# Méthodes de pilotage du chargement

**Résumé :**

Ce document décrit les méthodes de pilotage du chargement disponibles dans *Code_Aster* (par un degré de liberté, par longueur d'arc, par incrément de déformation et par prédiction élastique). Elles introduisent une inconnue supplémentaire, l'intensité de la part pilotable du chargement, et une équation supplémentaire, la contrainte de pilotage. Ces méthodes permettent en particulier de calculer la réponse d'une structure qui présenterait des instabilités, aussi bien d'origines géométrique (flambement) que matériau (adoucissement).



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

# 1 Introduction

## 1.1 Non-linéarité et échec de Newton

La méthode de Newton présentée dans [R5.03.01] échoue sur certains problèmes qui exhibent une réponse de la structure non strictement monotone en fonction des déplacements ou des chargements imposés. Sur la figure 1.1-a, on présente le cas où, pour un niveau de chargement donné, il existe plusieurs déplacements solutions. Ce problème n'est donc pas soluble avec Newton si l'on pilote le problème en chargement.

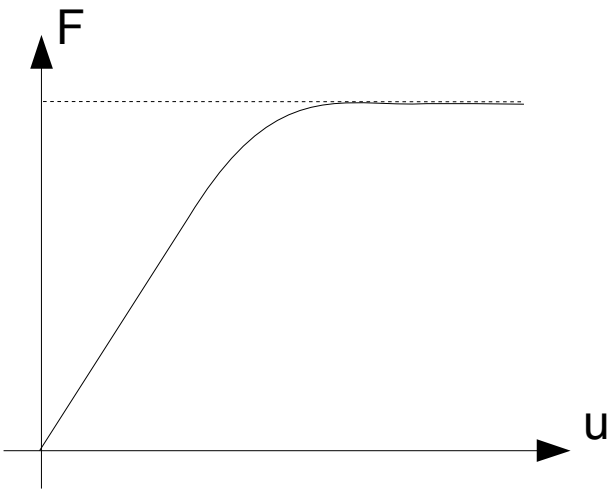



**Figure 1.1-a: Réponse non-monotone en fonction du déplacement**

Un second cas (figure 1.1-b)concerne les problèmes non-linéaires géométriques (typiquement le flambement des coques minces et les problèmes de snap-back « doux »). Dans ce cas, pour un niveau de chargement, on a plusieurs solutions en déplacement et pour un niveau de déplacement on a plusieurs chargements possibles. Ce problème n'est pas soluble directement par la méthode de Newton, que ce soit en pilotant en chargement ou en déplacement.

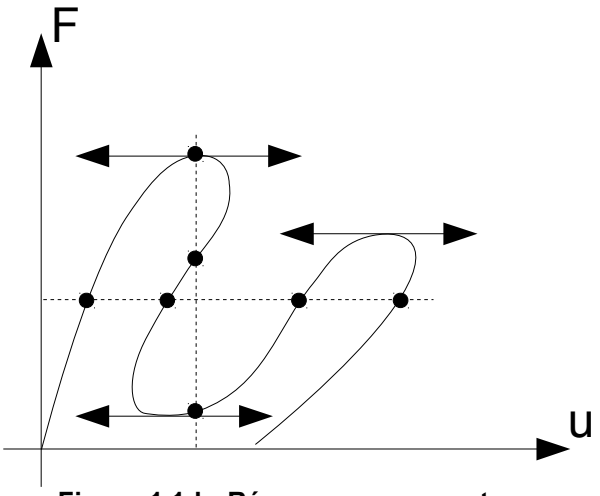



**Figure 1.1-b: Réponse non-monotone en déplacement et en chargement**

Le troisième cas (figure 1.1-c) décrit les problèmes d'endommagements dans lesquels de brusques retours élastiques (correspondant à l'endommagement progressif du matériau) rendent la courbe très irrégulière (snap-backs « brusques »)



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

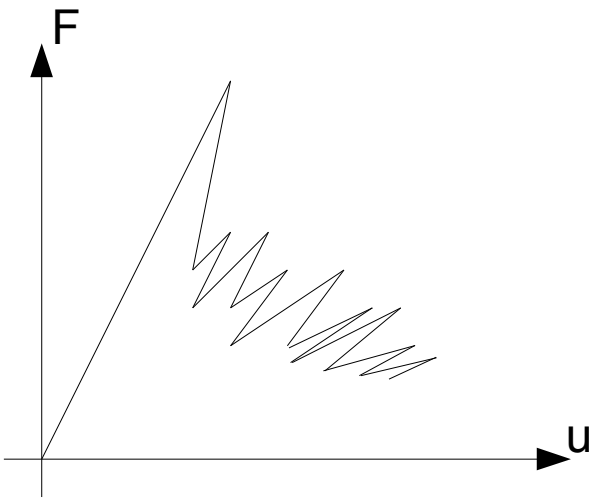



**Figure 1.1-c: Réponse avec pertes de rigidité**

Enfin, il existe enfin toute une catégorie de problèmes dans lesquels la perte d'ellipticité du problème crée des bifurcations et donc des branches de solutions différentes (voir figure 1.1-d). Dans ce cas, ce qui intéresse le plus l'ingénieur c'est généralement la branche dissipative et non la branche élastique. Nous verrons qu'il existe des techniques pour sélectionner les « bonnes » branches.

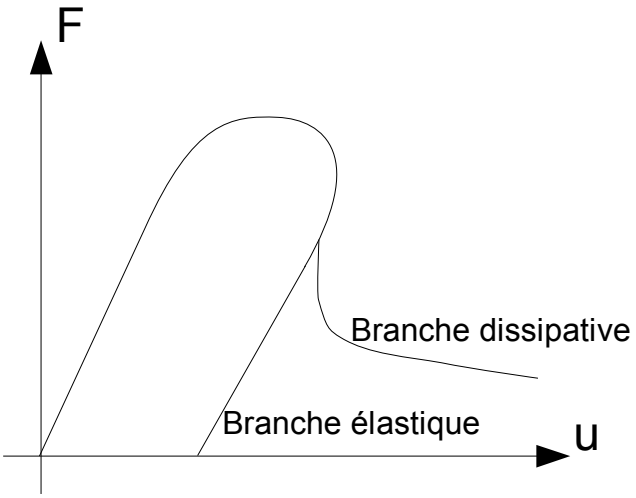



**Figure 1.1-d: Réponse avec branches multiples**

## 1.2 Problèmes avec intensité du chargement inconnu

Le pilotage peut aussi servir dans le cas ou le problème exhibe, par sa mise en données naturelle, une inconnue supplémentaire qui est l'intensité du chargement appliqué. Le pilotage permet de traiter le cas où seule la direction et le point d'application du chargement sont connus, l'intensité restant une inconnue du problème. Par exemple, sur la figure 1.2-a, on voit le cas d'un câble tendu entre deux pylônes. On sait qu'il faut appliquer un chargement qui tire le câble (on connait donc le point d'application et la direction du chargement) mais on ignore l'intensité $\eta$ à appliquer pour obtenir une flèche $f$ donnée.



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

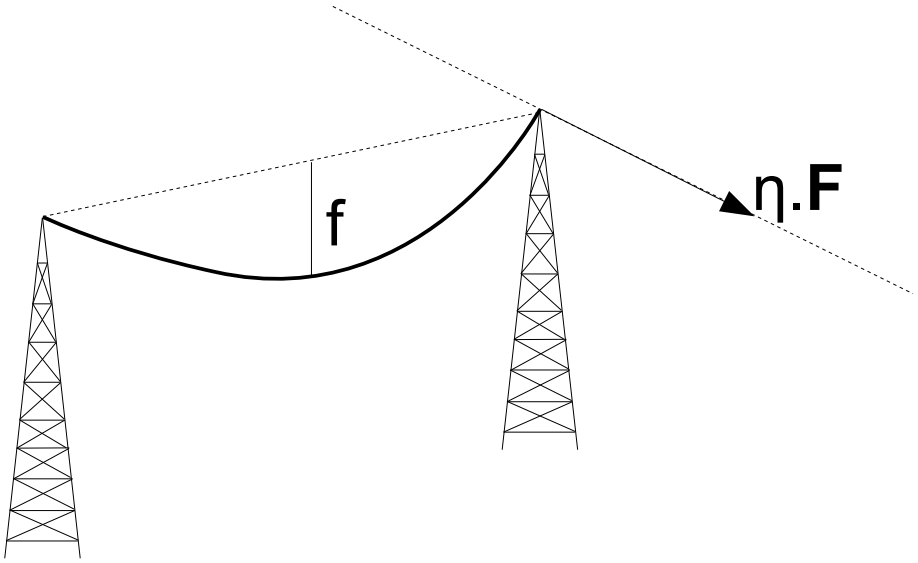



**Figure 1.2-a: Exemple de chargement partiellement inconnu**

# 2 Principe des méthodes de pilotage du chargement

De manière générale, les fonctionnalités de pilotage disponibles dans *Code_Aster* permettent de déterminer *l'intensité d'une partie du chargement* pour satisfaire une contrainte portant sur les déplacements. Leur emploi est limité à des simulations pour lesquelles le temps ne joue pas de rôle physique, ce qui exclut *a priori* les problèmes dynamiques ou visqueux. Il est également incompatible avec les problèmes exprimant une condition d'unilatéralité (contact et frottement). On peut distinguer plusieurs gammes d'utilisation auxquelles répondent autant de méthodes de pilotage (mot clé facteur `PILOTAGE`) :

