

Algorithmes de propagation de fissures avec X-FEM

Résumé :

Ce document présente les algorithmes de propagation de fissures dans le cadre de la méthode X-FEM [R7.02.12]. La fissure est alors représentée par des level sets. La propagation de la fissure consiste donc à mettre à jour les level sets compte-tenu de l'avancée du front de fissure.

Table des Matières

1	Introduction.....	3
2	Calcul du champ de vitesse de propagation.....	4
2.1	Vitesse d'avancée du fond de fissure.....	4
2.2	Champ de vitesse étendu au maillage.....	5
2.3	Calcul de $\mathbf{v}$ et $\mathbf{n}$ pour chaque nœud du maillage.....	6
2.4	Calcul de la base locale au point P.....	8
2.5	Ajustement du champ de vitesse normale.....	9
3	Mise à jour des level sets.....	11
3.1	Équations de mise à jour.....	11
4	Réinitialisation et réorthogonalisation des level sets	13
4.1	Nécessité de garder des fonctions de distances signées.....	13
4.2	Équations de Hamilton-Jacobi.....	14
4.3	Schémas explicites de résolution	15
4.4	Calcul des level sets à proximité des iso-zéros.....	44
5	Localisation du domaine de calcul des level sets.....	47
5.1	Sélection des nœuds/points et des éléments du domaine de calcul localisé.....	48
5.2	Déplacement du domaine localisé en suite à la propagation de la fissure.....	49
5.3	Limitations sur la valeur du rayon du domaine localisé entre deux propagation successives.....	50
6	Calcul direct des nouvelles level sets après avancée.....	53
6.1	Méthode de calcul des level sets.....	53
6.2	Traitement des points problématiques.....	54
6.3	Discontinuité de la level set normale et limitations de calcul.....	55
6.4	Localisation du domaine.....	56
7	Méthode maillage pour la mise à jour des level sets.....	58
7.1	Présentation de la méthode.....	58
7.2	Algorithme.....	58
8	Séparation et fusion des fronts de fissure.....	60
8.1	Méthode maillage.....	60
8.2	Méthode upwind et simplexe: introduction d'un fond virtuel.....	63
8.3	Méthode géométrique.....	67
9	Routines informatiques :	68
10	Bibliographie.....	71
11	Description des versions.....	72

1 Introduction

Un avantage conséquent de la méthode X-FEM étant la possibilité d'étudier plusieurs géométries de fissure sur un même maillage, on peut aisément étudier l'évolution de la fissure au cours du temps. L'opérateur `PROPA_F` [U4.82.11] calcule la fissure résultante d'une propagation en fatigue selon une loi de Paris locale.

À partir de la donnée d'une fissure à un instant donné et des coefficients d'intensité des contraintes correspondants (Opérateur `CALC_G` [U4.82.03]), l'opérateur calcule les level sets après avancée de la fissure, puis la structure de données `fiss_xfem` est enrichie de la même manière qu'avec l'opérateur `DEFI_FISS_XFEM` [U4.82.08].

Une introduction à la méthode des Level sets est donnée dans [R7.02.12]. La phase de propagation de la fissure se traduit simplement par la propagation des level sets

À l'heure actuelle on peut calculer les nouvelles level sets après avancée par trois approches différentes: par mise à jour des level sets de la fissure à propager suivant une méthode de type Hamilton-Jacobi, par mise à jour après un calcul géométrique direct ou par mise à jour à partir d'un maillage propagé reconstruit (le maillage de la fissure est reconstruit à partir des données de propagation et les level sets sont calculées dans un second temps par distance au maillage). Les différentes étapes de la propagation de fissure X-FEM dans *Code_Aster* changent selon l'approche utilisée.

Néanmoins l'étape de calcul de la vitesse d'avancée du fond de fissure est commune à toutes les méthodes (cf. [§4]). Dans le cas où on choisit d'utiliser l'approche par mise à jour sur la base des équations d'Hamilton-Jacobi, on définit les level sets traduisant la nouvelle géométrie de fissure (cf. [§6.2]) et on s'assure que celles-ci gardent des définitions proches de fonctions distances signées (cf. [§13]). La propagation d'une level set nécessite ainsi les trois étapes successives suivantes (bib69) :

- extension de la vitesse connue sur l'iso-zero vers le domaine entier,
- propagation de la level set à partir de ce champ de vitesse,
- réinitialisation de la fonction level set afin de conserver une fonction distance signée.

Au contraire, dans le cas où on utilise le calcul direct, on recalcule tout de suite les nouvelles valeurs des deux level sets (cf. [§52]).

2 Calcul du champ de vitesse de propagation

Les premières étapes consistent à calculer deux champs de vitesses, normale et tangentielle, définissant l'évolution dissociée des deux level sets. On se place dans le cadre d'une propagation en mode mixte.

2.1 Vitesse d'avancée du fond de fissure

L'avancée du fond de fissure en chacun de ses points, évolue dans la base locale selon une loi de Paris. Sa direction est déterminée par le principe du « maximum hoop stress criterion » utilisé classiquement en mécanique linéaire élastique de la rupture (bib69).

$$V = V_N \cdot e_1 + V_T \cdot e_2 \quad \text{dans la base locale } (e_1, e_2, e_3)$$
$$\text{avec } \begin{cases} V = C \cdot G^m (\sin \beta \cdot e_1 + \cos \beta \cdot e_2) \\ \beta = 2 \arctan \left[\frac{1}{4} \cdot \left(\frac{K_I}{K_{II}} - \text{sign}(K_{II}) \cdot \sqrt{\left(\frac{K_I}{K_{II}} \right)^2 + 8} \right) \right] \end{cases}$$

où C et m sont des caractéristiques du comportement en fatigue du matériau, G est le taux de restitution d'énergie local issu de la méthode G -Thêta, et K_I et K_{II} sont les facteurs d'intensité des contraintes en mode I et II

Remarque :

Tout comme pour G , l'angle de propagation β est calculé par l'opérateur `CALC_G` [U4.82.03], l'opérateur de propagation se contentant de lire les données G et β dans le tableau issu de l'opérateur `CALC_G`.

2.2 Champ de vitesse étendu au maillage

Les vitesses normale et tangente sont connues uniquement pour les points d'intersection entre le fond de la fissure et les faces des éléments du maillage. Ces points ne sont pas des nœuds du maillage. Pour la mise à jour des level sets après une propagation, la vitesse doit être connue pour chaque nœud du maillage [§11]. Il est donc nécessaire qu'on étende la vitesse à tous les nœuds du maillage à partir des valeurs sur le fond de fissure. Pour cela on projette chaque nœud sur le fond de fissure en direction normale et on assigne au nœud la vitesse du point qui est la projection du nœud.

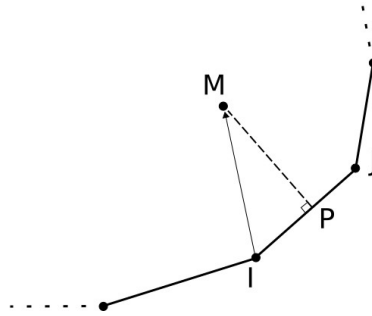


Figure 2.2-a: projection du nœud M sur le fond de fissure

La projection est mise en œuvre selon l'algorithme suivant (Figure 2.2-a):

- Boucle sur les nœuds M du maillage
- Initialisation: $d_{min} = \text{réel max}$
- Boucle sur les segments $\vec{IJ}$: entre deux points I et J consécutifs du fond de fissure
- On calcule $s = \frac{\vec{IJ} \cdot \vec{IM}}{\|\vec{IJ}\|^2}$
- Si $s < 0$, alors $s = 0$
- Si $s > 1$, alors $s = 1$
- Si on projette le nœud M sur $\vec{IJ}$, on obtient le point P . La position de ce point est déterminée par le vecteur $\vec{IP} : \vec{IP} = s \cdot \vec{IJ}$
- On calcule la distance entre le nœud M et sa projection $P : d = \|\vec{MP}\|$
- Si $d < d_{min}$, alors on stocke d_{min} , $I_{min} = I$, $J_{min} = J$ et $s_{min} = s$
- Fin boucle
- Fin boucle

Le paramètre s_{min} donne la position de la projection P de chaque nœud M du maillage par rapport à la longueur du vecteur $\vec{I_{min}J_{min}}$ du fond de fissure auquel la projection P appartient:

$$s = \frac{\vec{IJ} \cdot \vec{IM}}{\|\vec{IJ}\|^2} = \frac{\|\vec{IP}\| \cdot \|\vec{IJ}\|}{\|\vec{IJ}\|^2} = \frac{\|\vec{IP}\|}{\|\vec{IJ}\|}$$

La position de la projection P peut donc être regardée comme position paramétrique sur le vecteur $\vec{I_{min}J_{min}}$. Cela peut être utilisé pour calculer la valeur du vecteur vitesse V à assigner au nœud M .

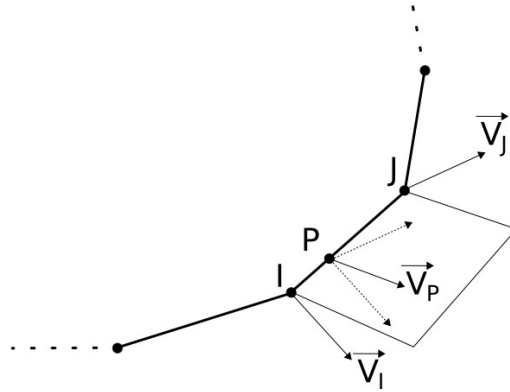


Figure 2.2-b: détermination de la vitesse du point N

On ramène ce problème à la détermination du vecteur vitesse V_P d'un point P du segment $\overline{IJ}$ quand les vecteurs vitesse sont connus aux points I et J (Figure 2.2-b).

La méthode utilisée pour le calcul de V_P est arbitraire. Si on regarde la signification physique de la vitesse, on peut dire qu'elle change graduellement entre les deux vecteurs limites V_I et V_J avec la position du point P . Pour la valeur du $\|V_P\|$ on peut donc utiliser une interpolation linéaire entre les deux valeurs $\|V_I\|$ et $\|V_J\|$:

$$\|V_P\| = (\|V_J\| - \|V_I\|) \cdot s_{min} + \|V_I\|$$

Pour la direction de V_P , on peut déterminer l'angle entre V_I et V_J et utiliser encore une fois une interpolation linéaire sur l'angle calculé, selon l'algorithme suivante:

- on calcule le vecteur normal au plan formé par les vecteurs V_I et V_J : $v_3 = V_I \wedge V_J$
- on se ramène dans une base locale formée par les vecteurs V_I et v_3 . On calcule le dernier axe de la base: $v_2 = v_3 \wedge V_I$. Ce vecteur est dans le plan formé par les vecteurs V_I et V_J .
- on calcule les vecteurs unitaire de la base dans ce plan:

$$u_1 = \frac{V_I}{\|V_I\|} \text{ et } u_2 = \frac{v_2}{\|v_2\|}$$

- on calcule l'angle entre les deux vecteurs vitesse V_I et V_J :

$$\alpha = \arccos \left(\frac{\sum_{i=1}^3 V_{Ii} \cdot V_{Ji}}{\|V_I\| \cdot \|V_J\|} \right)$$

- on calcule le vecteur unitaire de la vitesse au point P par interpolation linéaire:

$$u_P = \cos(\alpha s_{min}) \cdot u_1 + \sin(\alpha s_{min}) \cdot u_2$$

- enfin on calcule le vecteur vitesse au point P en utilisant la valeur $\|V_P\|$ déjà calculée:

$$V_P = \|V_P\| \cdot u_P$$

2.3 Calcul de V_N et V_T pour chaque nœud du maillage

Pour faire évoluer la level set normale convenablement [§9] et pour intégrer les deux équations de mise à jour des level set [§11], on doit connaître les vecteurs vitesse normale V_N et tangente V_T pour chaque nœud du maillage. Pour cela il suffit de calculer les directions normale et tangente des level sets en utilisant les gradients des level sets.

Malheureusement cette solution n'est pas applicable à tous les nœuds. En effet, après la mise à jour, la propriété d'orthogonalité des level sets n'est pas garantie [§13] pour tous les nœuds. Cette propriété

est très importante pour garantir l'orthogonalité des vecteurs vitesse V_N et V_T . On doit donc utiliser une autre solution.

En regardant la façon utilisée pour étendre le champ de vitesse aux nœuds du maillage [§5,§9], on envisage la solution suivante:

1. si la level set tangente au nœud est négative: elle évolue dans une direction parallèle à la surface de la fissure. On utilise donc les gradients des level sets pour calculer les directions normale et tangente. En effet, cela signifie que le nœud est dans l'espace de la fissure existante et que la valeur de la level set normale ne change pas pendant la mise à jour des level sets contrairement à la valeur de la level set tangente [§9].
2. si la level set tangente au nœud est positive, on utilise la base locale de la projection du nœud sur le fond de fissure. En effet, l'évolution des level sets doit être cohérente avec les modifications du fond de fissure.

Pour le cas 1 ci-dessus, le calcul des directions normale et tangente ne pose pas des problèmes parce que les gradients des level sets φ aux nœuds sont toujours connus: $V_N = V_N^P \cdot \nabla \varphi_n$ et $V_T = V_T^P \cdot \nabla \varphi_t$. Toutefois les composantes normale V_N^P et tangente V_T^P de la vitesse V_P doivent être calculées par rapport à la base locale du fond de fissure au point P en utilisant la solution suivante décrite pour le cas 2.

Dans ce cas, le calcul est plus difficile parce que la base locale du fond de fissure est connue uniquement pour les points d'intersection entre le fond de la fissure et les faces des éléments du maillage. La situation est visible en figure 2.3-a: on connaît la base locale aux points I et J et on veut calculer la base locale au point P , projection d'un nœud sur le fond de fissure. L'orientation de cette base doit être comprise entre les orientations de la base au point I et de la base au point J selon la position du point P . Cette dernière est bien décrite par le paramètre s_{min} déjà calculé [§5].

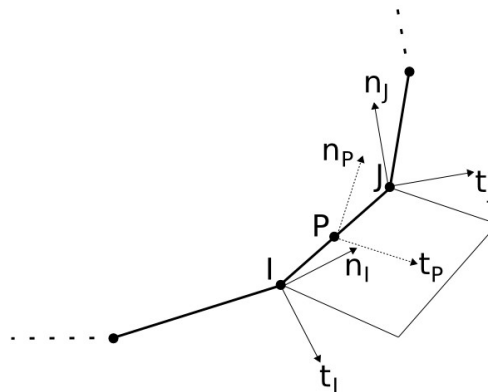


Figure 2.3-a: détermination de la
base locale au point P

Si on ne considère pas la translation de l'origine, le changement d'orientation de la base locale peut être vu comme une rotation de la base à partir de l'orientation au point I , pour laquelle $s_{min}=0$. On a la rotation maximale pour $s_{min}=1$, c'est à dire au point J . L'orientation de la base au point P est obtenue en utilisant une rotation inférieure proportionnelle à la valeur de s_{min} .

