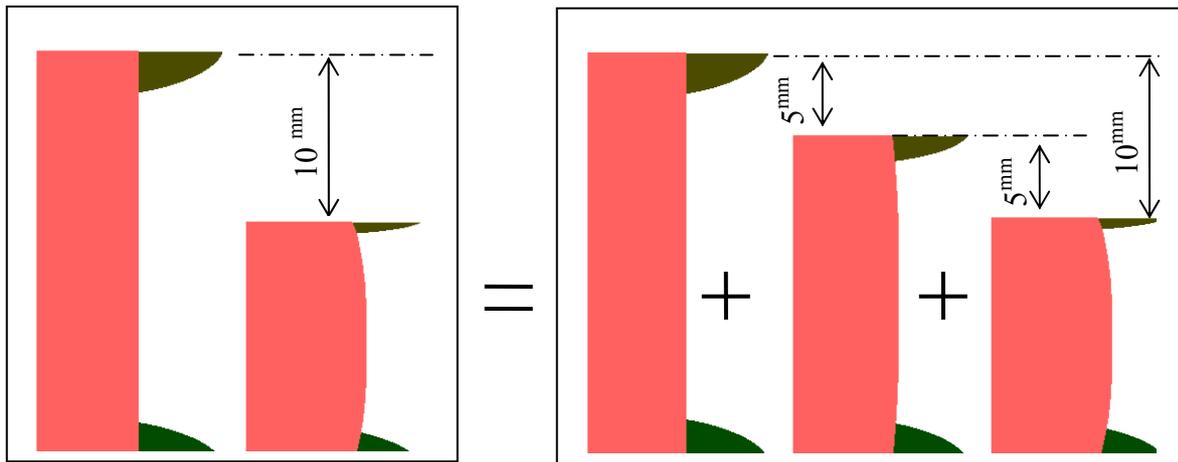


Figure 3.18. Décomposition du test d'écrasement pour valider le calcul du gradient lors du forgeage multi-passes

III.3.5.1. Validation pour la fonction coût énergie totale

Les résultats du calcul des dérivées de la fonction coût énergie totale de mise en forme par rapport au paramètre "forme d'outils de préforme" sont présentés dans le *Tableau 3.3*. Ils sont obtenus en utilisant la méthode de l'Etat Adjoint ainsi que la méthode des différences finies. Pour la méthode des différences finies, la perturbation choisie est égale à 10^{-5} en valeur relative. Nous voyons que l'erreur relative entre les dérivées obtenues avec ces deux méthodes de calcul dans le forgeage à une passe est faible (de l'ordre de 0.2%) et de l'ordre de grandeur de l'erreur du calcul Elément Finis. Le gradient de la fonction coût énergie totale issu de la méthode de l'Etat Adjoint est donc précis.

Valeur de dérivée de Φ_{ener} donnée par méthode	$\frac{d\Phi}{d\mu}$	$\frac{d\Phi/d\mu - d\Phi/d\mu _{DF1\ passe}}{d\Phi/d\mu _{DF1\ passe}}$	$\frac{d\Phi/d\mu - d\Phi/d\mu _{EA1\ passe}}{d\Phi/d\mu _{EA1\ passe}}$
Différences finies 1 passe	7.80×10^{12}	0%	0.20%
Etat Adjoint 1 passe	7.78×10^{12}	-0.20 %	0%
Etat Adjoint 2 passes	7.78×10^{12}	-0.19%	0.012%

Tableau 3.3 : Comparaison des dérivées de la fonction coût « énergie totale » obtenues par la méthode de l'Etat Adjoint et par la méthode des différences finies.

En comparant les dérivées obtenues avec la méthode de l'Etat Adjoint dans les deux cas forgeage, en une passe et en deux passes, nous obtenons presque la même valeur (l'ordre de grandeur de l'erreur relative est de 0.012%). Si nous comparons la dérivée obtenue avec la méthode de l'Etat Adjoint en 2 passes avec celle obtenue par la méthode des différences finies, nous obtenons aussi une précision de 0.19% (l'ordre de la précision des calculs éléments finis).

III.3.5.2. Validation pour la fonction coût repli

Pour la fonction coût « repli », les mêmes tests sont effectués. Les dérivées obtenues sont présentées dans le *Tableau 3.4*.

Valeur de dérivée de Φ donnée par méthode	$\frac{d\Phi}{d\mu}$	$\frac{d\Phi/d\mu - d\Phi/d\mu _{DF1\ passe}}{d\Phi/d\mu _{DF1\ passe}}$	$\frac{d\Phi/d\mu - d\Phi/d\mu _{EA1\ passe}}{d\Phi/d\mu _{EA1\ passe}}$
Différences finies 1 passe	-1.024×10^{-1}	0%	-0.58%
Etat Adjoint 1 passe	-1.03×10^{-1}	0.58 %	0%
Etat Adjoint 2 passes	-1.028×10^{-1}	0.39%	-0.19%

Tableau 3.4 : Comparaison des dérivées de la fonction coût « repli » obtenues par la méthode de l'Etat Adjoint avec celles obtenue par la méthode des différences finies.

Le *Tableau 3.4* nous permet de valider le calcul du gradient de la fonction coût «repli». En une passe de forgeage, lors de la comparaison des valeurs du gradient obtenues par la méthode de l'Etat Adjoint et par celle des différences finies, nous voyons que la dérivée est précise (l'erreur relative inférieure à 1%). En comparant les dérivées calculées par la méthode de l'Etat Adjoint dans les deux cas de forgeage en une passe et en deux passes, nous obtenons une erreur relative de 0.19%. Cette erreur est bien satisfaisante pour notre calcul.

III.4. BILAN ET CONCLUSIONS

En conclusion, dans le cadre de ce travail, nous voyons que :

- ↪ Une fonction coût adimensionalisée qui permet de mieux détecter le défaut de repli, est proposée.
- ↪ Un autre type de paramètres de forme moins spécifique, et qui permet d'aborder une gamme plus vaste de problème d'optimisation de forme en forgeage, est introduit. C'est celui de la "forme de l'outil de préforme".
- ↪ Le calcul du gradient des deux fonctions coûts "énergie totale" et "défaut de repli" est donc étendu pour ce type de paramètre. Le gradient obtenu est précis, pour le forgeage en une et plusieurs passes.

On pourra donc utiliser ces résultats dans la résolution des problèmes d'optimisation de forme.