- Contrôle physique des forces par le déplacement d'un point de la structure: pilotage par degré de liberté imposé (`TYPE ='DDL_IMPO'`) ;
- Suivi d'instabilités géométriques (flambement), la réponse de la structure pouvant exhiber des snap-back « doux » : pilotage par longueur d'arc (`TYPE ='LONG_ARC'`) ;
- Suivi d'instabilités matériaux (en présence de lois de comportement adoucissantes), la réponse de la structure pouvant exhiber des snap-back « brutaux » : pilotage par la prédiction élastique (`TYPE = 'PRED_ELAS'`) ou plus généralement par l'incrément de déformation, (`TYPE ='DEFORMATION') ;`
- Calcul des charges limites des structures `(TYPE='ANA_LIM')`, cf. [R7.07.01].

Plus précisément, les méthodes de pilotage disponibles dans *Code_Aster* reposent sur les deux idées suivantes.

D'une part, on considère que le chargement (forces extérieures et déplacements donnés) se décompose additivement en deux termes, l'un connu et imposé par l'utilisateur ( $\mathbf{L}_{\text{impo}}^{\text{méca}}$ et $\mathbf{u}_{\text{impo}}^{d}$ ) et l'autre ( $\mathbf{L}_{\text{pilo}}^{\text{méca}}$ et $\mathbf{u}_{\text{pilo}}^{d}$ ) dont seule la direction est connue, son intensité $\eta$ devenant une nouvelle inconnue du problème :

$$\begin{cases} \mathbf{L}^{\text{méca}} = \mathbf{L}_{\text{impo}}^{\text{méca}} + \eta . \mathbf{L}_{\text{pilo}}^{\text{méca}} \\ \mathbf{u}^{d} = \mathbf{u}_{\text{impo}}^{d} + \eta . \mathbf{u}_{\text{pilo}}^{d} \end{cases} \quad (1)$$

D'autre part, afin de pouvoir résoudre le problème, on lui associe une nouvelle équation qui porte sur les déplacements et qui dépend de l'incrément de temps : c'est la contrainte de pilotage, qui s'exprime par :

$$P(\Delta \mathbf{U}) = \Delta\tau \quad \text{avec} \quad P(0) = 0 \quad (2)$$



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

où $\Delta\tau$ est indirectement une donnée utilisateur qui s'exprime via le pas de temps courant $\Delta t$ et un coefficient de pilotage (COEF_MULT) tel que :

$$\Delta\tau = \frac{\Delta t}{\text{COEF_MULT}} \tag{3}$$

La condition $P(0)=0$ est requise afin d'obtenir un incrément de déplacement d'autant plus petit que le pas de temps est petit. Finalement, les inconnues du problème deviennent les déplacements $\mathbf{u}$, les multiplicateurs de Lagrange $\boldsymbol{\lambda}$ associés aux conditions aux limites et l'intensité du chargement piloté $\eta$, baptisée ETA_PILOTAGE. Le système non linéaire à résoudre s'écrit dorénavant :

$$\begin{cases} \mathbf{L}^{\text{int}}(\mathbf{u}) + \mathbf{B}^T.\boldsymbol{\lambda} &= \mathbf{L}^{\text{méca}}_{\text{impo}} + \eta.\mathbf{L}^{\text{méca}}_{\text{pilo}} \\ \mathbf{B}.\mathbf{u} &= \mathbf{u}^d_{\text{impo}} + \eta.\mathbf{u}^d_{\text{pilo}} \\ P(\Delta\mathbf{u}) &= \Delta\tau \end{cases} \tag{4}$$

**Remarques :**

*A l'heure actuelle, les chargements suiveurs (i.e. qui dépendent des déplacements) et les conditions de Dirichlet de type « DIDI » ne sont pas pilotables.*
*Le chargement ne dépend plus directement du temps mais résulte de la résolution de tout le système non linéaire (4). Cela implique que la part pilotée du chargement ne doit pas dépendre du temps physique, mais correspond à un effort que l'on ajuste pour satisfaire une contrainte cinématique supplémentaire.*

# 3 Résolution du système global

L'introduction d'une nouvelle équation ne perturbe pas outre mesure la méthode de résolution du système non linéaire. En effet, on procède comme en [R5.03.01], c'est-à-dire que la résolution est incrémentale. On note le pas de temps $i$ en indice et l'itération de Newton $n$ en exposant. Le problème non-linéaire est alors résolu en deux temps :

- Une phase de prédiction qui donne une première estimation des déplacements et des multiplicateurs de Lagrange notée $(\Delta\mathbf{u}_i^0, \Delta\boldsymbol{\lambda}_i^0)$ ;
- Une phase de correction de Newton $(\delta\mathbf{u}_i^n, \delta\boldsymbol{\lambda}_i^n)$ qui vient corriger cette première estimation;

Avec un nombre $n_{CV}$ suffisant d'itérations de Newton, on obtient un résultat convergé[1]:

$$\begin{cases} \mathbf{u}_i^{\text{convergé}} = \mathbf{u}_{i-1} + \Delta\mathbf{u}_i^0 + \sum_{j=1}^{n=n_{CV}} \delta\mathbf{u}_i^j \\ \boldsymbol{\lambda}_i^{\text{convergé}} = \boldsymbol{\lambda}_{i-1} + \Delta\boldsymbol{\lambda}_i^0 + \sum_{j=1}^{n=n_{CV}} \delta\boldsymbol{\lambda}_i^j \end{cases} \tag{5}$$

Le principe est donc de linéariser le système (4) que l'on écrit au pas de temps $i$ :

$$\begin{cases} \mathbf{L}_i^{\text{int}} + \mathbf{B}^T.\boldsymbol{\lambda}_i &= \mathbf{L}^{\text{méca}}_{\text{impo},i} + \eta_i.\mathbf{L}^{\text{méca}}_{\text{pilo},i} \\ \mathbf{B}.\mathbf{u}_i &= \mathbf{u}^d_{\text{impo},i} + \eta_i.\mathbf{u}^d_{\text{pilo},i} \\ P(\Delta\mathbf{u}_i) &= \Delta\tau_i \end{cases} \tag{6}$$

Il n'y aura pas de linéarisation par rapport à la variable de pilotage $\eta_i$. De la sorte, on préserve toute la méthodologie de réactualisation de l'opérateur tangent déjà mise en œuvre pour les calculs sans pilotage. De plus, la structure « bande » de la matrice tangente est conservée. Le schéma de résolution n'est donc plus strictement une méthode de Newton (ce qui n'est pas gênant car nous verrons que l'équation de pilotage est, au plus, de degré deux). En prédiction, si on linéarise le système (6) sans l'équation de

1 La notion de résultat « convergé » est plus amplement détaillée dans [R5.03.01].



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

pilotage, par rapport au temps autour de $(\mathbf{u}_{i-1}, \boldsymbol{\lambda}_{i-1})$, on obtient (voir tout le développement dans [R5.03.01]):

$$\begin{cases} \mathbf{K}_{i-1}.\Delta\mathbf{u}_i^0+\mathbf{B}^T.\Delta\boldsymbol{\lambda}_i^0 & =\mathbf{L}_{\text{impo},i}^{\text{méca}}+\eta_i.\mathbf{L}_{\text{pilo},i}^{\text{méca}}-\mathbf{Q}_{i-1}^T.\boldsymbol{\sigma}_{i-1}+\Delta\mathbf{L}_i^{\text{varc}} \\ \mathbf{B}.\Delta\mathbf{u}_i^0 & =\mathbf{u}_{\text{impo},i}^d+\eta_i.\mathbf{u}_{\text{pilo},i}^d-\mathbf{B}.\mathbf{u}_{i-1} \end{cases} \quad (7)$$