En utilisant le théorème d'Euler, on peut toujours décrire une rotation quelconque comme une seule rotation (angle d'Euler) autour d'un seul axe (axe d'Euler). La direction de ce dernier et la valeur de la rotation changent pour le segment IJ le long du fond de fissure.

La base locale au point P peut donc être obtenue par une rotation autour de l'axe d'Euler calculé pour la rotation qui transforme la base au point I dans la base au point J . La valeur de l'angle de rotation peut être calculée simplement en multipliant l'angle d'Euler par s_{min} . Dans le paragraphe suivant [§8], on décrit les équations qu'on utilise pour ce calcul.

Après la détermination de la base locale $\mathbf{t}_P$ et $\mathbf{n}_P$ au point P , on peut calculer la valeur des composantes de la vitesse V_P : $V_N^P = V_P \cdot \mathbf{n}_P$ et $V_T^P = V_P \cdot \mathbf{t}_P$. Enfin on calcule les deux vecteurs V_N et V_T : $V_N = V_N^P \cdot \mathbf{n}_P$ et $V_T = V_T^P \cdot \mathbf{t}_P$.

En résumé, pour chaque nœud M du maillage on calcule la projection P sur le fond de fissure. Ensuite on calcule la base locale $\mathbf{t}_P$ et $\mathbf{n}_P$ au point P et les composantes de la vitesse V_P (qui ont déjà été calculés [§5]):

$$\begin{cases} V_N^P = V_P \cdot \mathbf{n}_P \\ V_T^P = V_P \cdot \mathbf{t}_P \end{cases}$$

Enfin on calcule les vecteurs V_N et V_T au nœud M :

$$\begin{aligned} \varphi_i \leq 0 & \begin{cases} V_N = V_N^P \cdot \nabla \varphi_n \\ V_T = V_T^P \cdot \nabla \varphi_t \end{cases} \\ \varphi_i > 0 & \begin{cases} V_N = V_N^P \cdot \mathbf{n}_P \\ V_T = V_T^P \cdot \mathbf{t}_P \end{cases} \end{aligned}$$

où φ_n et φ_t sont respectivement la level set normale et la level set tangente.

2.4 Calcul de la base locale au point P

Comme déjà décrit dans le paragraphe précédent [§6], on connaît la base locale aux points I et J et on veut calculer la base locale au point P en utilisant une rotation.

La première étape est le calcul de l'axe et de l'angle d'Euler qui décrivent la rotation qu'on doit appliquer à la base $(\mathbf{t}_I, \mathbf{n}_I, \mathbf{b}_I)$ au point I pour obtenir la base $(\mathbf{t}_J, \mathbf{n}_J, \mathbf{b}_J)$ au point J . L'algorithme utilisé est le suivant:

- on calcule le troisième axe $\mathbf{b}$ de la base locale aux points I et J . En fait, uniquement les vecteurs unitaires $\mathbf{t}$ et $\mathbf{n}$ sont stockés:

$$\begin{cases} \mathbf{b}_I = \mathbf{t}_I \wedge \mathbf{n}_I \\ \mathbf{b}_J = \mathbf{t}_J \wedge \mathbf{n}_J \end{cases}$$

- on calcule la matrice de rotation d'Euler entre les deux bases:

$$\underline{\underline{R_E}} = \begin{bmatrix} \mathbf{t}_J \cdot \mathbf{t}_I & \mathbf{n}_J \cdot \mathbf{t}_I & \mathbf{b}_J \cdot \mathbf{t}_I \\ \mathbf{t}_J \cdot \mathbf{n}_I & \mathbf{n}_J \cdot \mathbf{n}_I & \mathbf{b}_J \cdot \mathbf{n}_I \\ \mathbf{t}_J \cdot \mathbf{b}_I & \mathbf{n}_J \cdot \mathbf{b}_I & \mathbf{b}_J \cdot \mathbf{b}_I \end{bmatrix}$$

Chaque colonne de la matrice représente un vecteur de la base au point J $(\mathbf{t}_J, \mathbf{n}_J, \mathbf{b}_J)$ dans la base locale au point I .

- on calcule l'angle d'Euler en utilisant les éléments de la matrice $\underline{\underline{R_E}}$:

$$\vartheta_E = \arccos\left(\frac{R_{11} + R_{22} + R_{33} - 1}{2}\right)$$

- on calcule l'axe d'Euler $\mathbf{e} = [e_1 \ e_2 \ e_3]^T$:

$$\begin{cases} e_1 = \frac{R_{23} - R_{32}}{2 \cdot \sin \vartheta_E} \\ e_2 = \frac{R_{31} - R_{13}}{2 \cdot \sin \vartheta_E} \\ e_3 = \frac{R_{12} - R_{21}}{2 \cdot \sin \vartheta_E} \end{cases}$$

L'angle et l'axe d'Euler calculés sont les même pour tous les points P sur l'arête définie par les points I et J .

La deuxième étape est le calcul de la base locale pour un point P pour lequel la position est déterminée par le paramètre s_{min} [§5]. L'algorithme utilisé est le suivant:

- on calcule l'angle de rotation effective à utiliser: $\vartheta_P = \vartheta_E \cdot s_{min}$
- on calcule la matrice de la rotation entre la base locale au point I et la base locale au point J pour une rotation autour de l'axe d'Euler e égale à ϑ_P :

$$\underline{T}_P = \cos(\vartheta_P) \cdot \underline{I} + (1 + \cos(\vartheta_P)) \cdot e \cdot e^T - \sin(\vartheta_P) \cdot \underline{\epsilon}$$

où la matrice $\underline{\epsilon}$ est la suivante:
$$\underline{\epsilon} = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}$$

- on calcule les axes t_P et n_P de la base locale au point P dans la base locale au point I :

$$\begin{cases} t_P(t_I, n_I, b_I) = \underline{T}_P \cdot [1 \ 0 \ 0]^T \\ n_P(t_I, n_I, b_I) = \underline{T}_P \cdot [0 \ 1 \ 0]^T \end{cases}$$

Les deux vecteurs sont donc coïncidents avec la première et la deuxième colonne de la matrice $\underline{T}_P$. En utilisant l'équation ci-dessus pour le calcul de cette matrice, on peut donc écrire:

$$\begin{aligned} t_P(t_I, n_I, b_I) &= \begin{bmatrix} \cos(\vartheta_P) + (1 - \cos(\vartheta_P)) \cdot e_1^2 \\ (1 - \cos(\vartheta_P)) \cdot e_1 \cdot e_2 - \sin(\vartheta_P) \cdot e_3 \\ (1 - \cos(\vartheta_P)) \cdot e_1 \cdot e_3 + \sin(\vartheta_P) \cdot e_2 \end{bmatrix} \\ n_P(t_I, n_I, b_I) &= \begin{bmatrix} (1 - \cos(\vartheta_P)) \cdot e_1 \cdot e_2 + \sin(\vartheta_P) \cdot e_3 \\ \cos(\vartheta_P) + (1 - \cos(\vartheta_P)) \cdot e_2^2 \\ (1 - \cos(\vartheta_P)) \cdot e_2 \cdot e_3 - \sin(\vartheta_P) \cdot e_1 \end{bmatrix} \end{aligned}$$

- enfin on peut exprimer les vecteurs t_P^I et n_P^I dans la base globale du maillage, qui est utilisée pour tous les calculs. Si on dénote la composante i du vecteur $t_P(t_I, n_I, b_I)$ par $t_P(t_I, n_I, b_I)(i)$ et la même composante du vecteur $n_P(t_I, n_I, b_I)$ par $n_P(t_I, n_I, b_I)(i)$, on peut écrire:

$$\begin{cases} t_P = t_P(t_I, n_I, b_I)(1) \cdot t_I + t_P(t_I, n_I, b_I)(2) \cdot n_I + t_P(t_I, n_I, b_I)(3) \cdot b_I \\ n_P = n_P(t_I, n_I, b_I)(1) \cdot t_I + n_P(t_I, n_I, b_I)(2) \cdot n_I + n_P(t_I, n_I, b_I)(3) \cdot b_I \end{cases}$$

2.5 Ajustement du champ de vitesse normale

Afin d'empêcher le déplacement de la fissure préexistante, et de faire évoluer l_{sn} convenablement, il est nécessaire d'ajuster le champ de vitesse normale (bib69).

On veut avoir une trajectoire de fissure relativement lissée, telle que :

$$\bar{V}_N(lst) = \begin{cases} 0 & \text{si } lst \leq 0 \text{ (et notamment en } lst = 0) \\ V & \text{si } lst = V \cdot \Delta t \end{cases}$$

La valeur du Δt est égale au temps total d'intégration des équations de mise à jour des level sets [§ 11].

Ainsi, le fond de fissure avance bien selon V_T et V_N calculées précédemment. On fait une approximation linéaire en lst .

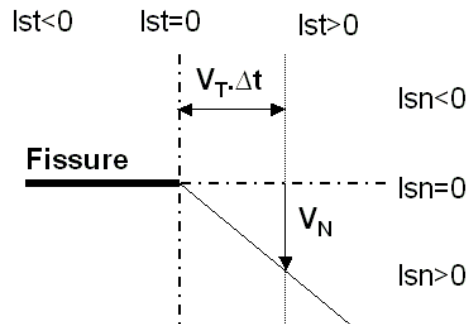


Figure 2.5-a : Ajustement de la vitesse normale

La formule appliquée sur tous les nœuds du maillage est la suivante :

$$\bar{V}_N = h(lst) \cdot \frac{V_N \cdot lst}{V_T \cdot \Delta t} \quad \text{avec} \quad h(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x > 0 \end{cases}$$

où V_N et V_T sont les champs de vitesse nodaux, et Δt le pas de temps choisi (voir paragraphe [§3]).

3 Mise à jour des level sets

Après la propagation, la géométrie de la fissure change et les level sets doivent évoluer pour bien décrire cette nouvelle géométrie. Même si on peut recalculer les level sets en utilisant leur définition de distance signée, en pratique cette solution est à éviter pour une question de performance du code. On peut recalculer plus rapidement les level sets en utilisant les équations de mise à jour des level sets qui décrivent leur évolution suite à la propagation de la fissure.

Dans les équations suivantes, on utilisera les symboles φ_n , φ_t et φ pour indiquer respectivement la level set normale (lsn), la level set tangente (lst) et une level set en général (lsn ou lst).

3.1 Équations de mise à jour

Les level sets sont mises à jour sur tous les nœuds selon l'équation de Hamilton-Jacobi suivante (bib69):

$$\frac{\partial \varphi}{\partial t} = -V_\varphi \cdot \|\nabla \varphi\|$$

où φ est la level set, $\nabla \varphi$ est son gradient et V_φ la vitesse de propagation de la level set φ (V_T ou V_N).

Cette équation est connue comme équation d'évolution des level sets. Dans (bib69) et (bib69) il est montré qu'une erreur est présente dans cette formulation de l'équation et que la forme correcte à utiliser est plutôt la suivante:

$$\frac{\partial \varphi}{\partial t} + \nabla \varphi \cdot V = 0$$

L'intégration de l'équation pose deux problèmes (bib69):

- le nombre d'itérations d'intégration est important si un schéma explicite est utilisé (par exemple Runge-Kutta) à cause de la limitation sur le pas de temps utilisable à chaque itération (condition CFL). L'ajustement du champ de vitesse normale ([§9]) est à l'origine de la très forte diminution de la valeur maximale du pas de temps utilisable et du conséquent incrément du nombre d'itérations nécessaires. Pour éviter ce problème, un schéma implicite peut être utilisé mais cela complique l'écriture informatique. Comme troisième alternative, on peut utiliser une méthode de type « fast marching method » (bib69) qui calcule la solution directement et sans itérations (bib69, bib69). Toutefois des erreurs peuvent être présentes dans la solution calculée par cette dernière méthode aux bords du domaine de calcul (bib69, bib69).
- en présence d'une valeur élevée de l'angle de propagation β ([§4]), la solution obtenue par intégration de l'équation des level set ci-dessus peut présenter des instabilités qui vont perturber la convergence des algorithmes d'intégration au cours des itérations de propagation successives.

Une équation alternative de mise à jour est donc proposée dans (bib69). Elle repose sur l'interprétation géométrique de la modification des level sets consécutive à la propagation de la fissure. Si on regarde l'iso-zéro de la level set tangente ($lst=0$ de la figure 2.5-a), on peut dire que chaque point de cette surface se déplace de la quantité $V_T \cdot \Delta t$. Par conséquent, la valeur de la level set au point diminue de la même quantité. Si on regarde l'iso-zéro de la level set normale ($lsn=0$ de la figure 2.5-a), on peut dire que dans la partie du volume de matériau sain ($lst > 0$) la level set est soumise à une rotation autour du fond de fissure et que chaque point se déplace suivant la direction normale à l'iso-zéro d'une quantité $V_N \cdot \Delta t$. C'est la variation linéaire de la vitesse V_N avec lst ([§9]) qui permet d'obtenir le bon déplacement en chaque point. En généralisant au cas 3D comme dans (bib69), on peut donc écrire les équations de mise à jour suivantes:

$$\Delta \varphi_n = -(V_N \cdot \nabla \varphi_n) \cdot \Delta t$$

$$\Delta \varphi_t = -(\mathbf{V}_T \cdot \nabla \varphi_t) \cdot \Delta t$$

L'avantage de ces équations est évident : il s'agit d'équations algébriques explicites qui ne nécessitent l'utilisation d'aucun schéma itératif d'intégration. La solution peut être calculée directement avec un temps de calcul très faible. De plus, aucun problème d'instabilité n'est présent même pour des angles de propagation β élevés comme montré dans (bib69).

La valeur de Δt à utiliser dans les équations de mise à jour des level sets peut être facilement calculée à partir de l'avancée maximale Δa_{max} de la fissure donnée par l'utilisateur :

$$\Delta t_{tot} = \frac{\Delta a_{max}}{\max_{i \in \text{fond}} \|\mathbf{V}(\mathbf{x}_i)\|}$$

où le vecteur $\mathbf{V}(\mathbf{x}_i)$ est le vecteur de la vitesse de propagation ($\mathbf{V} = \mathbf{V}_N + \mathbf{V}_T$) d'un point $\mathbf{x}_i$ du fond de fissure. Il faut observer que dans ce contexte on prend comme point du fond de fissure les points physiques à l'intersection entre le fond de fissure et les faces des éléments du maillage, pour lesquels on connaît la vitesse et l'angle de propagation, mais qu'on prend aussi les points géométriques du front qui sont la projection normale des nœuds du maillage sur le front lui-même [§5]. Toutefois la vitesse de propagation est calculée pour ces derniers points en utilisant une interpolation linéaire [§5] et pour chaque arête qui forme le fond de fissure la vitesse maximale est obtenue à l'une de ses deux extrémités. On peut donc limiter la recherche de la vitesse maximale aux points d'intersection entre le fond de fissure et les faces des éléments du maillage.