En correction, on linéarise le système (6) toujours sans l'équation de pilotage, par rapport au temps, mais autour de $(\mathbf{u}_i^n, \boldsymbol{\lambda}_i^n)$, on a :

$$\begin{cases} \mathbf{K}_i^{n-1}.\delta\mathbf{u}_i^n+\mathbf{B}^T.\delta\boldsymbol{\lambda}_i^n & =\mathbf{L}_{\text{impo},i}^{\text{méca}}+\eta_i.\mathbf{L}_{\text{pilo},i}^{\text{méca}}-\mathbf{L}_i^{\text{int},n-1}-\mathbf{B}^T.\boldsymbol{\lambda}_i^{n-1} \\ \mathbf{B}.\delta\mathbf{u}_i^n & =\eta_i.\mathbf{u}_{\text{pilo},i}^d \end{cases} \quad (8)$$

On va réunir les deux systèmes en une écriture commune, afin de simplifier l'exposé. Le système à résoudre s'écrit finalement:

$$\begin{bmatrix} \mathbf{K}_i^{n-1} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix}.\begin{Bmatrix} \delta\mathbf{u}_i^n \\ \delta\boldsymbol{\lambda}_i^n \end{Bmatrix}=\begin{Bmatrix} \mathbf{L}_{\text{impo},i}^{n-1} \\ \mathbf{u}_{\text{impo},i} \end{Bmatrix}+\eta_i.\begin{Bmatrix} \mathbf{L}_{\text{pilo},i} \\ \mathbf{u}_{\text{pilo},i} \end{Bmatrix} \quad (9)$$

On peut faire plusieurs remarques:

- La matrice $\mathbf{K}_i^{n-1}$ dépend à la fois du pas de temps courant et éventuellement de l'itération de Newton précédente. Les différentes manières de la construire (quasi-Newton, élastique, sécante, cohérente, tangente en vitesse, etc.) sont décrites dans [R5.03.01];
- On a supposé que les chargements extérieurs étaient linéaires (ils ne dépendent que du pas de temps). On ne considère donc pas les chargements suiveurs tels que la pression ou la force centrifuge bien que du point de vue théorique, ça ne pose pas de difficultés. Par contre, le matériau peut être décrit avec un comportement non-linéaire, ce qui implique que $\mathbf{L}_i^{\text{int},n-1}$ dépend de l'itération de Newton (résultat de la linéarisation des efforts intérieurs).
- Les conditions limites de Dirichlet sont toujours linéaires, ce qui permet à la matrice $\mathbf{B}$ d'être constante sur tout le transitoire.
- Avec le bon choix de la matrice tangente et du second membre, formellement, il y a équivalence entre $(\delta\mathbf{u}_i^{n=0}, \delta\boldsymbol{\lambda}_i^{n=0})$ et l'incrément en prédiction $(\Delta\mathbf{u}_i^0, \Delta\boldsymbol{\lambda}_i^0)$.

On peut maintenant exprimer les corrections de déplacements $\delta\mathbf{u}_i^n$ et de multiplicateurs de Lagrange $\delta\boldsymbol{\lambda}_i^n$ en fonction de $\eta_i$ moyennant la résolution du système linéaire (9) par rapport à chacun des deux seconds membres. C'est-à-dire que l'on sépare les deux solutions:

$$\begin{Bmatrix} \delta\mathbf{u}_i^n \\ \delta\boldsymbol{\lambda}_i^n \end{Bmatrix}=\begin{Bmatrix} \delta\mathbf{u}_{\text{impo},i}^n \\ \delta\boldsymbol{\lambda}_{\text{impo},i}^n \end{Bmatrix}+\eta_i.\begin{Bmatrix} \delta\mathbf{u}_{\text{pilo},i}^n \\ \delta\boldsymbol{\lambda}_{\text{pilo},i}^n \end{Bmatrix} \quad (10)$$

Ces deux solutions correspondent au découplage des deux chargements:

$$\begin{Bmatrix} \mathbf{L}_i^{n-1} \\ \mathbf{u}_i \end{Bmatrix}=\begin{Bmatrix} \mathbf{L}_{\text{impo},i}^{n-1} \\ \mathbf{u}_{\text{impo},i} \end{Bmatrix}+\eta_i.\begin{Bmatrix} \mathbf{L}_{\text{pilo},i} \\ \mathbf{u}_{\text{pilo},i} \end{Bmatrix} \quad (11)$$

On résout indépendamment[2]:

$$\begin{bmatrix} \mathbf{K}_i^{n-1} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix}.\begin{Bmatrix} \delta\mathbf{u}_{\text{impo},i}^n \\ \delta\boldsymbol{\lambda}_{\text{impo},i}^n \end{Bmatrix}=\begin{Bmatrix} \mathbf{L}_{\text{cst},i}^{n-1} \\ \mathbf{u}_{\text{cst},i} \end{Bmatrix} \quad (12)$$

Et:

$$\begin{bmatrix} \mathbf{K}_i^{n-1} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix}.\begin{Bmatrix} \delta\mathbf{u}_{\text{pilo},i}^n \\ \delta\boldsymbol{\lambda}_{\text{pilo},i}^n \end{Bmatrix}=\begin{Bmatrix} \mathbf{L}_{\text{pilo},i}^{n-1} \\ \mathbf{u}_{\text{pilo},i} \end{Bmatrix} \quad (13)$$

2 En pratique, la factorisation de la matrice n'est faite qu'une fois et l'on résout simultanément les deux systèmes.



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

On peut maintenant substituer la correction de déplacement $\Delta \mathbf{u}_i^n$ dans l'équation de contrôle du pilotage du système; il en résulte une équation scalaire en $\eta_i$ :

$$\tilde{P}\left(\eta_i\right) \underset{\text{déf.}}{=} P\left(\Delta \mathbf{u}_i^{n-1} + \delta \mathbf{u}_{\text{impo},i}^n + \eta_i \,.\, \delta \mathbf{u}_{\text{pilo},i}^n\right) = \Delta\tau_i \tag{14}$$

La méthode de résolution de cette équation dépend de la nature du contrôle de pilotage adopté cf. [§7]. Finalement, il ne reste plus qu'à réactualiser les inconnues déplacements et multiplicateurs de Lagrange :

$$\begin{cases} \Delta \mathbf{u}_i^n = \Delta \mathbf{u}_i^{n-1} + \delta \mathbf{u}_{\text{impo},i}^n + \eta_i \,.\, \delta \mathbf{u}_{\text{pilo},i}^n \\ \boldsymbol{\lambda}_i^n = \boldsymbol{\lambda}_i^{n-1} + \delta \boldsymbol{\lambda}_{\text{impo},i}^n + \eta_i \,.\, \delta \boldsymbol{\lambda}_{\text{pilo},i}^n \end{cases} \tag{15}$$

# 4 Équation de contrôle du pilotage

## 4.1 Pilotage par contrôle d'un degré de liberté des déplacements : DDL_IMPO

Pour ce premier type de pilotage, la fonction $P$ se borne à extraire un degré de liberté de l'incrément de déplacement. En particulier, il s'agit donc d'une fonction linéaire :

$$P\left(\Delta \mathbf{u}_i^n\right) = \langle \mathbf{S} \rangle \,.\, \left\{\Delta \mathbf{u}_i^n\right\} = \Delta\tau_i \tag{16}$$

Où le vecteur nodal $\langle \mathbf{S} \rangle$ est un vecteur de sélection qui est nul partout excepté pour le degré de liberté à extraire où il vaut un:

$$\langle \mathbf{S} \rangle = \langle 0 \ldots 0 \ldots \underset{\text{noeud n, ddl i}}{1} \ldots 0 \ldots 0 \rangle \tag{17}$$

L'équation (14) se réduit alors également à une équation linéaire :

$$\eta_i = \frac{\Delta\tau_i - \langle \mathbf{S} \rangle \,.\, \left\{\Delta \mathbf{u}_i^{n-1}\right\} - \langle \mathbf{S} \rangle \,.\, \left\{\delta \mathbf{u}_{\text{impo},i}^n\right\}}{\langle \mathbf{S} \rangle \,.\, \left\{\delta \mathbf{u}_{\text{pilo},i}^n\right\}} \tag{18}$$

On notera qu'il n'y a pas de solution lorsque la correction de déplacement piloté $\delta \mathbf{u}_{\text{pilo},i}^n$ ne permet pas d'ajuster le degré de liberté requis, ce qui peut arriver si, par erreur, on bloque le degré de liberté en question. Dans ce cas, le code s'arrêtera en `ECHEC DU PILOTAGE`.

## 4.2 Pilotage en modélisation X-FEM par contrôle du saut de déplacement selon une direction : SAUT_IMPO

Dans le cas d'une modélisation FEM classique, on contrôle l'incrément de déplacement d'un seul nœud projeté sur une seule direction, les deux étant spécifiés par l'utilisateur. Avec la méthode des éléments finis étendus (XFEM), on peut contrôler l'incrément de saut de déplacement à travers l'interface. Ce saut est contrôlé en moyenne, pour un certain nombre de points d'intersection de l'interface avec les arêtes du maillage et projeté sur une seule direction $\boldsymbol{n}_a$. Il s'écrit à l'aide des degrés de liberté enrichi $\boldsymbol{b}$ détaillés dans [R7.02.12]. Pour une arête intersectée $a$, avec les notations définies sur la figure (4.2-a), la fonction de pilotage est:

$$P\left(\Delta \mathbf{u}_i^n\right) = P\left(\left[\!\left[\Delta \mathbf{u}\right]\!\right]_i^n\right) = \frac{1}{N} \sum_{\text{aretes } a}^{N} \boldsymbol{n}_a \cdot \left(\alpha_a \Delta \boldsymbol{b}_i^n(N_a^-) + (1-\alpha_a) \Delta \boldsymbol{b}_i^n(N_a^+)\right) = \Delta\tau_i \tag{19}$$

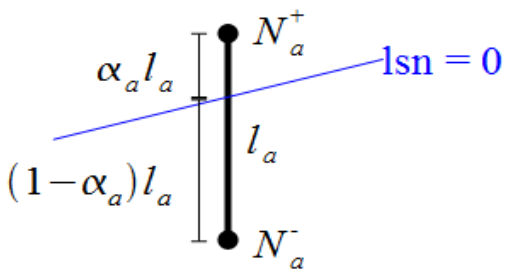



**Figure 4.2-a: Arête intersectée par l'interface**


Document diffusé sous licence GNU FDL (htt

Pour reprendre les notations précédentes, le vecteur nodal $\langle \mathbf{S} \rangle$ s'écrit donc :

$$\langle \mathbf{S} \rangle = \frac{1}{N} \left\langle 0 \ldots \underset{\substack{noeud\ N_1^+ \\ ddl\ h_x}}{\alpha_1 n_x} \quad \underset{ddl\ h_y}{\alpha_1 n_y} \ldots (1-\alpha_1) \underset{noeud\ N_1^-}{n_x} \quad (1-\alpha_1) n_y \quad \ldots \quad \underset{noeud\ N_N^+}{\alpha_N n_x} \quad \alpha_N n_y \ldots 0 \right\rangle \tag{20}$$

Les arêtes $a$ sont choisies comme appartenant à un ensemble d'arêtes indépendantes, c'est à dire qu'elles n'ont aucun nœud en commun, et qu'il est impossible d'en rajouter une telle que cette propriété soit conservée (figure 4.2-b). Un tel set d'arêtes contrôlées satisfait la condition LBB pour le traitement du contact (voir [R7.02.12]) et donc pour toute imposition de condition de contrôle sur l'interface. Néanmoins, la condition de type SAUT_IMPO étant imposée en moyenne, un pilotage sur l'ensemble des degrés de liberté devrait être possible, puisque celui-ci a lieu in fine degré de liberté par degré de liberté. Le choix d'un ensemble d'arêtes indépendantes est tout de même justifié car il facilite grandement le stockage informatique des coefficients puisque on a une seule information à stocker par degré de liberté dans le vecteur $\langle \mathbf{S} \rangle$ (voir formule 20). Par ailleurs, la réutilisation d'algorithmes de sélection expliqués dans [R7.02.12] permet de construire facilement un tel ensemble.

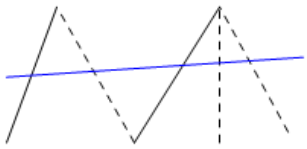


**Figure 4.2-b: Sélection d'arêtes indépendantes**

## 4.3 Pilotage par longueur d'arc : LONG_ARC

Une autre forme de pilotage très largement utilisée consiste à contrôler la norme de l'incrément de déplacement (par rapport à certains nœuds et certaines composantes) : on parle alors de pilotage par longueur d'arc cylindrique, voir Bonnet and Wood [bib1]. Plus précisément, la fonction $P$ s'exprime par :

$$P\left(\Delta \mathbf{u}_i^n\right) = \left\| \Delta \mathbf{u}_i^n \right\|_{<S>} = \sqrt{\left(\langle \mathbf{S} \rangle . \left\{ \Delta \mathbf{u}_i^n \right\}\right) . \left(\langle \mathbf{S} \rangle . \left\{ \Delta \mathbf{u}_i^n \right\}\right)} = \Delta \tau_i \tag{21}$$

Où, à nouveau, le vecteur nodal $\langle \mathbf{S} \rangle$ permet de sélectionner les degrés de liberté employés pour le calcul de la norme (il vaut un pour les degrés de liberté sélectionnés et zéro ailleurs). Dans ce cas là on remarquera que $\langle \mathbf{S} \rangle . \left\{ \Delta \mathbf{u}_i^n \right\}$ n'est pas un scalaire, mais un vecteur correspondant au produit des composantes de $\langle \mathbf{S} \rangle$ par celles de $\left\{ \Delta \mathbf{u}_i^n \right\}$. Dans ce cas, l'équation de pilotage se réduit à une équation du second degré :

$$\begin{aligned} &\eta_i^2 . \left[ \left( \langle \mathbf{S} \rangle . \left\{ \delta \mathbf{u}_{\text{pilo},i}^n \right\} \right) . \left( \langle \mathbf{S} \rangle . \left\{ \delta \mathbf{u}_{\text{pilo},i}^n \right\} \right) \right] + \\ &2 . \eta_i . \left[ \left( \langle \mathbf{S} \rangle . \left\{ \Delta \mathbf{u}_i^{n-1} + \delta \mathbf{u}_{\text{impo},i}^n \right\} \right) . \left( \langle \mathbf{S} \rangle . \left\{ \delta \mathbf{u}_{\text{pilo},i}^n \right\} \right) \right] + \\ &\left[ \left( \langle \mathbf{S} \rangle . \left\{ \Delta \mathbf{u}_i^{n-1} + \delta \mathbf{u}_{\text{impo},i}^n \right\} \right) . \left( \langle \mathbf{S} \rangle . \left\{ \Delta \mathbf{u}_i^{n-1} + \delta \mathbf{u}_{\text{impo},i}^n \right\} \right) - \Delta \tau_i^2 \right] = 0 \end{aligned} \tag{22}$$

Cette équation peut ne pas admettre de solution. Dans ce cas, on choisit la valeur $\eta_i$ qui minimise le polynôme (22). On vérifie alors bien $\tilde{P}\left(\eta_i\right) > \Delta \tau_i$. Dans le cas contraire, elle admet deux racines (ou une racine double). On choisit celle des deux qui minimise :

- soit la norme de l'incrément de déplacement $\left\| \Delta \mathbf{u}_i^n \right\|$ ;
- soit le résidu d'équilibre $\mathbf{R}\left(\mathbf{u}_i, t_i\right) = \mathbf{L}_i^{\text{int}} - \mathbf{L}_i^{\text{méca}}$ ;



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

- soit l'angle formé par $\Delta \mathbf{u}_{i-1}$ et $\Delta \mathbf{u}_i^n$ (où $\Delta \mathbf{u}_{i-1}$ est l'incrément de déplacement solution du pas de temps précédent), c'est-à-dire celle qui maximise le cosinus de cet angle dont l'expression est :

$$\cos\left(\Delta \mathbf{u}_{i-1}, \Delta \mathbf{u}_i^n\right) = \frac{\left\langle \Delta \mathbf{u}_i^{n-1} + \delta \mathbf{u}_{\text{impo},i}^n + \eta_i . \delta \mathbf{u}_{\text{pilo},i}^n \right\rangle . \left\{\Delta \mathbf{u}_{i-1}\right\}}{\left\|\Delta \mathbf{u}_i^{n-1} + \delta \mathbf{u}_{\text{impo},i}^n + \eta_i . \delta \mathbf{u}_{\text{pilo},i}^n\right\| . \left\|\Delta \mathbf{u}_{i-1}\right\|} \tag{23}$$

Pour calculer l'angle, on n'utilise que les inconnues correspondant aux déplacements, à l'exclusion de toutes les autres (en particulier les rotations dans les éléments de structures).