Remarque:

Dans la description ci-dessus, les symboles t et Δt ont été utilisés pour indiquer le temps. Toutefois la grandeur décrite par ces symboles est effectivement le nombre de cycles de fatigue qui cause l'avancée de la fissure. En fait, si on regarde l'équation utilisée ci-dessus pour le calcul de Δt , Δt est obtenu par division entre la distance d'avancée de la fissure (par exemple : millimètres) et la vitesse d'avancée. La vitesse calculée en utilisant une loi de propagation (par exemple la loi de Paris) est donnée comme une avancée de la fissure par cycle de fatigue (par exemple : millimètres/cycle). On en déduit donc que Δt représente le nombre de cycles de fatigue que l'on doit appliquer pour avoir l'avancée maximale voulue. Le lien entre le nombre de cycles de fatigue et le temps est donné par la fréquence d'application du chargement. Cela est vrai pour toutes les équations utilisées et on peut dire que toutes les grandeurs qui se ramènent au temps (t , Δt , ...) expriment en fait un nombre de cycles de fatigue.

4 Réinitialisation et réorthogonalisation des level sets

4.1 Nécessité de garder des fonctions de distances signées

Au cours de leurs évolutions, les level sets doivent impérativement garder une définition relativement proche de fonctions de distance signée.

La première raison en est le calcul des facteurs d'intensité de contraintes en fond de fissure par la méthode G -Thêta, pour lequel les coordonnées polaires (r, θ) sont définies dans la base locale comme $r = \sqrt{lsn^2 + lst^2}$ et $\theta = \arctan(lsn/lst)$; l'utilisation de distances signées pour le calcul des coordonnées polaires est donc nécessaire.

De plus, lors de la propagation de fissure, l'équation de mise à jour présentée plus haut (paragraphe [§3]) met en jeu les normes des gradients des level sets, qui doivent rester suffisamment proches de l'unité, car si $|\nabla \varphi|$ ne vaut pas 1, la fissure n'avance pas avec la vitesse V_φ définie.

Après la phase de mise à jour, il arrive que les level sets perdent leur sens de fonction de distance signée, notamment dans le cas d'une propagation hors plan (cas d'une sollicitation où $K_{II} \neq 0$), qui introduit une vitesse normale (voir Figure 4.1-a).

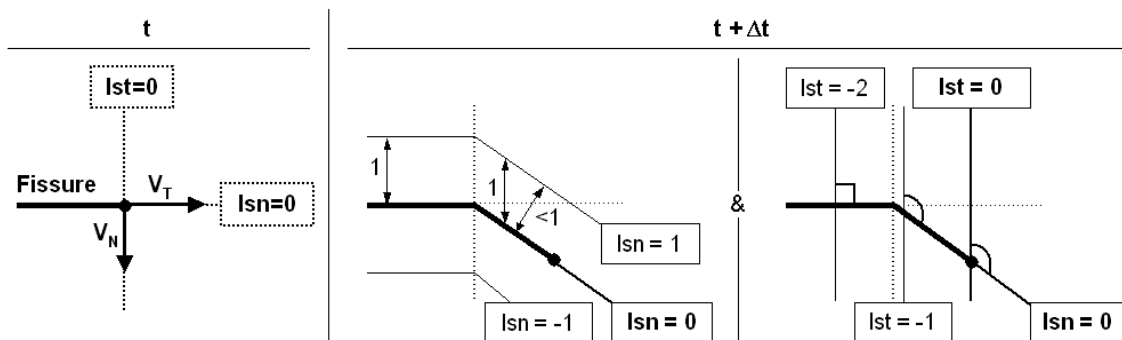


Figure 4.1-a : Propagation de level sets et perte de fonction de distance signée et d'orthogonalité

Après propagation il est donc nécessaire de passer par une étape dite de « réinitialisation » dont le but sera de minimiser $||\nabla lsn| - 1|$ et $||\nabla lst| - 1|$ ainsi qu'une étape de « réorthogonalisation » dont l'objectif sera de minimiser $|\nabla lsn \cdot \nabla lst|$. En effet, les level sets ne sont rigoureusement définies que si, en tout point de l'espace, $|\nabla lsn| = 1$, $|\nabla lst| = 1$ et $\nabla lsn \cdot \nabla lst = 0$.

Cependant, dans le cas d'une fissure non plane, il est impossible d'avoir deux fonctions dont les gradients sont orthogonaux entre eux et de normes unitaires en tout point de l'espace. L'antagonisme entre les deux propriétés apparaît sur la Figure 4.1-b sur lesquelles sont définies, à partir d'une iso-zéro de lsn, une fonction lst soit normée, soit orthogonale.

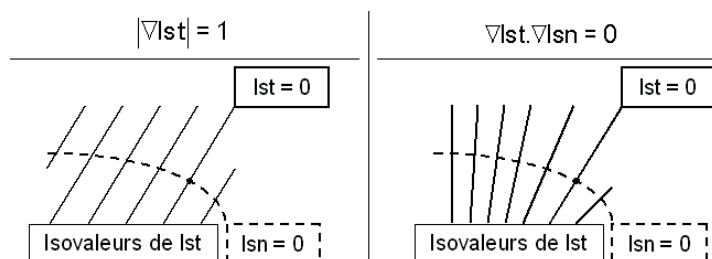


Figure 4.1-b : Antagonisme Initialisation - Orthogonalisation

Il est donc nécessaire de privilégier l'une ou l'autre de ces propriétés. Étant donnée l'utilisation principale de ces fonctions pour la méthode G -Thêta, il apparaît plus important que les gradients soient bien normés, plutôt qu'orthogonaux, afin que les coordonnées polaires (r, θ) aient un sens.

Afin d'avoir une base locale orthonormée, on impose aussi sur le fond de fissure que les gradients soient normés et orthogonaux.

L'enchaînement des étapes proposé dans (bib69) est donc:

- réinitialisation de l_{sn} :

L'iso-zéro de l_{sn} , d'importance majeure pour la définition de fissure, est inchangée ;

- réorthogonalisation de l_{st} par rapport à l_{sn} fixé :

l_{st} est redressée par rapport à l_{sn} sans modifier le fond de fissure $(l_{sn}=0) \cap (l_{st}=0)$;

- réinitialisation de l_{st} :

La level set l_{st} est initialisé par rapport à son iso-zéro ayant été orthogonalisé.

4.2 Équations de Hamilton-Jacobi

Les équations de réinitialisation et de réorthogonalisation dont l'objectif est de minimiser les quantités $\|\nabla \varphi - 1\|$ et $|\nabla \varphi \cdot \nabla \psi|$, respectivement, sont exprimées en fonction d'un temps virtuel τ , que l'on fera évoluer jusqu'à un état stationnaire.

L'équation de réinitialisation de la level set φ est :
$$\frac{\partial \varphi}{\partial \tau} = -\text{sign}(\varphi) \cdot (|\nabla \varphi| - 1)$$

L'équation de réorthogonalisation de φ_t par rapport à φ_n est :
$$\frac{\partial \varphi_t}{\partial \tau} = -\text{sign}(\varphi_n) \cdot \frac{\nabla \varphi_n}{|\nabla \varphi_n|} \cdot \nabla \varphi_t$$

Ces deux équations peuvent donc se mettre sous la forme générale :

$$\frac{\partial \varphi}{\partial \tau} + F(\nabla \varphi, x) \cdot |\nabla \varphi| = f(x)$$

où φ est la level set, $\nabla \varphi$ son gradient, et x la position dans l'espace,

avec $F(\nabla \varphi, x) = \text{sign}(\varphi)$ et $f(x) = \text{sign}(\varphi)$ pour la phase de réinitialisation,

et $F(\nabla \varphi, x) = \text{sign}(\psi) \cdot \frac{\nabla \psi}{|\nabla \psi|} \cdot \frac{\nabla \varphi}{|\nabla \varphi|}$ et $f(x) = 0$ pour la phase de réorthogonalisation.

Une définition rigoureuse des level sets n'étant nécessaire que dans une zone localisée autour du fond de fissure, la convergence de ces équations sera estimée par un résidu local dans un tore dont le rayon est une donnée utilisateur qui doit impérativement être plus grand que le rayon supérieur d'intégration de la méthode G -Thêta, pour que les résultats soient corrects.

4.3 Schémas explicites de résolution

Dans la suite on analysera deux schémas d'intégration pour la résolution des équations de réinitialisation et réorthogonalisation. Le premier schéma (méthode « Simplexe ») peut être appliqué seulement aux éléments simplexes (triangles en 2D et tétraèdres en 3D) et à certains éléments hexaèdres. Le deuxième schéma (méthode « Upwind ») peut être appliqué seulement à des éléments rectangulaires en 2D et hexaèdres en 3D si certaines conditions sont vérifiées.

Dans tous les cas, s'agissant de schémas explicites, le pas d'intégration ne peut pas excéder une certaine valeur critique. Les équations à résoudre étant équivalentes à des équations de convection dont la vitesse de convection est $|F(\nabla \varphi, x)|$, le pas de temps critique peut être estimé par :

$$\Delta t_{CFL} = \min \left(\frac{h(x)}{|V_{conv}|} \right) = \min \left(\frac{h(x)}{|F(\nabla \varphi, x)|} \right)$$

avec $h(x)$ longueur caractéristique d'éléments.

On supposera : $|sign(\varphi)| \approx 1$ et $|sign(\psi) \cdot \frac{\nabla \psi}{|\nabla \psi|} \cdot \frac{\nabla \varphi}{|\nabla \varphi}|| \leq 1$

Une condition CFL peut donc être : $\Delta \tau \leq \Delta \tau_{CFL} = \min(h(x))$

4.3.1 Méthode « Simplexe »

La méthode présentée ci-dessous est celle proposée dans (bib69). Elle consiste à évaluer, par des coefficients K_i , la contribution de chaque élément à la réactualisation de la level set en un nœud, puis à trier les contributions positives afin d'obtenir un schéma monotone. Sa principale limite provient du fait qu'elle est développée uniquement pour les éléments simplexes (triangles en 2D et tétraèdres en 3D). Elle pourra cependant être étendue à certains éléments hexaèdres (voir paragraphe [§4.3.3]).

Soit l'équation de la forme : $\frac{\partial \varphi}{\partial \tau} + H(\nabla \varphi) = f(x)$,

avec : $H(\nabla \varphi) = sign(\varphi) \cdot |\nabla \varphi|$ pour la réinitialisation de φ ,

et : $H(\nabla \varphi) = sign(\psi) \cdot \frac{\nabla \psi}{|\nabla \psi|} \cdot \nabla \varphi$ pour la réorthogonalisation de φ par rapport à ψ .

On montre que l'équation discrétisée sur l'espace décomposé en t éléments simplexes $T_l | l \in (1, 2, \dots, t)$ est approchée ainsi :

$$\varphi_j^{n+1} = \varphi_j^n - \Delta t \cdot H_j(\nabla \varphi_1^n, \nabla \varphi_2^n, \dots, \nabla \varphi_t^n, x)$$

avec $\nabla \varphi_i^n$ gradient de φ calculé sur le simplexe i à l'instant n ,

$$\text{et } H_j(\nabla \varphi_1^n, \nabla \varphi_2^n, \dots, \nabla \varphi_t^n, x) = \frac{\sum_{l=1}^t \alpha_j^l \cdot (H(\nabla \varphi) - f(x))_{T_l}}{\sum_{l=1}^t \alpha_j^l \cdot |T_l|},$$

où : $(\cdot)_{T_l} = \int_{T_l} (\cdot) d\Omega$ est l'intégrale de $(\cdot)$ sur le simplexe T_l ,

: $|T_l|$ est la mesure de l'élément T_l (volume en 3D, aire en 2D),

et $\alpha_j^l = \alpha_j^l(\nabla \varphi_l)$ est un coefficient borné, positif, de la contribution au Hamiltonien numérique H_j du simplexe T_l au nœud j .

La stabilité et la convergence de ces schémas (en négligeant le terme source $f(x)$) a été montrée dans (bib69) :

$$\begin{aligned}\varphi_j^{n+1} &= \varphi_j^n - \Delta t \cdot H_j(\varphi_n) = \varphi_j^n - \Delta t \cdot \left[\frac{\partial H_j}{\partial \varphi_i} (\varphi_i^n - \varphi_j^n) \right] = \varphi_j^n \left[1 + \Delta t \cdot \frac{\partial H_j}{\partial \varphi_i} \right] \\ &= \varphi_j^n \left[1 + \Delta t \cdot \frac{\partial H_j}{\partial \varphi_i} \right]\end{aligned}$$

converge si $0 \leq 1 + \Delta t \cdot \frac{\partial H_j}{\partial \varphi_i} \leq 1$.

Donc : $\varphi_j^{n+1} = \varphi_j^n - \Delta t \cdot H_j(\varphi_n)$ converge si $\frac{\partial H_j}{\partial \varphi_i} \leq 0$ et

$$0 \leq \Delta t \leq - \left(\frac{\partial H_j}{\partial \varphi_i} \right)^{-1}$$

À l'intérieur d'un élément, la fonction φ est interpolée linéairement :

$$\varphi(x) = \sum_{i=1}^{d+1} N_i(x) \cdot \varphi_i \quad \text{où}$$

- : d est la dimension de l'espace
- : φ_i est la valeur de φ au noeud i
- : N est la fonction de forme de l'élément au noeud i

Le gradient de φ dans l'élément simplexe est :

$$\nabla \varphi = \sum_{i=1}^{d+1} \nabla N_i \cdot \varphi_i \quad \text{où les } \nabla N_i \text{ sont les dérivées des fonctions de forme}$$

Ce gradient est décomposé en terme de normale intérieure $\vec{n}_i$ à la facette (en 3D) ou au coté (en 2D) opposé au noeud i dans l'élément :

$$\nabla \varphi = \frac{1}{d \cdot |T|} \cdot \sum_{i=1}^{d+1} \vec{n}_i \cdot \varphi_i$$

On note que, comme $\sum_{i=1}^{d+1} \nabla N_i = \vec{0}$, on a $\sum_{i=1}^{d+1} \vec{n}_i = \vec{0}$.