On remarque que ce dernier critère angulaire sélectionne la mauvaise solution lorsqu'on passe d'une phase de charge à une phase de décharge ou vice-versa, c'est à dire lorsque le coefficient de pilotage $C_i$ change de signe. La valeur $C_{i-1}$ est donc mise en mémoire et actualisée à chaque pas de temps convergé, son initialisation pour chaque appel à STAT_NON_LINE étant transparente pour l'utilisateur : $C_{i-1}$ est ainsi initialisé lors de la lecture de l'état initial, $C_i$ ayant été inclus dans les paramètres de calcul à archiver dans le concept résultat. Si $C_i C_{i-1} > 0$ , on maximise le critère (23) . Dans le cas contraire, on le minimise afin de sélectionner l'incrément de déplacement en sens opposé de l'incrément précédent.

## 4.4 Pilotage en modélisation X-FEM par contrôle de la norme du saut de déplacement : SAUT_LONG_ARC

Dans le cas d'une modélisation XFEM, le pilotage par longueur d'arc consiste à contrôler la norme de l'incrément du saut de déplacement pour un certain nombre de points d'intersection du maillage avec l'interface, qui sont choisis sur un ensemble d'arêtes indépendantes de la même façon qu'en 4.2. La fonction de pilotage s'écrit alors :

$$P\left(\Delta \mathbf{u}_i^n\right) = P\left(\left[\Delta \mathbf{u}\right]_i^n\right) = \sqrt{\frac{1}{N} \sum_{aretes\, a}^{N} \left[\alpha_a \Delta \boldsymbol{b}_i^n(N_a^-) + (1-\alpha_a) \Delta \boldsymbol{b}_i^n(N_a^+)\right]^2} = \Delta \tau_i \tag{24}$$

Si nous développons cette dernière équation, nous avons :

$$P\left(\left[\Delta \mathbf{u}\right]_i^n\right)^2 =$$
$$\underbrace{\frac{1}{N} \sum_{aretes\, a}^{N} \left(\alpha_a \Delta \boldsymbol{b}_i^n(N_a^-)\right)^2 + \left((1-\alpha_a) \Delta \boldsymbol{b}_i^n(N_a^+)\right)^2}_{N\left\|\langle \mathbf{S}\rangle . \{\Delta u_i^n\}\right\|^2} + \underbrace{\frac{1}{N} \sum_{aretes\, a}^{N} 2\alpha_a (1-\alpha_a) \Delta \boldsymbol{b}_i^n(N_a^+) \Delta \boldsymbol{b}_i^n(N_a^-)}_{N\left([M]\langle \mathbf{S}\rangle . \{\Delta u_i^n\}\right) \cdot \left(\langle \mathbf{S}\rangle . \{\Delta u_i^n\}\right)} \tag{25}$$

où $\langle \mathbf{S} \rangle$ est le vecteur nodal défini par la formule 20 (on remarquera que $\langle \mathbf{S} \rangle . \left\{\Delta \mathbf{u}_i^n\right\}$ n'est pas un scalaire, mais un vecteur correspondant au produit des composantes de $\langle \mathbf{S} \rangle$ par celles de $\left\{\Delta \mathbf{u}_i^n\right\}$ ), et $[M]$ la matrice qui aux degrés de liberté d'un nœud fait correspondre les degrés de liberté du nœud situé sur la même arête indépendant, à l'autre extrémité, c'est-à-dire :

$$\forall \text{ arête } a, \forall \text{ direction } i, [M]\left\langle \mathbf{S^i}(N_a^+)\right\rangle = \left\langle \mathbf{S^i}(N_a^-)\right\rangle \quad et \quad [M]\left\langle \mathbf{S^i}(N_a^-)\right\rangle = \left\langle \mathbf{S^i}(N_a^+)\right\rangle$$
$$\text{où} \left\langle \mathbf{S^i}(N)\right\rangle = \langle 0 \ldots \underset{\substack{\text{noeud } N \\ \text{direction } i}}{1} \ldots 0 \rangle \tag{26}$$

Finalement, la notation en termes de vecteurs est :



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

$$P\left(\Delta \mathbf{u}_i^n\right)=\sqrt{N\left\|\langle \mathbf{S}\rangle.\{\Delta u_i^n\}\right\|^2+N\left([M]\langle \mathbf{S}\rangle.\{\Delta u_i^n\}\right)\cdot\left(\langle \mathbf{S}\rangle.\{\Delta u_i^n\}\right)}=\Delta\tau_i \quad (27)$$

Lors de l'initialisation du pilotage, il faut donc initialiser en plus du vecteur nodal $\langle \mathbf{S}\rangle$ une structure $[M]$ qui repère quels points appartiennent à la même arête. En décomposant $\left\{\Delta \mathbf{u}_i^n\right\}$, on trouve une équation analogue à (22) qui tient compte du terme supplémentaire.

Pour la modélisation XFEM, les méthodes de sélection de la solution restent strictement identiques, mis à part le remplacement dans les expressions des critères de norme et d'angle du déplacement par le saut de déplacement à travers l'interface. Ainsi, le critère angulaire revient par exemple à maximiser $\cos\left(\left[\!\left[\Delta \mathbf{u}\right]\!\right]_{i-1},\left[\!\left[\Delta \mathbf{u}\right]\!\right]_i^n\right)$.

## 4.5 Pilotage par l'incrément de déformation : DEFORMATION

Le pilotage par incrément de déformation consiste à exiger que l'incrément de déformation du pas courant reste proche en direction de la déformation au début du pas de temps, et ce pour au moins un point de Gauss de la structure. On requiert ainsi qualitativement qu' **a minima** un point de la structure conserve le mode de déformation qu'il avait au préalable (par exemple, traction dans une direction donnée).

### 4.5.1 Cas des petites déformations

Pour les petites déformations, on peut rendre compte de cette exigence moyennant le choix de la fonction de pilotage suivante :

$$P\left(\Delta \mathbf{u}_i^n\right)=\underset{g}{\mathrm{Max}}\frac{\boldsymbol{\varepsilon}_{g,i-1}}{\left\|\boldsymbol{\varepsilon}_{g,i-1}\right\|}.\Delta\boldsymbol{\varepsilon}_{g,i}=\Delta\tau_i \quad (28)$$

Où l'indice $g$ balaie les points de Gauss de la structure (ou seulement les mailles précisées par le mot-clef GROUP_MA dans PILOTAGE) et où la déformation en un point de Gauss se déduit du vecteur nodal des déplacements via l'emploi des matrices « partie symétrique du gradient des fonctions de forme » $\mathbf{B}_g$ (à ne pas confondre avec la matrice des conditions de Dirichlet):

$$\boldsymbol{\varepsilon}_{g,i-1}=\mathbf{B}_g\cdot\mathbf{u}_{i-1} \text{ et } \Delta\boldsymbol{\varepsilon}_{g,i}=\mathbf{B}_g\cdot\Delta\mathbf{u}_i^n \quad (29)$$

Le contrôle du pilotage en fonction de $\eta_i$ s'écrit alors :

$$P\left(\Delta \mathbf{u}_i^n\right)=\underset{g}{\mathrm{Max}}\left(A_g^{(0)}+\eta_i\cdot A_g^{(1)}\right)=\underset{g}{\mathrm{Max}}\left(L_g\left(\eta_i\right)\right)=\Delta\tau_i \quad (30)$$

Avec les deux termes :

$$A_g^{(0)}=\frac{\boldsymbol{\varepsilon}_{g,i-1}}{\left\|\boldsymbol{\varepsilon}_{g,i-1}\right\|}\cdot\mathbf{B}_g\left(\Delta\mathbf{u}_i^{n-1}+\delta\mathbf{u}_{\mathrm{impo},i}^n\right) \quad (31)$$

Et :

$$A_g^{(1)}=\frac{\boldsymbol{\varepsilon}_{g,i-1}}{\left\|\boldsymbol{\varepsilon}_{g,i-1}\right\|}\cdot\mathbf{B}_g\left(\delta\mathbf{u}_{\mathrm{pilo},i}^n\right) \quad (32)$$

Ce pilotage ne dépend pas de la loi de comportement, pourvue qu'elle utilise le tenseur des petites déformations (option DEFORMATION='PETIT' ou DEFORMATION='PETIT_REAC').