Rappelons le théorème d'Euler pour une fonction F homogène de degré p en u :

$$p \cdot F(u) = \frac{\partial F}{\partial u} \cdot u$$

Le Hamiltonien $H(\nabla \varphi)$ étant une fonction homogène de degré $p=1$ en $\nabla \varphi$:

$$(H(\nabla \varphi, x))_T = \frac{|T|}{p} \nabla H \cdot \nabla \varphi = \frac{1}{d \cdot p} \sum_{i=1}^{d+1} (\nabla H \cdot \vec{n}_i) \varphi_i$$

Dans les conditions de stabilité évoquées précédemment, certaines orientations de ∇H par rapport au simplexe T donnent lieu à des évolutions monotones de φ (voir Figure 4.3.1-a).

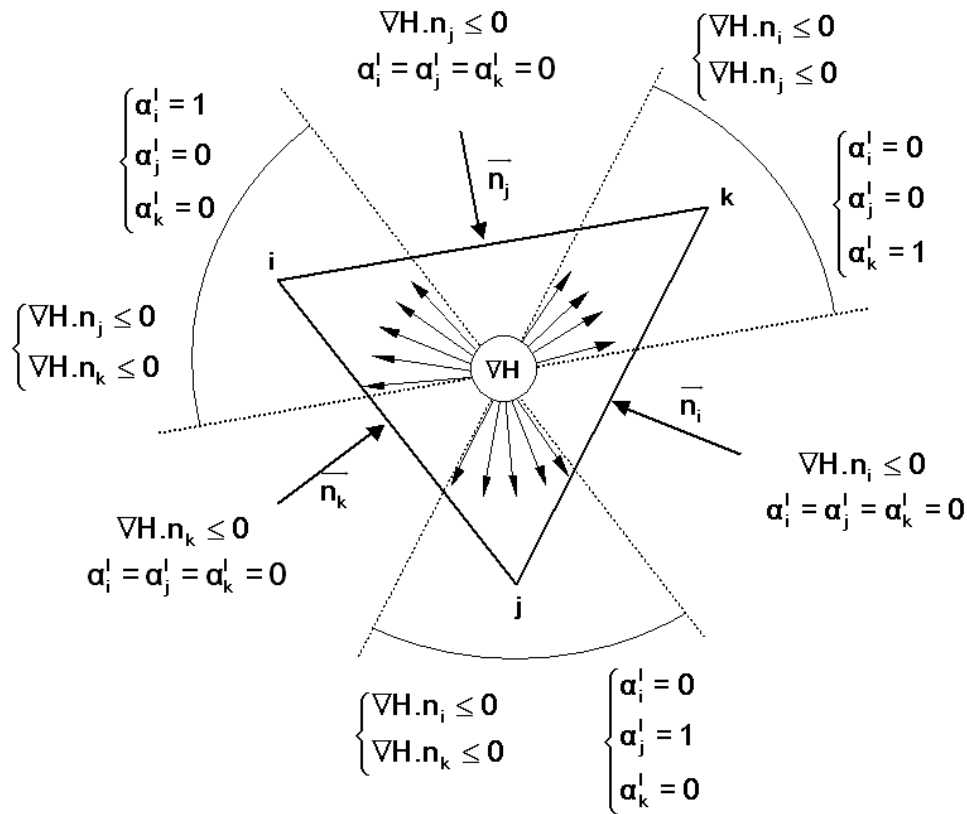


Figure 4.3.1-a : Simplexe 2D avec normales intérieures et coefficients α_j^l (bib69)

Ainsi, si la direction de ∇H est telle que $\nabla H \cdot n_j \leq 0$ et $\nabla H \cdot n_k \leq 0$, alors le nœud i est mis à jour par :

$$\varphi_i^{n+1} = \varphi_i^n - \Delta t \frac{(H(\nabla \varphi^n))_T}{|T|}$$

De même pour le nœud j si $\nabla H \cdot n_i \leq 0$ et $\nabla H \cdot n_k \leq 0$, et du nœud k si $\nabla H \cdot n_i \leq 0$ et $\nabla H \cdot n_j \leq 0$.

Plus généralement, plusieurs éléments peuvent contribuer à la mise à jour d'un nœud. On écrira alors la formule du Hamiltonien numérique généralisé :

$$\varphi_j^{n+1} = \varphi_j^{n+1} - \Delta t \cdot \frac{\sum_{l=1}^t \alpha_j^l \cdot (H(\nabla \varphi))_{T_l}}{\sum_{l=1}^t \alpha_j^l \cdot |T_l|} \quad \text{où}$$

$$\alpha_j^l = \begin{cases} 1 & \text{si } \nabla H \cdot n_i \leq 0, \quad k=1,2,\dots,d+1, \quad j \neq k \\ 0 & \text{sinon} \end{cases}$$

La méthode est cohérente s'il peut être montré que pour chaque nœud il existe au moins un coefficient α_i^l non nul. D'après la Figure 4.3.1-a, cela apparaît évident pour tout nœud situé à l'intérieur du maillage. Pour certains nœuds i en bord de maillage dont tous les α_i^l sont nuls il sera néanmoins nécessaire d'adapter la méthode. Cela fait l'objet du paragraphe [§4.3.2].

En posant :

$$K_i = \frac{\nabla H \cdot \vec{n}_i}{d \cdot p}$$

On peut décomposer $(H(\nabla \varphi, x))_T$ en : $(H(\nabla \varphi))_T = \sum_{i=1}^{d+1} K_i \cdot \varphi_i$

Comme $\sum_{i=1}^{d+1} \vec{n}_i = \vec{0}$, on a $\sum_{i=1}^{d+1} K_i = 0$.

Les K_i représentant les participations par élément du nœud i au calcul de $(H(\nabla \varphi))_T$, on utilise des techniques bien connues (bib69, bib69 et bib69) pour trier les contributions positives de K_i .

On part de la décomposition de la forme canonique :

$$\begin{aligned} \sum_{i=1}^{d+1} K_i \cdot \varphi_i &= \sum_{j=1}^{d+1} K_j^- \cdot \varphi_j + \sum_{i=1}^{d+1} K_i^+ \cdot \varphi_i \\ &= \left(\sum_{l=1}^{d+1} K_l^+ \right)^{-1} \left(\sum_{i=1}^{d+1} K_i^+ \right) \sum_{j=1}^{d+1} K_j^- \cdot \varphi_j - \left(\sum_{l=1}^{d+1} K_l^+ \right)^{-1} \left(\sum_{j=1}^{d+1} K_j^- \right) \sum_{i=1}^{d+1} K_i^+ \cdot \varphi_i \quad \text{car} \\ &\quad \sum_{i=1}^{d+1} K_i^+ + \sum_{i=1}^{d+1} K_i^- = 0 \\ &= \left(\sum_{l=1}^{d+1} K_l^+ \right)^{-1} \left(\sum_{l=1}^{d+1} K_l^+ \right) \left(\sum_{j=1}^{d+1} K_j^- \cdot (\varphi_j - \varphi_i) \right) \end{aligned}$$

Pour arriver à l'expression du Hamiltonien :

$$(H(\nabla \varphi, x))_T = \sum_{i=1}^{d+1} \delta_i \quad \text{avec} \quad \delta_i = K_i^+ \left(\sum_{l=1}^{d+1} K_l^+ \right)^{-1} \left(\sum_{j=1}^{d+1} K_j^- \cdot (\varphi_j - \varphi_i) \right)$$

On définit alors : $\alpha_i^l = \frac{\delta_i}{(H(\nabla \varphi))_{T_i}}$,

avec, dans notre cas, $H(\nabla \varphi, x) \geq 0$.

On aboutit à l'expression du Hamiltonien numérique :

$$H_j(\nabla \varphi_1^n, \nabla \varphi_2^n, \dots, \nabla \varphi_{|T|}^n, x) = \frac{\sum_{l=1}^{\tau} \alpha_j^l \cdot (H(\nabla \varphi))_{T_l}}{\sum_{l=1}^{\tau} \alpha_j^l \cdot |T_l|} = \frac{\sum_{l=1}^{\tau} \delta_j^l}{\sum_{l=1}^{\tau} \alpha_j^l \cdot |T_l|}$$

Les coefficients α_i^l ainsi obtenus sont ceux définis plus haut, mais afin de conserver leur positivité dans le cas de triangulations obtuses, il est proposé dans (bib69) d'introduire la correction supplémentaire :

$$\tilde{\alpha}_i^l = \frac{\max(0, \alpha_i^l)}{\sum_{k=1}^{d+1} \max(0, \alpha_k^l)}$$

Les coefficients $\tilde{\alpha}_i^l$ ainsi obtenus sont compris entre 0 et 1 et conduisent à un schéma robuste qui s'exprime, après réintroduction du terme source :

$$\varphi_j^{n+1} = \varphi_j^n - \Delta t \cdot \frac{\sum_{l=1}^{\tau} \tilde{\alpha}_j^l \cdot (H(\nabla \varphi) - f(x))_{T_l}}{\sum_{l=1}^{\tau} \tilde{\alpha}_j^l \cdot |T_l|}$$

4.3.2 Implémentation pour les éléments tétraèdres

L'algorithme suivi est celui proposé dans (bib69):

Soit l'équation :
$$\frac{\partial \varphi}{\partial \tau} + F(\nabla \varphi, x) \cdot |\nabla \varphi| = f(x)$$

L'espace de dimension d est discrétisé en éléments simplexes T (à $d+1$ nœuds).

• Sur chaque simplexe T , on calcule :

$$\bullet \bar{f}_T = \frac{1}{|T|} \int_T f(\nabla \varphi) dx = \frac{1}{d+1} \cdot \sum_{i=1}^{d+1} f_i$$

$$\bullet |T|$$

$$\bullet (\forall \text{ nœud } i \in T) \quad \vec{n}_i = d \cdot |T| \cdot \nabla N_i$$

• Boucle sur le temps virtuel τ

• Initialisation : $V_i = W_i = 0 (\forall \text{ nœud } i)$

• Sur chaque simplexe T , on calcule :

$$\bullet \bar{F}_T = \frac{1}{|T|} \int_T F(\nabla \varphi) dx = \frac{1}{d+1} \cdot \sum_{i=1}^{d+1} F_i$$

$$\bullet \nabla \varphi_T = \sum_{i=1}^{d+1} \nabla N_i \cdot \varphi_i$$

$$\bullet K_i^T = \frac{\bar{F}_T \cdot \nabla \varphi_T \cdot \vec{n}_i^T}{d \cdot |\nabla \varphi_T|}$$

$$\bullet \delta \varphi_T = \sum_{i=1}^{d+1} K_i^T \cdot \varphi_i$$

$$\bullet \delta \varphi_T^i = K_i^{T^+} \cdot \left(\sum_{j=1}^{d+1} K_j^{T^+} \right)^{-1} \cdot \sum_{l=1}^{d+1} K_l^{T^-} (\varphi_i - \varphi_l)$$

$$\bullet \tilde{\alpha}_i^T = \frac{\max \left(0, \frac{\delta \varphi_i^T}{\delta \varphi_T} \right)}{\sum_{j=1}^{d+1} \max \left(0, \frac{\delta \varphi_j^T}{\delta \varphi_T} \right)}$$

• Boucle sur les éléments T_l

$$(\forall \text{ nœud } i \in T_l)$$

$$\bullet \begin{cases} V_i = V_i + \tilde{\alpha}_i^{T_l} \cdot (\delta \varphi_{T_l} - \bar{f}_{T_l} \cdot |T_l|) \\ W_i = W_i + \tilde{\alpha}_i^{T_l} \cdot |T_l| \end{cases}$$

• Fin boucle

• Boucle sur les nœuds i

$$\bullet \varphi_i^{n+1} = \varphi_i^n - \Delta t \cdot \frac{V_i}{W_i}$$

• Fin boucle

• Estimation de :

$$\frac{\sum_{\substack{i \in \text{zone} \\ d' \text{ estimation}}} (\varphi_i^{n+1} - \varphi_i^n)^2}{\sum_{\substack{i \in \text{zone} \\ d' \text{ estimation}}} (\varphi_i^n)^2}$$

• Si (résidu ≤ tolérance) ou (iteration ≥ nombre max d'itérations) ⇒ Sortie de la boucle

• Fin boucle

4.3.3 Méthode étendue aux éléments hexaèdres à faces opposées parallèles

La méthode précédemment décrite, n'étant théoriquement applicable qu'à des éléments simples (TETRA en 3D et TRIA en 2D), on a tenté de l'étendre autant que possible à certains hexaèdres, en essayant de retrouver pour ces derniers un cadre proche des éléments simples.

Les calculs de $\bar{f}_T$, $\bar{F}_T$, et $|T|$ ne présentent pas de difficultés supplémentaires, mais les fonctions de forme des éléments hexaédriques étant bilinéaires (linéaires pour des simples), les gradients des fonctions de formes des nœuds ∇N_i qui entrent dans le calcul de $\nabla \varphi_T$, $\vec{n}_i$, et des quantités qui en découlent, ne sont plus constants sur un élément donné.

On va alors exprimer les ∇N_i en un même point de l'élément, son centre O , qui dans le cas d'éléments hexaédriques à faces opposées parallèles se trouve être central à l'élément et donc un bon candidat pour une expression moyenne des gradients sur l'élément.

Les valeurs des gradients des fonctions de forme n'étant pas accessibles directement au point O , on va les calculer à partir des valeurs disponibles exprimées aux nœuds sommets de la maille.