### 4.5.2 Cas des grandes déformations

En présence de grandes déformations, on peut généraliser la fonction de pilotage (28) en employant des déformations de Green-Lagrange (mesure lagrangienne des déformations dans la configuration initiale) :

$$P\left(\Delta \mathbf{u}_i^n\right)=\underset{g}{\mathrm{Max}}\frac{\mathbf{E}_{g,i-1}}{\left\|\mathbf{E}_{g,i-1}\right\|}.\Delta\mathbf{E}_{g,i} \quad (33)$$

La mesure des déformations $\mathbf{E}$ est écrite en fonction du tenseur gradient de transformation $\mathbf{F}$ :



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

$$\mathbf{E}=\frac{1}{2}\cdot\left(\mathbf{F}^T.\mathbf{F}-\mathbf{I}\right) \text{ avec } \mathbf{F}=\mathbf{I}+\nabla\mathbf{u} \quad (34)$$

Toutefois, on n'aboutirait plus comme précédemment à une fonction affine par morceaux. Pour y remédier, on décide de procéder à une linéarisation de $\Delta\mathbf{E}_{g,i}$ par rapport à $\Delta\mathbf{u}_i^{n-1}$ . $P$ a alors une expression similaire à ( 30 ) avec :

$$A_g^{(0)}=\frac{\mathbf{E}_{g,i-1}}{\|\mathbf{E}_{g,i-1}\|}\cdot\mathrm{sym}\left[\mathbf{F}_{g,i}^T.\nabla_g\left(\Delta\mathbf{u}_i^{n-1}+\delta\mathbf{u}_{\mathrm{impo},i}^n\right)\right] \quad (35)$$

Et:

$$A_g^{(1)}=\frac{\mathbf{E}_{g,i-1}}{\|\mathbf{E}_{g,i-1}\|}\cdot\mathrm{sym}\left[\mathbf{F}_{g,i}^T.\delta\mathbf{u}_{\mathrm{pilo},i}^n\right] \quad (36)$$

Où $\nabla_g(\mathbf{u})$ désigne le gradient (non symétrisé) des déplacements $\mathbf{u}$ évalué au point de Gauss d'indice $g$ . Tout comme dans le cas précédent, ce pilotage ne dépend pas de la loi de comportement, pourvue qu'elle utilise le tenseur des grandes déformations.

### 4.5.3 Résolution de l'équation non-linéaire de pilotage

La fonction $P\left(\Delta\mathbf{u}_i^n\right)$ est convexe et linéaire par morceaux. Elle admet généralement aucune, une ou deux solutions, cf. figure (4.6.4-a). Lorsqu'elle n'admet pas de solutions, on arrête l'algorithme de STAT_NON_LINE (échec du pilotage) : si l'utilisateur utilise la subdivision automatique du pas de temps, ce dernier sera alors activé.
Dans le cas où elle admet deux solutions, trois possibilités sont offertes : on choisit celle des deux qui minimise :

- soit la norme de l'incrément de déplacement $\left\|\Delta\mathbf{u}_i^n\right\|$ ;
- soit le résidu d'équilibre $\mathbf{R}(\mathbf{u}_i,t_i)=\mathbf{L}_i^{\mathrm{int}}-\mathbf{L}_i^{\mathrm{méca}}$ ;
- soit l'angle formé par $\Delta\mathbf{u}_{i-1}$ et $\Delta\mathbf{u}_i^n$ ;

Pour résoudre l'équation (30), on propose l'algorithme présenté ci-dessous. Il est basé sur la construction d'intervalles emboîtés : les bornes du dernier d'entre eux sont les solutions de l'équation et, comme annoncé précédemment, on choisit celle qui conduit au $\tilde{\mathbf{u}}(\eta)$ le plus proche de $\mathbf{u}_{i-1}$ . Cet algorithme, rapide, s'appuie sur la résolution de $G$ équations scalaires linéaires, où $G$ désigne le nombre total de points de Gauss. L'algorithme peut prendre fin prématurément lorsque l'un des intervalles est vide, ce qui signifie que l'équation (30) n'admet pas de solutions.

| | | |
|---|---|---|
| (1) | Initialisation de l'intervalle | $I_0=]-\infty,+\infty[$ |
| (2) | Boucle sur les points de Gauss $g$ | |
| (2.0) | La pente est nulle | Si $A_g^{(1)}=0$ |
| (2.0.1) | Si $A_g^{(0)}>\Delta\tau$ échec<br>Sinon on continue | |
| (2.1) | Racine de la fonction linéaire active | $\eta_g$ tel que $L_g(\eta_g)=\Delta\tau$ |
| (2.2) | Construction de l'intervalle suivant | |
| (2.2.1) | Si la fonction linéaire active est croissante | Si $A_g^{(1)}>0\Rightarrow I_g=I_{g-1}\cap]-\infty,\eta_g]$ |
| (2.2.2) | Si la fonction linéaire active est décroissante | Si $A_g^{(1)}<0\Rightarrow I_g=I_{g-1}\cap[\eta_g,+\infty[$ |
| (2.3) | Arrêter si l'intervalle est vide | |
| (3) | Les solutions sont les bornes de l'intervalle[3] | $\eta\in\mathrm{Fr}(I_G)\Rightarrow\max_g L_g(\eta)=\Delta\tau$ |

3 On note $\eta\in\mathrm{Fr}(I_G)$ le fait que $\eta$ appartient aux bornes de l'intervalle $I_G$



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

**Algorithme de résolution de l'équation affine par morceaux**

## 4.6 Pilotage par la prédiction élastique : PRED_ELAS

### 4.6.1 Algorithme

Si le pilotage par l'incrément de déformation s'avère suffisant pour suivre des solutions dissipatives dans la plupart des instabilités matériaux, l'existence de solutions n'est néanmoins pas prouvée. On lui préfère alors une méthode de pilotage fondée sur la prédiction élastique pour laquelle l'existence de solutions est démontrée mais qui, en revanche, est spécifique à chaque loi de comportement (implantée uniquement pour les lois ENDO_SCALAIRE, ENDO_FRAGILE, ENDO_ISOT_BETON, ENDO_ORTH_BETON, BETON_DOUBLE_DP, VMIS_ISOT_* et celles relatives aux éléments de joints [R5.06.09]). Plus précisément, lorsque la loi de comportement est gouvernée par un seuil, on définit $P$ comme le maximum sur tous les points de Gauss de la valeur de la fonction seuil dans le cas d'un essai élastique (réponse incrémentale élastique du matériau).
Ainsi, considérons que l'état du matériau est décrit par la déformation $\boldsymbol{\varepsilon}$ et un ensemble de variables internes $a$. Appelons respectivement $\boldsymbol{\sigma}\left(\boldsymbol{\varepsilon},a\right)$ et $A\left(\boldsymbol{\varepsilon},a\right)$ les contraintes et les forces thermodynamiques associées à $a$. Supposons en outre que les lois d'évolution de $a$ soient gouvernées par un seuil $f\left(A,\boldsymbol{\varepsilon},a\right)$ :

$$f\left(A,\boldsymbol{\varepsilon},a\right)\leq 0 \text{ et } \lambda\cdot f\left(A,\boldsymbol{\varepsilon},a\right)=0 \tag{37}$$

Et une fonction d'écoulement $G\left(A,\boldsymbol{\varepsilon},a\right)$ :

$$\dot{a}=\lambda\, G\cdot\left(A,\boldsymbol{\varepsilon},a\right) \tag{38}$$

Une telle formulation englobe la plupart des modèles de comportement dissipatifs et indépendants de la vitesse de chargement. La fonction seuil vaut alors pour un essai élastique :

$$f^{\mathrm{el}}\left(\boldsymbol{\varepsilon}\right)=f\left(A\left(\boldsymbol{\varepsilon},a_{i-1}\right),\boldsymbol{\varepsilon},a_{i-1}\right) \tag{39}$$

On simplifie le problème en linéarisant $f^{\mathrm{el}}$ par rapport à $\boldsymbol{\varepsilon}$ au voisinage du point $\boldsymbol{\varepsilon}^{*}$ tel que $f^{\mathrm{el}}=\Delta\tau$ :

$$f_{L}^{\mathrm{el}}\left(\boldsymbol{\varepsilon}\right)\underset{\mathrm{def}}{=}f^{\mathrm{el}}\left(\boldsymbol{\varepsilon}^{*}\right)+\left(\frac{\partial f}{\partial A}\cdot\frac{\partial A}{\partial\boldsymbol{\varepsilon}}+\frac{\partial f}{\partial\boldsymbol{\varepsilon}}\right)\Bigg|_{\boldsymbol{\varepsilon}^{*}}\cdot\left(\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}^{*}\right) \tag{40}$$