Soit l'élément isoparamétrique de référence HEXA8 :

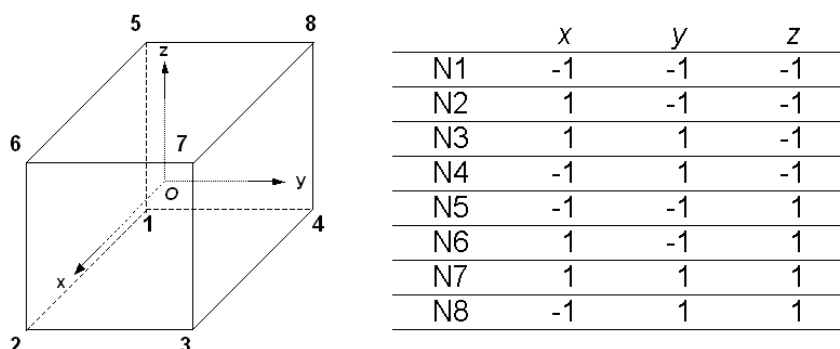


Figure 4.3.3-a : Élément isoparamétrique HEXA8 (voir doc R3.01.01)

Les fonctions de forme des nœuds sont :

$$\begin{aligned} w_1 &= \frac{1}{8}(1-x)(1-y)(1-z) & w_5 &= \frac{1}{8}(1-x)(1-y)(1+z) \\ w_2 &= \frac{1}{8}(1+x)(1-y)(1-z) & w_6 &= \frac{1}{8}(1+x)(1-y)(1+z) \\ w_3 &= \frac{1}{8}(1+x)(1+y)(1-z) & w_7 &= \frac{1}{8}(1+x)(1+y)(1+z) \\ w_4 &= \frac{1}{8}(1-x)(1+y)(1-z) & w_8 &= \frac{1}{8}(1-x)(1+y)(1+z) \end{aligned}$$

Le gradient de la fonction de forme du nœud 1 est :

$$\nabla w_1 = \frac{1}{8} \begin{pmatrix} -1+y+z-yz \\ -1+x+z-xz \\ -1+x+y-xy \end{pmatrix} \quad \text{d'où} \quad \nabla w_1(N_1) = \frac{1}{2} \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} \quad \text{et} \quad \nabla w_1(O) = \frac{1}{8} \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$$

Pour le nœud 2 :

$$\nabla w_2 = \frac{1}{8} \begin{pmatrix} 1-y-z+yz \\ -1-x+z+xz \\ -1-x+y+xy \end{pmatrix} \quad \text{d'où} \quad \nabla w_2(N_2) = \frac{1}{2} \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix} \quad \text{et} \quad \nabla w_2(O) = \frac{1}{8} \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}$$

De même pour les autres nœuds :

$$\begin{aligned} \nabla w_3(N_3) &= 1/2 \cdot (1; 1; -1) & \text{et} & \quad \nabla w_3(O) = 1/8 \cdot (1; 1; -1) \\ \nabla w_4(N_4) &= 1/2 \cdot (-1; 1; -1) & \text{et} & \quad \nabla w_4(O) = 1/8 \cdot (-1; 1; -1) \\ \nabla w_5(N_5) &= 1/2 \cdot (-1; -1; 1) & \text{et} & \quad \nabla w_5(O) = 1/8 \cdot (-1; -1; 1) \\ \nabla w_6(N_6) &= 1/2 \cdot (1; -1; 1) & \text{et} & \quad \nabla w_6(O) = 1/8 \cdot (1; -1; 1) \\ \nabla w_7(N_7) &= 1/2 \cdot (1; 1; 1) & \text{et} & \quad \nabla w_7(O) = 1/8 \cdot (1; 1; 1) \\ \nabla w_8(N_8) &= 1/2 \cdot (-1; 1; 1) & \text{et} & \quad \nabla w_8(O) = 1/8 \cdot (-1; 1; 1) \end{aligned}$$

Pour cet élément de référence, on peut remarquer que :

$$\nabla w_i(O) = \frac{1}{4} \cdot \nabla w_i(N_i)$$

On calculera donc $\nabla \varphi_T$ et $\vec{n}_i$ en les approximant par leurs valeurs au centre O de l'élément :

$$\begin{cases} \nabla \varphi_T = \sum_{i=1}^8 \nabla N_i(O) \cdot \varphi_i = \frac{1}{4} \cdot \sum_{i=1}^8 \nabla N_i(i) \cdot \varphi_i \\ \vec{n}_i = d \cdot |T| \cdot \nabla N_i(O) = \frac{1}{4} \cdot d \cdot |T| \cdot N_i(i) \end{cases}$$

D'autre part, on observe que pour tout élément hexaédrique dont les faces opposées sont parallèles, les gradients des fonctions de forme des nœuds opposés s'annulent deux à deux ($\nabla N_1 + \nabla N_7 = \nabla N_2 + \nabla N_8 = \nabla N_3 + \nabla N_5 = \nabla N_4 + \nabla N_6 = \vec{0}$) en tout point (x, y, z) de l'élément. La propriété $\sum_{i=1}^8 \nabla N_i(x, y, z) = \vec{0}$, équivalente à $\sum_{i=1}^8 \vec{n}_i = \vec{0}$, nécessaire au tri des contributions positives, est vérifiée en tout point de l'élément.

La méthode est donc applicable dans les cas où les faces opposées des éléments HEXA8 sont parallèles.

L'implémentation dans *Code_Aster* pour les éléments HEXA est quasiment inchangée par rapport à celle des éléments TETRA. On pondère uniquement les ∇N_i calculés aux nœuds par un coefficient de $1/4$, pour se ramener à leurs valeurs au centre de la maille.

Il sera effectué, lors du calcul des normales $\vec{n}_i$, une vérification de l'applicabilité de la méthode, en testant sur chaque élément :

$$\begin{cases} \left| \sum_{i=1}^8 \vec{n}_i \cdot \vec{x} \right| < \text{Tolérance} \\ \left| \sum_{i=1}^8 \vec{n}_i \cdot \vec{y} \right| < \text{Tolérance} \\ \left| \sum_{i=1}^8 \vec{n}_i \cdot \vec{z} \right| < \text{Tolérance} \end{cases}$$

La tolérance est fixée à une valeur de 10^{-6} .

Remarque :

Le critère est pour l'instant mesuré en absolu. Il pourrait éventuellement être remplacé par un critère relatif de la forme $\left| \sum_{i=1}^8 \left(\frac{\vec{n}_i \cdot \vec{x}_i}{\|\vec{n}_i\|} \right) \right|$.

4.3.4 Calcul sur les éléments de bord

Une condition de monotonie du schéma est que chaque nœud doit avoir au moins un de ses éléments périphériques qui participe à sa réactualisation, i.e. $(\forall \text{noeud } i) (\exists \text{élément } j | \alpha_i^j \neq 0)$. Or, pour certains éléments en bord de modèle et certaines orientations de ∇H , on peut avoir un nœud dont tous les α_i^j sont nuls (voir Figure 4.3.4-a).

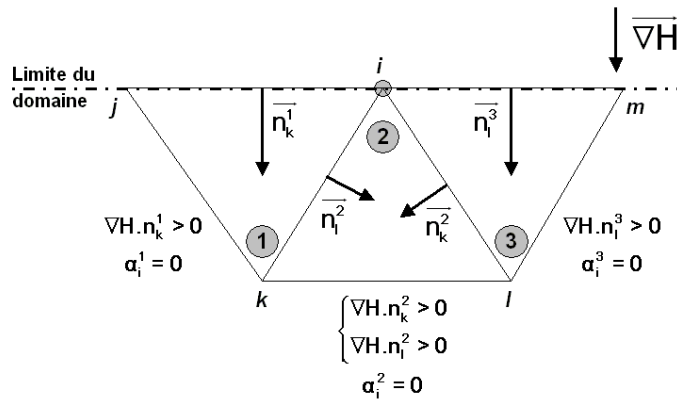


Figure 4.3.4-a : Point de bord singulier

Les level sets de ces nœuds vont alors être réactualisés, en utilisant l'évolution des nœuds qui les entourent (i.e. les nœuds appartenant aux mailles dont le nœud considéré est sommet). On va ainsi évaluer la nouvelle valeur de la level set à partir des nœuds voisins ayant été mis à jour, et du gradient sur ces nœuds (calculé avant la mise à jour).

Pour un nœud singulier i ayant un nœud voisin réactualisé j , l'approximation de la nouvelle valeur de la level set φ_i est :

$$\varphi_{i,n+1} \approx \varphi_{j,n+1} + \nabla \varphi_j^n \cdot \vec{j}i$$

Pour la phase de réinitialisation, on va normer le gradient, et estimer φ_i par $\frac{\nabla \varphi_j}{\|\nabla \varphi_j\|} \cdot \vec{j}i$, homogène à une distance physique (distance entre les points i et j dans la direction du gradient de φ).

On effectue alors une moyenne des valeurs de φ_i évaluées sur les nœuds voisins réactualisés.

Pour tout nœud i dont tous les α_i^j sont nuls :

$$\begin{aligned} \bullet \text{Réinitialisation} & : \quad \varphi_i^{n+1} = \underset{\substack{\text{noeuds } j \text{ voisins de } i \\ \text{réactualisés}}}{\text{moyenne}} \left(\varphi_j^{n+1} + \frac{\overrightarrow{\nabla} \varphi_j^n}{|\overrightarrow{\nabla} \varphi_j^n|} \cdot \vec{j}i \right) \\ \bullet \text{Réorthogonalisation} & : \quad \varphi_i^{n+1} = \underset{\substack{\text{noeuds } j \text{ voisins de } i \\ \text{réactualisés}}}{\text{moyenne}} \left(\varphi_j^{n+1} + \overrightarrow{\nabla} \varphi_j^n \cdot \vec{j}i \right) \end{aligned}$$

4.3.5 Pas de temps et critères de convergence

La condition de stabilité prépondérante étant le critère CFL, le pas de temps Δt est pris légèrement, inférieur à celui-ci. Le schéma étant cependant relativement stable, on prend un pas de temps voisin de la condition CFL afin de converger plus rapidement.

Ainsi : $\Delta t = 0,45 \cdot \Delta t_{CFL}$

Les résidus estimés sont :

$$\begin{aligned} \text{résidu local} &= \sum_{\substack{\text{noeuds } i \in \\ \text{tore d'estimation}}} \left(\frac{\varphi_j^{n+1} - \varphi_j^n}{\varphi_j^n} \right) \\ \text{résidu global} &= \sum_{\substack{\text{noeuds } i \in \\ \text{maillage}}} \left(\frac{\varphi_j^{n+1} - \varphi_j^n}{\varphi_j^n} \right) \end{aligned}$$

Les critères de tolérance sur les résidus locaux sont :

- Réinitialisation de l_{sn} : $\text{tolérance} = 10^{-8}$
- Réorthogonalisation de l_{st} : $\text{tolérance} = 10^{-3}$
- Réinitialisation de l_{st} : $\text{tolérance} = 10^{-5}$

En effet, on peut laisser converger la réinitialisation de l_{sn} , car elle ne contrarie pas la réorthogonalisation qui lui succède, contrairement à la réinitialisation de l_{st} , dont la tolérance est donc moins importante, afin de stopper cette dernière étape plus rapidement, et d'avoir un compromis acceptable entre orthogonalité et norme unitaire des gradients.

Bien que la condition de sortie de la boucle temporelle porte sur le résidu local (i.e. estimé dans un tore autour du fond de fissure), on estime aussi le résidu global sur tous les nœuds du maillage ; d'une part à des fins d'affichage, d'autre part, l'étude de la convergence du schéma montre que le laisser converger en phase de réorthogonalisation tend à perturber les résultats car il ne converge pas forcément vers un résidu strictement nul (voir Figure 4.3.5-a).

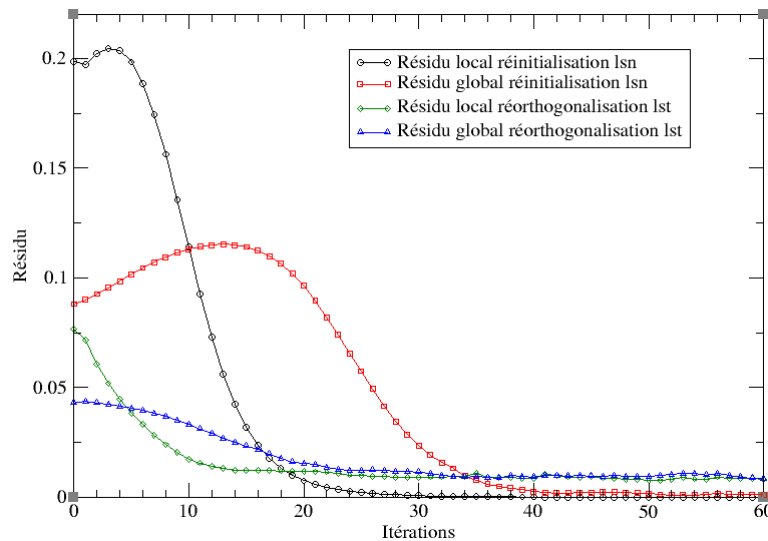


Figure 4.3.5-a : Allure des résidus des phases de réinitialisation et réorthogonalisation

On introduit alors un critère de sortie de la boucle portant sur la présence d'un minimum local du résidu global à partir de la 5^{ème} itération :

$$Si \left\{ \begin{array}{l} (résidu\ global)^{n+1} > (résidu\ global)^n \\ itération > 5 \end{array} \right\} \Rightarrow \text{Sortie de la boucle}$$

D'autre part, le nombre maximal d'itérations pour les 2 phases est fixé à 100.
Les critères de sortie de la boucle sont donc :

$$Si \left\{ \begin{array}{l} \text{Résidu local} < \text{Tolérance} \\ \text{ou Nombre maxi d'itérations atteint} \\ \text{ou Minimum local du résidu global} \end{array} \right\} \Rightarrow \text{Sortie de la boucle}$$

4.3.6 Méthode « Upwind »

La méthode dite « Upwind » est une méthode de différences finies centrées sélectives applicable exclusivement sur des maillages hexaédriques (ou rectangulaires en 2D) dont les arêtes sont bien orientées selon trois directions $(\mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3)$ formant une base de l'espace $\mathfrak{R}^3$, qui peut être utilisée pour résoudre des équations de la forme:

$$\frac{\partial \varphi}{\partial t} + \mathbf{V} \cdot \nabla \varphi = C$$

avec C une constante par rapport au temps. Ce type d'équation est communément utilisé dans le domaine fluide-dynamique. Pour cela, le vecteur $\mathbf{V}$ représente la vitesse avec laquelle la grandeur φ se propage. Les équations de réinitialisation et de réorthogonalisation des level sets peuvent être écrites sous la même forme et donc la méthode upwind est très intéressante pour ce problème aussi.

Pour l'application de la méthode des différences finies, on doit arranger tous les points de la grille utilisée pour effectuer l'intégration selon l'orientation des trois directions orthogonales $\mathbf{j}_1$, $\mathbf{j}_2$ et $\mathbf{j}_3$ de la base. Dans la description suivante, les symboles $\mathbf{x}_k$, $\mathbf{x}_{k,i+}$ et $\mathbf{x}_{k,i-}$ sont utilisés pour indiquer respectivement un point de la grille ($\mathbf{x}_k$), le point suivant ($\mathbf{x}_{k,i+}$) et le point précédent ($\mathbf{x}_{k,i-}$) selon la direction $\mathbf{j}_i$. Le vecteur $\mathbf{x}_k$ représente la position du point de la grille par rapport à la base utilisée. La distance (scalaire positive) entre ces points est donc écrite comme ceci: $(\mathbf{x}_{k,i+} - \mathbf{x}_k) \cdot \mathbf{j}_i$ et $(\mathbf{x}_k - \mathbf{x}_{k,i-}) \cdot \mathbf{j}_i$.