Le choix de $\boldsymbol{\varepsilon}^{*}$ permet de résoudre le problème exact, comme montré dans [bib3]. Il nécessite donc de résoudre un problème non-linéaire local pour chaque point de Gauss, puis l'algorithme global se fonde sur des droites, ce qui lui assure une grande efficacité. Il existe zéro, une ou deux solutions au problème local $f^{\mathrm{el}}=\Delta\tau$. Dans le cas d'absence de solution au problème local, le problème global ne peut avoir de solution : on arrête. Sinon (cas à une ou deux solutions), on linéarise autour de chacune des solutions.
Finalement, la fonction de contrôle du pilotage est définie comme le maximum de $f_{L}^{\mathrm{el}}$ par rapport à tous les points de Gauss $g$, fonction qui ne dépend que de $\Delta\mathbf{U}$ :

$$P\left(\Delta\mathbf{u}_{i}^{n}\right)=\underset{g}{\mathrm{Max}}\, f_{L}^{\mathrm{el}}\left(\boldsymbol{\varepsilon}_{g,i-1}+\mathbf{B}_{g}\cdot\Delta\mathbf{u}_{i}^{n}\right)=\Delta\tau_{i} \tag{41}$$

Finalement, l'équation de contrôle du pilotage s'écrit :

$$P\left(\Delta\mathbf{u}_{i}^{n}\right)=\underset{g}{\mathrm{Max}}\left(A_{g}^{(0)}+\eta_{i}\cdot A_{g}^{(1)}\right)=\underset{g}{\mathrm{Max}}\left(L_{g}\left(\eta_{i}\right)\right)=\Delta\tau_{i} \tag{42}$$

Avec :

$$A^{(0)}=f^{el}\left(\boldsymbol{\varepsilon}^{*}\right)+\left(\frac{\partial f}{\partial A}\cdot\frac{\partial A}{\partial\boldsymbol{\varepsilon}}+\frac{\partial f}{\partial\boldsymbol{\varepsilon}}\right)\Bigg|_{\boldsymbol{\varepsilon}^{*}}\cdot\left(\mathbf{B}_{g}\left(\mathbf{u}_{i-1}+\Delta\mathbf{u}_{i}^{n-1}+\delta\mathbf{u}_{\mathrm{impo},i}^{n}\right)-\boldsymbol{\varepsilon}^{*}\right) \tag{43}$$

Et :

$$A^{(1)}=\left(\frac{\partial f}{\partial A}\cdot\frac{\partial A}{\partial\boldsymbol{\varepsilon}}+\frac{\partial f}{\partial\boldsymbol{\varepsilon}}\right)\Bigg|_{\boldsymbol{\varepsilon}^{*}}\cdot\left(\mathbf{B}_{g}\cdot\delta\mathbf{u}_{\mathrm{pilo},i}^{n}\right) \tag{44}$$

### 4.6.2 Expressions en fonction des lois de comportement



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

On est ainsi ramené à un problème identique à celui du pilotage par l'incrément de la déformation. On emploie bien entendu les mêmes algorithmes de résolution que ceux présentés dans le paragraphe précédent (voir §11).

Les caractéristiques de l'équation de pilotage dépendent de la loi de comportement car l'expression de la fonction seuil élastique $f^{\mathrm{el}}$ est elle-même une donnée du problème non-linéaire de comportement :

- Pour les éléments à discontinuité interne PLAN_ELDI et AXIS_ELDI, la loi de comportement est de type CZM_EXP et l'expression de la loi de pilotage se trouvera dans la documentation [R7.02.14]. La fonction seuil élastique s'écrit $f_{el}(\sigma_n,\sigma_t)=\sqrt{<\sigma_n>_+^2+\sigma_t^2}-R(\kappa)$ ;
- Pour les éléments de joint *_JOINT, les lois de comportement sont des lois CZM régularisées CZM_EXP_REG et CZM_LIN_REG (voir [R7.02.11]), et pour les éléments de type interface *_INTERFACE, les lois de comportement sont des lois CZM non-régularisées comme CZM_OUV_MIX et CZM_TAC_MIX. L'expression de la loi de pilotage se trouvera dans la documentation [R7.02.11]. La fonction seuil élastique s'écrit $f_{el}\simeq \underset{pt\ gauss}{max}\left\{\frac{\|\delta^0+\eta\,\delta^1\|-\kappa^-}{G_c/\sigma_c+\kappa^-}\right\}=\frac{\Delta\tau}{C}$ ;
- Pour la loi ENDO_ISOT_BETON ([R7.01.04]), il y a une particularité. Au lieu de chercher un paramètre de pilotage $\eta$ qui fasse sortir le critère d'une valeur $\Delta\tau$ avec l'endommagement issu du pas de temps précédent, on cherche un paramètre $\eta$ qui nous ramène sur le critère avec un endommagement augmenté de $\Delta\tau$. La fonction seuil élastique s'écrit donc $\tilde{f}^{el}(\eta,d^-)=\Delta\tau \Rightarrow \tilde{f}^{el}(\eta,d^-+\Delta\tau)=0$ ;
- Pour la loi ENDO_ORTH_BETON, voir [R7.01.09] ;
- Pour les autres lois d'endommagement de type élastique comme ENDO_SCALAIRE ou ENDO_FRAGILE (avec formulation locale ou à gradients), tout comme pour la loi ENDO_ISOT_BETON, la fonction seuil élastique est choisie de manière à chercher un paramètre de pilotage contrôle par la valeur d'endommagement et non pas par la sortie du critère. La fonction seuil élastique s'écrit :

$$\tilde{f}^{\mathrm{el}}(\eta)=\frac{1}{2}\left(\boldsymbol{e_0}+\eta\ \boldsymbol{e_1}\right)\cdot\boldsymbol{E}\cdot\left(\boldsymbol{e_0}+\eta\ \boldsymbol{e_1}\right)-s$$

$$\begin{cases} \text{loi locale} & \boldsymbol{e_0}=\boldsymbol{\epsilon_0} \quad e_1=\boldsymbol{\epsilon_1} \quad s=k\left(d^-\right) \\ \text{gradient d'endommagement} & \boldsymbol{e_0}=\boldsymbol{\epsilon_0} \quad \boldsymbol{e_1}=\boldsymbol{\epsilon_1} \quad s=k\left(d^-\right)-c\,\Delta\,d^- \\ \text{déformation régularisée} & \boldsymbol{e_0}=\bar{\boldsymbol{\epsilon}}_0 \quad \boldsymbol{e_1}=\bar{\boldsymbol{\epsilon}}_1 \quad s=k\left(d^-\right) \end{cases} \tag{45}$$

### 4.6.3 Illustration de l'algorithme de maximisation

Afin d'illustrer notre propos et notamment de comprendre pourquoi un tel choix de $\boldsymbol{\varepsilon}^*$ permet de résoudre le problème exact, on prend l'exemple d'une loi de comportement cohésive CZM_EXP_REG ou CZM_LIN_REG (cf [R7.02.11]) implémentée sur un élément de joint (cf [R3.06.09]). Le saut de déplacement $\boldsymbol{\delta}$ est la variable primale (notée génériquement $\boldsymbol{\epsilon}$ dans ce qui précède), et $\kappa$ la variable interne (notée génériquement $A$). La fonction seuil s'écrit alors :

$$f^{\grave{e}l}(\boldsymbol{\delta})=\frac{\|\boldsymbol{\delta}\|_{\grave{e}q}-\kappa_{i-1}}{\kappa_{REF}} \quad \text{avec} \quad \kappa_{i-1}=\underset{v\in[0,t_{i-1}]}{max}\|\boldsymbol{\delta}(v)\|_{\grave{e}q} \ ; \ \|\boldsymbol{\delta}\|_{\grave{e}q}=\sqrt{\langle\delta_n\rangle_+^2+\delta_\tau^2} \tag{46}$$

Sans détailler la démonstration, on peut montrer que $\|\boldsymbol{\delta}\|_{\grave{e}q}$ vérifie une inégalité triangulaire. Ceci nous permet de prouver la convexité de la fonction :



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

$$f^{èl}(\eta)=f^{èl}\left(\delta(\eta)\right)=\frac{\left\|\delta_{i-1}+\Delta\delta_i^{\,n}+d\,\delta^{\,cte}+\eta\,d\,\delta^{\,pilo}\right\|_{èq}-\kappa_{i-1}}{\kappa_{REF}} \qquad (47)$$

Nous avons tracé en figure 4.6.3-a une représentation graphique **qualitative** des fonctions $f^{èl}$ pour les différents points de Gauss des éléments de joint, ainsi que les solutions du problème exact de maximisation global sur l'ensemble des points de Gauss :

$$\max_g f_g^{èl}(\eta) \overset{déf}{=} \max_g f^{èl}\left(\delta_g(\eta)\right)=\Delta\tau_i$$

Il apparaît alors que les solutions $\eta$ de ce problème exact sont identiques aux solutions obtenues avec l'algorithme de résolution de l'équation affine par morceaux, c'est-à-dire les solutions de l'équation (42) $\max_g A_1^{(g)}\eta+A_0^{(g)}=\Delta\tau_i$, avec dans notre exemple :

$A_1^{(g)}=\dfrac{\partial f_g^{èl}}{\partial\eta}(\eta_g^*)$ ; $A_0^{(g)}=\Delta\tau-A_1^{(g)}\eta_g^*$ où $\eta_g^*$ est solution de l'équation locale $f_g^{èl}(\eta)=\Delta\tau_i$

Nous renvoyons une fois de plus le lecteur à [3] pour une démonstration rigoureuse.