Pour la résolution avec la méthode upwind, on doit discrétiser l'équation par rapport au temps :

$$\frac{\varphi^{n+1}(\mathbf{x}_k) - \varphi^n(\mathbf{x}_k)}{\Delta t} + \mathbf{V}^n(\mathbf{x}_k) \cdot \nabla \varphi^n(\mathbf{x}_k) = C$$

avec φ^n valeur de φ à l'instant de temps n et $\mathbf{V}^n$ valeur du vecteur $\mathbf{V}$ au même instant. On peut ensuite calculer le produit scalaire selon les trois directions $\mathbf{j}_1$, $\mathbf{j}_2$ et $\mathbf{j}_3$:

$$\frac{\varphi^{n+1}(\mathbf{x}_k) - \varphi^n(\mathbf{x}_k)}{\Delta t} + \mathbf{V}_{j_1}^n(\mathbf{x}_k) \cdot \left(\frac{\partial \varphi}{\partial j_1} \right)_{\mathbf{x}_k}^n + \mathbf{V}_{j_2}^n(\mathbf{x}_k) \cdot \left(\frac{\partial \varphi}{\partial j_2} \right)_{\mathbf{x}_k}^n + \mathbf{V}_{j_3}^n(\mathbf{x}_k) \cdot \left(\frac{\partial \varphi}{\partial j_3} \right)_{\mathbf{x}_k}^n = C$$

Pour la discrétisation spatiale de chaque dérivée, on doit vérifier le signe de la composante de la vitesse:

$$\left(\frac{\partial \varphi}{\partial j_i} \right)_{\mathbf{x}_k}^n \simeq D_i \varphi^n(\mathbf{x}_k) \triangleq \begin{cases} \frac{\varphi^n(\mathbf{x}_k) - \varphi^n(\mathbf{x}_{k,i-})}{(\mathbf{x}_k - \mathbf{x}_{k,i-}) \cdot \mathbf{j}_i} & \text{si } \mathbf{V}_{j_i}^n(\mathbf{x}_k) > 0 \\ \frac{\varphi^n(\mathbf{x}_{k,i+}) - \varphi^n(\mathbf{x}_k)}{(\mathbf{x}_{k,i+} - \mathbf{x}_k) \cdot \mathbf{j}_i} & \text{si } \mathbf{V}_{j_i}^n(\mathbf{x}_k) < 0 \end{cases}$$

Enfin on peut donc écrire:

$$\varphi^{n+1}(\mathbf{x}_k) = \varphi^n(\mathbf{x}_k) - \Delta t \cdot \left(\sum_{i=1}^3 \mathbf{V}_{j_i}^n(\mathbf{x}_k) \cdot D_i \varphi^n(\mathbf{x}_k) - C \right)$$

Le calcul de la dérivée $D_i \varphi^n(\mathbf{x}_k)$ peut être problématique pour tous les points de la grille qui sont sur la frontière du domaine utilisé lors de la résolution de l'équation. En fait, pour ces points on ne connaît pas les valeurs de la fonction φ à l'extérieur du domaine et donc on ne peut pas calculer la valeur de $D_i \varphi^n(\mathbf{x}_k)$.

Prenons par exemple le domaine de la Figure 4.3.6-a et calculons $D_1 \varphi^n(\mathbf{x}_k)$. On envisage les deux situations suivantes. Si $\mathbf{V}_{j_1}^n(\mathbf{x}_k) > 0$, alors le point $\mathbf{x}_{k,1-}$ est utilisé pour le calcul. Ce point est dans le domaine et la valeur de $\varphi^n(\mathbf{x}_{k,1-})$ est connue. Si $\mathbf{V}_{j_1}^n(\mathbf{x}_k) < 0$, alors le point $\mathbf{x}_{k,1+}$ est utilisé pour le calcul. Ce point est à l'extérieur du domaine et donc la valeur de $\varphi^n(\mathbf{x}_{k,1+})$ n'est pas connue et le calcul de la dérivée n'est pas possible.

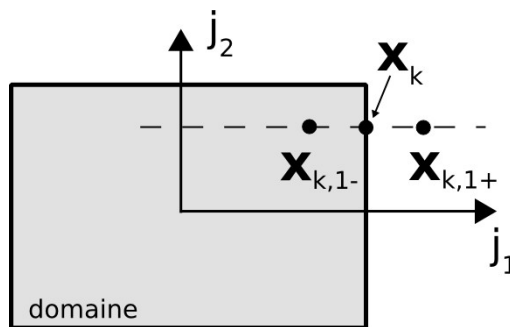


Figure 4.3.6-a : Calcul de la dérivée des level sets pour les points à la frontière du domaine

La solution la plus simple serait d'extrapoler la valeur inconnue, ou la dérivée liée à cette grandeur, à partir des valeurs aux points dans le domaine. Toutefois, comme décrit dans (bib69), cette solution ne donne pas des bons résultats et des oscillations peuvent apparaître au niveau des résultats de l'intégration. La solution proposée dans (bib69) est simple: il suffit de injecter les bonnes valeurs de la level set aux points problématiques du bord du domaine, c'est-à-dire que la nouvelle valeur de la level set est calculée à ces points en utilisant la définition de distance signée et pas par intégration. Le calcul de la distance signée sera détaillé dans le [§33].

4.3.7 Application de la méthode upwind aux équations de réinitialisation et de réorthogonalisation des level sets

L'équation de réinitialisation de la level set φ est la suivante [§14]):

$$\frac{\partial \varphi}{\partial t} = -\text{sign}(\varphi) \cdot (\|\nabla \varphi\| - 1)$$

On peut facilement mettre cette équation sous la forme suivante utilisée pour la description de la méthode upwind [§24]:

$$\frac{\partial \varphi}{\partial t} + V \cdot \nabla \varphi = C$$

En fait, on peut écrire l'équation de réinitialisation de la level set φ comme ceci:

$$\frac{\partial \varphi}{\partial t} + \text{sign}(\varphi) \frac{\nabla \varphi}{\|\nabla \varphi\|} \cdot \nabla \varphi = \text{sign}(\varphi)$$

On peut donc identifier les grandeurs suivantes:

$$V = \text{sign}(\varphi) \frac{\nabla \varphi}{\|\nabla \varphi\|}$$
$$C = \text{sign}(\varphi)$$

où φ est la level set normale ou tangente respectivement pour la réinitialisation de la level set normale et la réinitialisation de la level set tangente.

L'équation de réorthogonalisation de la level set φ (tangente) par rapport à la level set ψ (normale) est la suivante (voir paragraphe [§14]):

$$\frac{\partial \varphi}{\partial t} = -\text{sign}(\psi) \frac{\nabla \psi}{\|\nabla \psi\|} \cdot \nabla \varphi$$

On peut facilement mettre cette équation dans la forme suivante utilisée par la description de la méthode upwind [§24]:

$$\frac{\partial \varphi}{\partial t} + V \cdot \nabla \varphi = C$$

En fait, on peut écrire l'équation de réorthogonalisation comme ceci:

$$\frac{\partial \varphi}{\partial t} + \text{sign}(\psi) \frac{\nabla \psi}{\|\nabla \psi\|} \cdot \nabla \varphi = 0$$

On peut donc identifier les grandeurs suivantes:

$$V = \text{sign}(\psi) \frac{\nabla \psi}{\|\nabla \psi\|}$$
$$C = 0$$

On peut noter que l'expression du vecteur V pour l'équation de réinitialisation est la même que l'expression pour l'équation de réorthogonalisation. Donc la réinitialisation et la réorthogonalisation des level sets peuvent être calculées par la même routine.

4.3.8 Application de la méthode upwind à un maillage irrégulier

Pour l'application de la méthode des différences finies, on doit utiliser un maillage très régulier, constitué par des mailles hexaédriques en 3D et des mailles rectangulaires en 2D, dont les arêtes sont bien orientées selon les directions de la base de l'espace $\mathcal{R}^3$ en 3D et $\mathcal{R}^2$ en 2D. Cela représente

une très forte restriction pour la création du maillage et donc limite excessivement les cas qui peuvent être traités par la méthode.

Une solution à ce problème consiste à utiliser deux maillages différents (bib 69) : un maillage est utilisé pour la structure physique (*maillage physique*) et un maillage est utilisé pour la représentation des level sets (*maillage level sets*). Les restrictions de la méthode de différences finies s'appliquent uniquement à ce dernier maillage. Les valeurs des level sets sur le maillage physique peuvent être calculées par projection des valeurs connues sur le maillage level sets. Cela est nécessaire pour la localisation du fond de la fissure dans le maillage physique. En fait, les calculs de la mécanique de la rupture sont effectués uniquement sur ce dernier maillage. On peut donc réaliser la mise à jour, la réinitialisation et la réorthogonalisation des level sets en utilisant uniquement le maillage level sets. Après on peut projeter les level sets sur le maillage physique à partir du maillage level sets. Enfin on peut calculer les facteurs d'intensité des contraintes et la direction de propagation de la fissure en utilisant uniquement le maillage physique.

La projection permet d'obtenir aussi l'indépendance géométrique entre les deux maillages: le maillage des level sets n'a pas besoin d'être orienté selon les directions de la base choisie pour le maillage physique et il peut uniquement occuper le volume de la zone de propagation de la fissure dans la structure.

Les nœuds du maillage level set, construit comme décrit ci-dessus, constituent donc un nuage de points alignés selon trois directions orthogonales. La grille formée par ces points peut être utilisée pour l'application de la méthode «upwind». Dans la suite de ce document et dans la syntaxe du *Code_Aster*, on utilisera l'expression **grille auxiliaire** pour indiquer l'ensemble de ces points et pour souligner que, même si les points sont donnés par un maillage (maillage level-set), la méthode «upwind» n'utilise les nœuds de la grille auxiliaire que pour intégrer les équations de réinitialisation et réorthogonalisation des level sets. Également, pour simplifier les notations utilisées et les rendre plus claires, on utilisera tout simplement le terme **maillage** pour indiquer le maillage physique.

Remarque:

même si la grille auxiliaire a été introduite avec la méthode upwind, elle peut être aussi utilisée avec la méthode simplexe.

4.3.9 Implémentation dans *Code_Aster* de la projection entre la grille auxiliaire et le maillage physique

Dans *Code_Aster* la grille auxiliaire n'est pas donnée comme un ensemble de points bien orientés mais comme un maillage et donc quelques étapes intermédiaires sont nécessaires pour l'application de la méthode de différences finies upwind. En effet, pour cette méthode tous les points de la grille, pris coïncidents avec les nœuds du maillage level-set donné, doivent être arrangés selon les directions d'une base locale (voir paragraphe [§24]). On doit donc calculer cette base locale et après on doit ordonner les nœuds et calculer la distance entre deux nœuds successifs.

Le *calcul de la base locale* est très facile parce qu'on peut utiliser la définition des éléments du maillage de la grille auxiliaire. En effet, les arêtes des éléments sont bien orientées par rapport aux axes d'une base locale et on peut définir cette base en prenant des arêtes qui sont orthogonales entre elles. En plus on ne recherche que la direction des axes de la base locale. On peut donc utiliser les arêtes de n'importe quel élément du maillage. En 3D, le premier élément du maillage de la grille auxiliaire permet de déterminer à partir de l'un de ses nœuds, les trois arêtes qui y aboutissent, et ainsi de calculer les trois vecteurs unitaires selon les trois directions des arêtes sélectionnées. En 2D on fait la même opération avec deux arêtes.

Pour le *classement des nœuds* selon les directions des axes de la base locale calculée, on peut utiliser encore une fois les arêtes des éléments. Pour chaque nœud du maillage, on peut sélectionner les éléments qui le contiennent. Pour chacun des éléments sélectionnés, on sélectionne les arêtes qui partent du nœud. Enfin pour chacune des arêtes sélectionnées, on ordonne les points selon une des directions de la base locale du maillage en utilisant le produit scalaire entre l'arête et les vecteurs unitaires de la base locale et on peut calculer la distance entre les nœuds de l'arête.

En 3D l'algorithme mis en place est le suivant (tous les vecteurs sont représentés dans la base choisie pour représenter les nœuds du maillage physique):

- On repère les trois vecteurs unitaires de la base locale: $\vec{j}_1$, $\vec{j}_2$ et $\vec{j}_3$
- Boucle sur les nœuds N du maillage
- On initialise à zéro les valeurs stockées des six nœuds qui sont connectés (successifs ou précédents) au nœud N selon les trois vecteurs unitaires de la base locale
- On repère les éléments qui incluent le nœud N dans leur définition
- Boucle sur les éléments sélectionnés
- On repère les trois arêtes qui partent du nœud N
- Boucle sur les trois arêtes sélectionnées
- On calcule le vecteur $\vec{NA}$ entre les deux nœuds d'extrémité de l'arête. Ce vecteur est orienté selon la direction qui va du nœud N à l'autre nœud A extrémité de l'arête.
- On calcule les composantes du vecteur $\vec{NA}$ selon la base locale:
$$NA_1 = \vec{NA} \cdot \vec{j}_1$$
$$NA_2 = \vec{NA} \cdot \vec{j}_2$$
$$NA_3 = \vec{NA} \cdot \vec{j}_3$$
- On calcule la tolérance qui est utilisée pour vérifier si une composante est nulle. Cette valeur est calculée comme un pourcentage p (fixé à 1% dans Code_Aster) de la longueur du vecteur:
$$l_{max} = p \cdot |\vec{NA}|$$
- On repère la composante n plus grande:
$$\Delta = \max(|NA_1|, |NA_2|, |NA_3|)$$
$$n \in [1, 2, 3] / \Delta = |NA_n|$$
- On vérifie que les deux autres composantes sont nulles ($< l_{max}$), sinon on s'arrête avec un message d'erreur. En fait, cela signifie que les arêtes du maillage ne sont pas toutes alignées selon les trois mêmes directions.
- Si $NA_n > 0$, alors on stocke le nœud A comme le nœud **successif** du nœud N selon la direction $\vec{j}_n$. Si $NA_n < 0$, alors on stocke le nœud A comme le nœud **précédent** le nœud N selon la direction $\vec{j}_n$. On stocke aussi Δ comme la valeur de la distance entre les nœuds A et N selon la direction $\vec{j}_n$. Si $NA_n = 0$, alors on s'arrête avec un message d'erreur parce que cela signifie que plusieurs nœuds de l'élément sont coïncidents.
- Fin de la boucle
- Fin de la boucle
- Fin de la boucle

En 2D on peut utiliser le même algorithme en prenant seulement deux arêtes par nœud et donc deux composantes. Il est très important qu'avant le début de la première boucle on initialise à zéro les valeurs stockées aux six nœuds qui sont connectés au nœud N . En effet, si le numéro stocké d'un de ces nœuds est égal à zéro, on peut en déduire que le nœud N n'a pas un nœud précédent ou successif selon une direction bien définie. Cela est essentiel pour appliquer correctement les conditions aux limites pour la méthode upwind (voir paragraphe [§24]). L'arrangement des nœuds selon les directions des axes de la base locale de la grille doit être effectué une seule fois avant l'utilisation de la méthode upwind.

Dans la description ci-dessus, on n'a rien dit sur les modifications que l'on doit apporter au code avant les phases de réinitialisation et de réorthogonalisation des level sets. En fait, il n'y a pas de grands changements à faire. Le calcul des facteurs d'intensité des contraintes et de la vitesse d'avancée du fond de fissure ne sont pas affectés [§4]. Les algorithmes utilisés pour les phases d'extension du champ de vitesse au maillage grille [§5] et l'ajustement de la vitesse normale [§9] ne changent pas mais on doit donner en entrée aux sous-routines correspondantes les nœuds de la grille auxiliaire et pas ceux du maillage physique. Pour cela, les changements du code à faire peuvent être réalisés très rapidement parce que les valeurs de la vitesse d'avancée du fond de fissure sont indépendantes du maillage physique. En fait, le calcul de G , K_I et K_{II} est conduit sur le maillage physique mais les valeurs de ces grandeurs sont liées aux points du fond de fissure qui sont stockés de façon

indépendante par rapport au maillage utilisé. Cela permet d'effectuer l'extension du champ de vitesse et l'ajustement de la vitesse normale directement sur la grille auxiliaire, par projection sur le fond de fissure discrétisé par segments, aucune projection de champ de vitesse entre le maillage physique et la grille n'étant nécessaire car ils utilisent tous les deux la même discrétisation du fond de fissure. Enfin, pour la phase de mise à jour, rien ne change dans les sous-routines sauf l'information sur le maillage utilisé (physique ou level set): les données sont les mêmes et sont stockées aux nœuds du maillage.

En présence de maillages constitués d'un grand nombre d'éléments, la projection entre la grille auxiliaire et le maillage physique peut prendre beaucoup de temps de calcul. Si on regarde l'utilisation des level sets pour le maillage physique, on peut dire qu'il n'est pas nécessaire qu'on connaisse la valeur des level sets pour tous les nœuds de ce maillage. En effet, dans le maillage physique, on utilise les level set pour trois calculs:

1. la représentation du fond de la fissure,
2. le calcul de la base locale en chaque point du fond de la fissure,
3. le calcul énergétique des facteurs d'intensité des contraintes.

Pour le premier et le deuxième point, il suffit que la valeur des level sets soit connue pour les nœuds des éléments coupés par le fond de fissure. Pour le dernier point il suffit que la valeur des level sets soit connue pour tous les nœuds qui sont compris dans le volume du maillage utilisé pour le calcul énergétique. Ce volume coïncide avec un tore construit autour du fond de fissure. Le rayon de ce tore est spécifié par l'utilisateur dans l'opérateur énergétique utilisé pour le calcul, par exemple `CALC_G`.

Les trois calculs ci-dessus définissent trois groupes de nœuds pour lesquels la valeur des level sets doit être connue. Le groupe défini pour le calcul énergétique (point 3) inclut les autres. Il suffit donc qu'on sélectionne dans chaque maillage (physique et grille auxiliaire) les nœuds qui sont compris dans le tore décrit ci-dessus et qu'on fasse la projection uniquement entre les deux groupes définis par ces nœuds.

On peut sélectionner les nœuds à projeter si la distance entre chaque nœud du maillage et le fond de fissure est connue. Pour la grille auxiliaire cette distance est déjà calculée par la routine qui étend aux nœuds du maillage le champ de vitesse connu en fond de fissure [§5]. Dans ce cas, la sélection des nœuds à projeter est donc rapide. Par contre, pour le maillage physique cette distance n'est pas connue et elle doit être calculée en utilisant le même algorithme que celui décrit au [§5].

La projection peut être mise en œuvre en utilisant l'opérateur `PROJ_CHAMP` disponible dans *Code_Aster*, ou mieux en utilisant les routines fortran associées. En effet l'opérateur ne peut pas être utilisé dans les routines fortran de `PROPA_FISS` car il demande au superviseur la valeur d'opérandes qui n'existent pas pour l'opérateur `PROPA_FISS`. Par exemple, `PROJ_CHAMP` attend que les deux modèles sur lesquels la projection est faite soient passés en utilisant les mots-clé `MODELE_1` et `MODELE_2`, qui n'existent pas pour `PROPA_FISS`. On doit donc appeler directement les routines utilisées par `PROJ_CHAMP` (`pj2dco.f`, `pj3dco.f` et `cnsprj.f`). Les domaines sur lesquels la projection doit être faite sont donnés à ces routines sous la forme suivante:

- le domaine du champ à projeter (grille auxiliaire) doit être donné comme groupe d'éléments,
- le domaine sur lequel le champ est projeté (maillage physique) doit être donné comme groupe de nœuds.

Comme décrit ci-dessus, on a déjà sélectionné les nœuds du maillage physique pour lesquels la valeur des level sets doit être connue. La sélection est faite en prenant tous les nœuds qui sont compris dans le tore utilisé pour les calculs énergétiques (point 3 ci-dessus). Cela permet d'avoir immédiatement le domaine sur lequel le champ est projeté avec la forme demandée par les routines de `PROJ_CHAMP`. Par contre, un travail supplémentaire est nécessaire pour définir le domaine du champ à projeter. On peut sélectionner les nœuds de ce domaine (sur le maillage level sets) en utilisant le tore déjà utilisé pour définir l'autre domaine. Ensuite on peut définir le domaine du champ à projeter en cherchant les éléments qui ont au moins un des nœuds sélectionnés avec la technique précédente.

Le dernier problème à résoudre est le suivant: quelle valeur doit-on prendre pour le rayon du tore utilisé pour la sélection des nœuds à projeter ? On a déjà dit que la valeur utilisée dans l'opérateur du calcul énergétique (point 3 ci-dessus) est celle que l'on doit utiliser. Malheureusement cette valeur n'est pas connue de l'opérateur `PROPA_FISS`. On peut donc utiliser la valeur du paramètre `RAYON` de cet opérateur qui définit le tore autour du fond de la fissure utilisé pour le calcul du résidu local [§31]. Il doit être toujours plus grand que la valeur du rayon utilisé dans l'opérateur du calcul énergétique parce que, après la propagation, les propriétés d'orthogonalité et de distance signée des level sets sont

garanties uniquement dans ce tore. En effet, la convergence de la méthode upwind est vérifiée uniquement dans ce tore. Toutefois la distance à utiliser pour la sélection des nœuds doit être calculée entre chaque nœud et la nouvelle position du fond de la fissure après la propagation. Cette position est définie par les nouvelles valeurs des level sets après la propagation qui sont connues uniquement pour la grille auxiliaire. La nouvelle position du fond sera connue pour le maillage physique après la projection. La sélection des nœuds intéressés par la projection doit donc forcément être mise en œuvre à partir de la position du fond de la fissure avant la propagation. Le tore à utiliser doit donc inclure toutes les possibles nouvelles positions du fond de fissure et tous les nœuds du tore utilisé pour le calcul du résidu local.

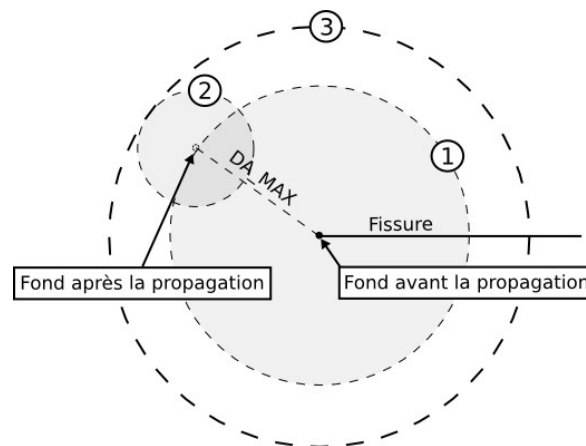


Figure 4.3.9-a: sélection des nœuds à projeter

On peut regarder l'exemple donné figure 4.3.9-a pour une fissure en 2D. L'avancement maximal de la fissure est choisi par l'utilisateur en utilisant le paramètre `DA_MAX` de l'opérateur `PROPA_FISS`. La position du fond de la fissure après la propagation est donc comprise dans le cercle de centre coïncidant avec la position du fond avant la propagation et de rayon égal à la valeur de `DA_MAX` (cercle 1 de la figure). Les nœuds utilisés pour le calcul du résidu local sont compris dans le cercle de centre coïncidant avec la position du fond après la propagation et de rayon égal à la valeur du paramètre `RAYON` de l'opérateur `PROPA_FISS` (cercle 2 de la figure). Pour la projection on peut donc sélectionner tous les nœuds compris dans le cercle de centre coïncidant avec la position du fond avant la propagation et de rayon égal à l'addition du `DA_MAX` et `RAYON` (cercle 3 de la figure).

Il faut observer enfin que la projection proposée ci-dessus calcule la nouvelle valeur des level sets uniquement pour les nœuds qui ont été projetés. Toutefois, pour le bon fonctionnement des autres routines X-FEM de *Code_Aster*, il est nécessaire que les level sets du maillage physique soient définies pour tous les nœuds. Pour cela on peut simplement préserver la valeur des level sets pour tous les nœuds qui ne sont pas affectés par la projection. À chaque fois, on peut donc prendre les level sets du maillage physique avant la propagation et on peut changer la valeur des level sets uniquement pour les nœuds affectés par la projection. Cette solution est très simple à développer dans le code et est très rapide. En plus elle garantit que la surface existante de la fissure et le fond de la fissure sont toujours correctement représentés par les level sets parce que les résultats de chaque projection sont préservés en arrière du fond de fissure. Un exemple est donné figure 4.3.9-b. À gauche de la figure on voit la configuration des level sets avant la mise à jour. La surface de la fissure est représentée en orange et le fond est marqué avec un petit orange. La configuration des level sets après la propagation (et la projection) est visible à droite. Seulement les level sets aux nœuds dans le cercle 3 ont été remplacées par les nouvelles valeurs calculées sur la grille auxiliaire. La nouvelle surface de la fissure est représentée en rouge et la nouvelle position du fond est représentée par un petit rouge. La position du fond avant la propagation est toujours marquée par un point orange. On peut voir que la surface de la fissure qui existait avant la propagation est préservée par la projection (ligne rouge à droite du point orange), ainsi que la continuité de la surface de la fissure entre le domaine de projection (cercle 3) et la partie restante du maillage physique (la ligne rouge est la continuation de la ligne orange). Avec définition du domaine de projection utilisée, les level sets autour de la position du fond de fissure avant la propagation (point orange) sont toujours affectées par la projection et il n'y a donc pas de risques que de faux fonds de fissure soient détectés ($lsn = lst = 0$) après la projection. On peut aussi noter que les level sets sont discontinues pour le maillage physique mais que cela n'est pas important. En

effet, les propriétés d'orthogonalité et de distance signée des level sets sont garanties pour tous les nœuds impliqués dans les calculs sur le maillage physique.

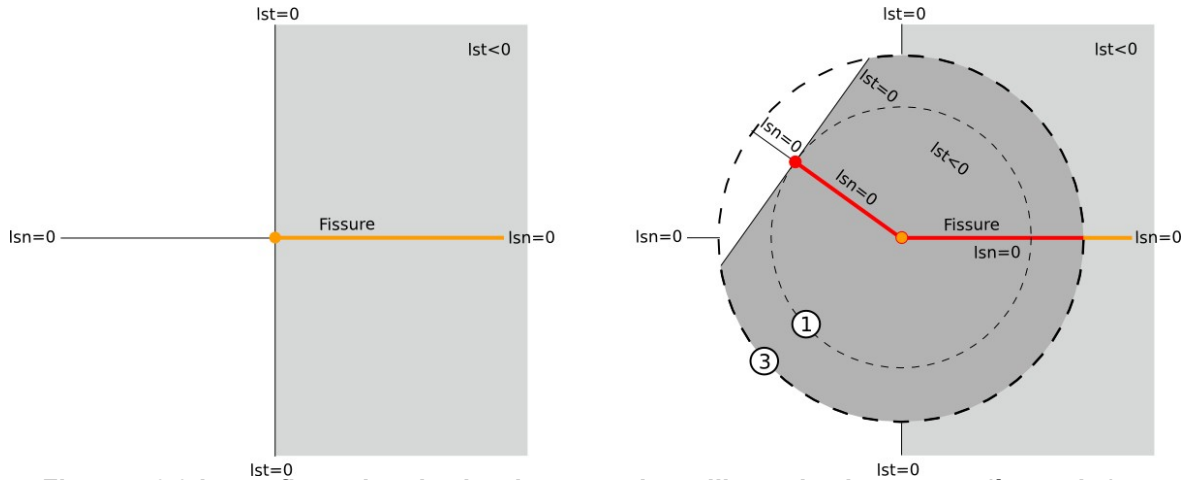


Figure 4.3.9-b: configuration des level sets sur le maillage physique avant (à gauche) et après (à droite) la propagation. La zone où la level set tangente est négative est tintée en gris. La surface et le fond de la fissure sont représentés en orange avant la propagation et en rouge après.

Il faut remarquer que les level sets sont mises à jour sur tous les nœuds de la grille auxiliaire et que pour ce maillage leur représentation est correcte partout. Donc pour la grille auxiliaire on n'a pas les discontinuités qu'on trouve sur le maillage physique après la projection (figure 4.3.9-b), sauf dans le cas d'utilisation de la localisation du domaine de calcul décrite au paragraphe [§ 46], ce qui ne représente pas un problème.

4.3.10 Implémentation de la méthode upwind dans Code_Aster

Après classement des nœuds selon les directions des axes de la base locale du maillage, on peut utiliser la méthode upwind pour résoudre les phases de réinitialisation et réorthogonalisation des level sets. Les équations à utiliser sont les suivantes [§24]:

$$\varphi^{n+1}(\mathbf{x}_k) = \varphi^n(\mathbf{x}_k) - \Delta t \cdot \left(\sum_{i=1}^3 V_{j_i}^n(\mathbf{x}_k) \cdot D_i \varphi^n(\mathbf{x}_k) - C \right)$$

$$D_i \varphi^n(\mathbf{x}_k) \triangleq \begin{cases} \frac{\varphi^n(\mathbf{x}_k) - \varphi^n(\mathbf{x}_{k,i-})}{(\mathbf{x}_k - \mathbf{x}_{k,i-}) \cdot \mathbf{j}_i} & \text{si } V_{j_i}^n(\mathbf{x}_k) > 0 \\ \frac{\varphi^n(\mathbf{x}_{k,i+}) - \varphi^n(\mathbf{x}_k)}{(\mathbf{x}_{k,i+} - \mathbf{x}_k) \cdot \mathbf{j}_i} & \text{si } V_{j_i}^n(\mathbf{x}_k) < 0 \end{cases}$$

Pour l'utilisation de ces équations, on se ramène dans la base locale du maillage ($\mathbf{j}_1$, $\mathbf{j}_2$ et $\mathbf{j}_3$). On doit faire attention au calcul des composantes du vecteur $\mathbf{V}$. En effet, les expressions de ce vecteur sont les suivantes (voir paragraphe [§26]):

- $\mathbf{V} = \text{sign}(\varphi) \frac{\nabla \varphi}{\|\nabla \varphi\|}$, pour la phase de réinitialisation de la level set φ ,
- $\mathbf{V} = \text{sign}(\psi) \frac{\nabla \psi}{\|\nabla \psi\|}$, pour la phase de réorthogonalisation des level sets.