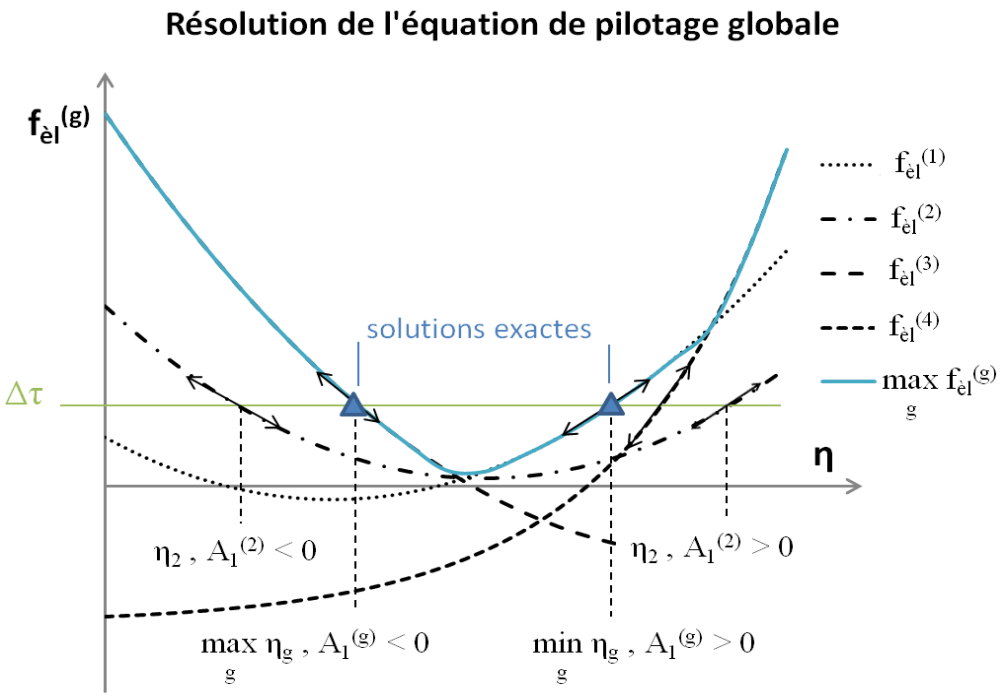



**Figure 4.6.3-a: Résolution de l'équation de pilotage, représentation graphique qualitative**

## 4.6.4 Choix de solution

Dans le cas où le problème global possède deux solutions, les algorithmes de choix décrits en 4.3 sont réutilisés. En général, la sélection par résidu est retenue en raison de sa robustesse. Cependant, le fait de choisir la solution qui minimise le résidu ne nous garantit pas la convergence de la méthode de Newton. Certains cas ont pu être exhibés pour lesquels la solution de résidu initial maximal converge tandis que ce n'est pas le cas de la solution de résidu initial minimal.

En règle générale pour les calculs piloté le pas de temps ne remplis plus le rôle de chargement, mais juste un paramètre de suivi de la solution. Bien que les solutions mécaniques soient identique, on peut ainsi obtenir une différence significative au même pas de pilotage pour les problèmes avec les instabilités importantes, en



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

changeant par exemple la machine d'exécution. Ce qui est illustré sur la figure 4.6.4-a, ou lors du snap-back dans la réponse globale force-déplacement on observe le décalage d'instant de calcul entre les deux courbes.

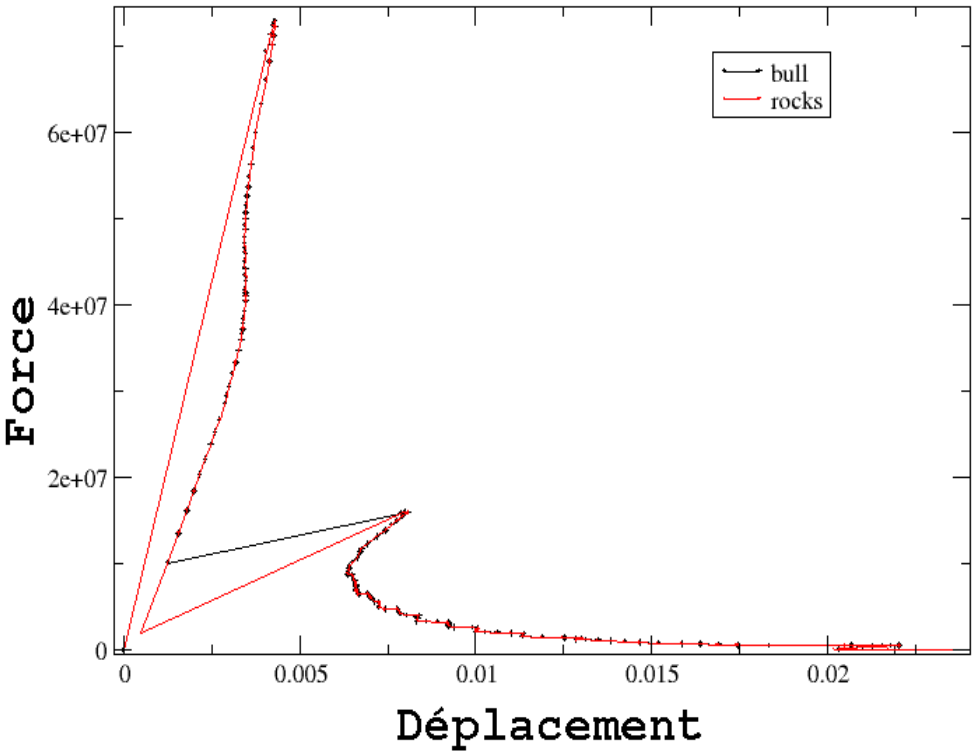



**Figure 4.6.4-a: Illustration de décalage de l'instant de pilotage sur une courbe force-déplacement pour un problème fortement instable en fonction de la précision machine**



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

# 5 Bibliographie

1) Bonnet J. and Wood R. D. [1997]. Nonlinear continuum mechanics for finite element analysis. Cambridge university press.

2) Bonnans J. F., Gilbert J.-C., Lemaréchal C. et Sagastizabal C. [1997]. Optimisation numérique : aspects théoriques et pratiques. Mathématiques et Applications, 27, ed. Springer.

3) Lorentz E. and Badel P. [2001]. A new path-following constraint for strain-softening finite element simulations. International journal for numerical methods in engineering. 2004, vol. 60, no 2, pp. 499 – 526.

4) Shi J. and Crisfield M. A. [1995]. Combining arc-length control and line searches in path following. Comm. Num. Meth. Eng. 11, pp. 793-803.

# 6 Description des versions du document

| Version Aster | Auteur(s) Organisme(s) | Description des modifications |
|---|---|---|
| 7 | E.Lorentz EDF-R&D/AMA | Texte initial |
| 8.4 | V.Godard, P.Badel, E.Lorentz EDF-R&D/AMA | |
| 10.2 | M.Abbas EDF-R&D/AMA | Mise en conformité des notations avec [R5.03.01], ajout d'une introduction explicative de l'utilité du pilotage |
| 10.5 | P.Massin, G. Ferte EDF-R&D/AMA | Introduction de pilotage pour XFEM |
| 10.5 | K.Kazymyrenko EDF-R&D/AMA | Correction de la fiche d'anomalie doc 17120 |



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