La direction du vecteur $\mathbf{V}$ est donc coïncidente avec la direction de $\nabla \varphi$ ou $\nabla \psi$. Dans Code_Aster le calcul du gradient d'un champ scalaire est effectué par une sous-routine externe à la partie X-FEM du code. Pour le calcul, cette sub-routine utilise les valeurs du champ scalaire et les coordonnées des nœuds, qui sont données dans la base correspondant à celle du maillage physique. Le vecteur $\mathbf{V}$ est donc représenté dans cette dernière base. A contrario les équations de la méthode

upwind ci-dessus doivent être forcément exprimées dans la base locale de la grille auxillaire. On doit donc d'abord exprimer le vecteur V dans la base locale de la grille auxillaire:

$$V_{j_i} = V \cdot j_i \quad i \in N, i \in [1,3]$$

La condition de stabilité pour l'intégration des équations est la condition CFL déjà explicitée [§14]. Pour n'avoir pas de problèmes de convergence, la valeur du pas de temps choisie dans *Code_Aster* est la suivante:

$$\Delta \tau = 0.45 \cdot \Delta \tau_{CFL}$$

Le résidu local est utilisé comme critère de convergence. Il est calculé par tous les nœuds qui sont dans un tore qui s'étend autour du fond de la fissure. Le rayon de ce tore est donné par l'utilisateur. À chaque itération, le résidu local R est calculé comme ceci:

$$R = \sqrt{\frac{\sum_{x_k \in \text{tore}} \left(\varphi^{n+1}(x_k) - \varphi^n(x_k) \right)^2}{\sum_{x_k \in \text{tore}} \left(\varphi^n(x_k) \right)^2}}$$

On s'arrête si le résidu local est inférieur à une tolérance ou si le nombre d'itérations est supérieur à une valeur maximale. Dans *Code_Aster*, les valeurs suivantes sont utilisées:

Tolérance sur le résidu local = 10^{-7}

Nombre maximum de itérations = 300

L'algorithme mis en place est donc le suivant:

- Boucle sur le temps virtuel τ
- Boucle sur les nœuds $\mathbf{x}_k$ du maillage level sets
- On calcule le vecteur $\mathbf{V}^n$ dans la base du maillage physique:

$$\mathbf{V}^n(\mathbf{x}_k) = \text{sign}(\varphi^n(\mathbf{x}_k)) \frac{\nabla \varphi^n(\mathbf{x}_k)}{\|\nabla \varphi^n(\mathbf{x}_k)\|} \quad (\text{phase de réinitialisation})$$

$$\mathbf{V}^n(\mathbf{x}_k) = \text{sign}(\psi^n(\mathbf{x}_k)) \frac{\nabla \psi^n(\mathbf{x}_k)}{\|\nabla \psi^n(\mathbf{x}_k)\|} \quad (\text{phase de réorthogonalisation})$$

- On calcule les trois composantes du vecteur $\mathbf{V}^n$ dans la base de la grille auxiliaire:

$$V_{j_1}^n = \mathbf{V}^n(\mathbf{x}_k) \cdot \mathbf{j}_1$$

$$V_{j_2}^n = \mathbf{V}^n(\mathbf{x}_k) \cdot \mathbf{j}_2$$

$$V_{j_3}^n = \mathbf{V}^n(\mathbf{x}_k) \cdot \mathbf{j}_3$$

- On calcule les trois composantes $D_i \varphi^n(\mathbf{x}_k)$ de la dérivée de la level set en utilisant les informations données par la phase où les nœuds sont ordonnés selon les directions des axes de la base locale du maillage. Si le nœud n'a aucun nœud précédent ou successif selon la direction i voulue, on l'exclut des itérations upwind et la nouvelle valeur de la level set est calculée directement comme décrit au [§ 33].
- On calcule et on stocke provisoirement la valeur du produit scalaire entre le vecteur $\mathbf{V}^n$ et le gradient de la level set:

$$\varphi_{tmp}(\mathbf{x}_k) = \sum_{i=1}^3 V_{j_i}^n(\mathbf{x}_k) \cdot D_i \varphi^n(\mathbf{x}_k)$$

- Fin de la boucle
- On initialise à zéro les variables utilisées pour le calcul de l'erreur: $S = L = 0$
- Boucle sur les nœuds $\mathbf{x}_k$ de la grille auxiliaire (les nœuds qui ont été exclus des itérations upwind à la boucle précédente ne sont pas considérés dans cette boucle)
- On calcule la différence entre $\varphi^{n+1}(\mathbf{x}_k)$ et $\varphi^n(\mathbf{x}_k)$:

$$\Delta \varphi = -\Delta \tau \cdot (\varphi_{tmp}(\mathbf{x}_k) - C)$$

- Si le nœud est dans le volume d'estimation de l'erreur locale, on l'utilise pour la mise à jour des variables d'erreur:

$$S = S + (\Delta \varphi)^2$$

$$L = L + (\varphi^n(\mathbf{x}_k))^2$$

- On calcule la nouvelle valeur de la level set:

$$\varphi^{n+1}(\mathbf{x}_k) = \varphi^n(\mathbf{x}_k) + \Delta \varphi$$

- Fin de la boucle

- On calcule le résidu local: $R = \sqrt{\frac{S}{L}}$

- Si ($R \leq \text{tolérance}$) ou ($\text{iterations} \geq \text{nombre max d'itération}$), on sort de la boucle
- Fin de la boucle

4.3.11 Conditions aux limites pour la méthode upwind

Comme déjà décrit au [§24], pendant l'intégration des équations de réinitialisation et de réorthogonalisation des level sets, la méthode upwind n'arrive généralement pas à calculer correctement la nouvelle valeur de la level set à l'itération courante pour des points sur la frontière du domaine de calcul. Pour résoudre ce problème, des conditions aux limites doivent être appliquées sur tous les points problématiques de la frontière. Afin de garantir la robustesse de la méthode upwind, la solution proposée dans (bib69) a été implémentée dans *Code_Aster*. Quand un point problématique est détecté pendant les itérations d'intégration, il est exclu du processus itératif d'intégration et la nouvelle valeur de la level set est calculée directement en utilisant la propriété distance signée. Même si le

calcul de la distance signée peut sembler simple à implémenter, il nécessite le développement d'algorithmes dédiés pour éviter l'augmentation excessive des temps de calcul.

La méthode upwind est utilisée trois fois, comme décrit au [§26] : pour la réinitialisation de la level set normale, pour la réorthogonalisation de la level set tangente par rapport à la level set normale, pour la réinitialisation de la level set tangente. Suite à chacune de ces réinitialisations, la surface iso-zéro de la level set est préservée (bib69), c'est-à-dire que la valeur de la level set aux nœuds des éléments de la grille coupés par la surface iso-zéro n'est pas changée. En effet ces points sont exclus des itérations de la méthode upwind, la phase de mise à jour des level set donnant déjà les bonnes valeurs pour la localisation de la surface iso-zéro ([§44]). Si on considère un point du domaine, la valeur réinitialisée de la level set est la distance normale entre le point et la surface iso-zéro de la level set à réinitialiser. La condition aux limites à imposer au point coïncide donc avec cette distance.

Dans le cas de la réorthogonalisation de la level set tangente par rapport à la level set normale, c'est l'iso-zéro de cette dernière qui est préservée pendant toute la phase de réorthogonalisation. Les valeurs de la level set tangente sont inchangées pour tous les points de l'iso-zéro de la level set normale. La réorthogonalisation propage dans le domaine de calcul les valeurs en ces points orthogonalement à l'iso-zéro de la level set normale (bib69). Cela signifie que si l'on prend un point P du domaine, la valeur réorthogonalisée de la level set tangente coïncide avec la valeur de cette level set au point iso-zéro de la level set normale qui est la projection du point P sur cette iso-zéro.

On peut donc conclure que, indépendamment de la phase (réinitialisation ou réorthogonalisation) et de la level set considérée, l'imposition des conditions aux limites se réduit aux opérations suivantes:

OP1 . projection du point problématique sur l'iso-zéro de la level set à réinitialiser, dans le cas de la réinitialisation, ou sur la level set normale, dans le cas de la réorthogonalisation,

OP2 . dans le cas de la réinitialisation, calcul de la distance entre le point problématique et son projeté. Dans le cas de la réorthogonalisation, calcul de la level set tangente au point projeté.

L'implémentation de ces deux phases dans le code dépend fortement de la façon dont la surface iso-zéro de chaque level set est stockée. Comme décrit au [§44], on peut calculer les points d'intersection entre l'iso-zéro et chaque arête de chaque élément de la grille auxiliaire (figure 4.3.11-a).

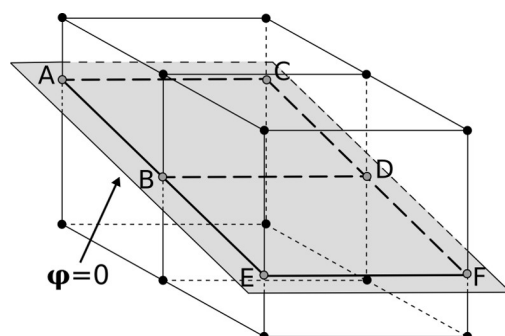


Figure 4.3.11-a: exemple de calcul des points d'intersection (du A au F) entre la surface iso-zéro de la level set φ et deux éléments de la grille auxiliaire.

En conséquence, l'iso-zéro de la level set est stockée comme un groupe de points pour chaque élément coupé. Ces points doivent être utilisés pour reconstituer l'iso-zéro dans l'élément. La façon la plus simple de faire est de créer des triangles à partir de ces points. Toutefois le choix des triangles à former n'est pas unique et plusieurs triangulations peuvent être formées à partir des mêmes points (figure 4.3.11-b). Dans le cas où l'iso-zéro n'est pas plane, la qualité de la surface reconstituée dépend de ce choix. A défaut d'un critère pour dire quelle représentation est la meilleure parmi celles qui sont possibles, on décide de retenir tous les triangles qu'on peut former avec chaque groupe de points: attention, la surface n'est pas reconstituée du point de vue géométrique, mais elle est remplacée par un ensemble de triangles qui peuvent se recouvrir partiellement. Cela permet aussi d'éliminer le besoin d'avoir un algorithme évitant la création de triangulations incorrectes du point de vue géométrique, ce qui est possible dans tous les cas où le groupe de points utilisés pour la triangulation est formé de plus de quatre points.

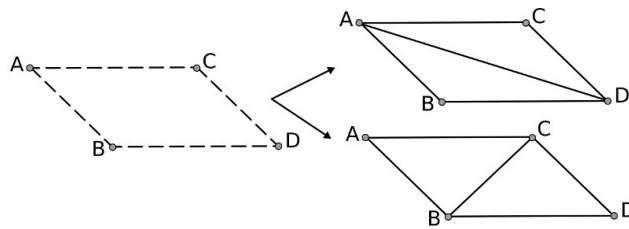


Figure 4.3.11-b: on considère un des deux éléments de la figure 4.4-a. L'intersection entre la surface iso-zéro de la level set et les arêtes de l'élément est constituée par les 4 points A , B , C et D . À droite on voit les triangulations qui peuvent être utilisées pour reconstituer la surface.

On peut en conclure que le calcul de la projection d'un point problématique sur l'iso-zéro de la level set (OP1) se réduit à la projection d'un point sur un triangle. Cela peut être réalisé en suivant ce qui est décrit au [§44] et qui sera utilisé pour le calcul des level sets à proximité de l'iso-zéro. En ce qui concerne la deuxième opération (OP2) le calcul de la distance entre un point problématique et son projeté est immédiat dans le cas de la réinitialisation. Pour la réorthogonalisation, le calcul de la level set au point projeté sera détaillé au paragraphe [§44].

L'algorithme de projection d'un point sur un triangle détaillé au paragraphe [§44] est efficace pour le calcul des level set à proximité de la surface iso-zéro, où les nœuds à projeter sont toujours contenus dans l'élément coupé par l'iso-zéro, mais il peut donner de mauvais résultats dans certains cas où le point à projeter est loin de l'iso-zéro. Ces cas sont traités dans la partie qui suit. Il faut remarquer que la sélection du triangle le plus proche du point à projeter suit toujours l'algorithme détaillé au paragraphe [§44]. Les problèmes traités dans la partie suivante concernent seulement le choix du bon point projeté et non pas celui du bon triangle.

La méthode de projection considère chaque triangle de l'iso-zéro de la level set et calcule la projection pour chacun. Le point problématique est projeté suivant la normale au plan du triangle. Si le point projeté est en dehors du triangle, on le corrige par une deuxième projection qui le ramène sur l'arête la plus proche du triangle (figure 5-a). À la fin du processus, on retient le triangle pour lequel la distance entre le point problématique et le point projeté, après la correction éventuelle, est minimale. Le triangle ainsi sélectionné est toujours le bon triangle. On peut avoir des problèmes de sélection au niveau du point projeté sélectionné, c'est-à-dire que dans certains cas le point projeté avant correction est celui à retenir et non pas celui après correction. En effet, dans le cas où le triangle retenu ne contient pas la projection du point problématique et où le point projeté, avant correction, est en dehors du volume de la structure, cette méthode de projection donne de mauvais résultats. Un exemple est visible en figure 4.3.11-c. Le point P est sur la surface libre de gauche de la structure. Le triangle retenu par la méthode de projection est celui coloré en gris. La projection du point P sur le plan du triangle coïncide avec le point P_1 . Étant en dehors du triangle, le point P_1 est ramené sur l'arête du triangle la plus proche par projection et le point P_2 obtenu est sélectionné comme projeté du point P . La distance du point P à l'iso-zéro de la level set est donc calculée comme la distance entre P et P_2 . Cependant c'est évident que la bonne distance est celle entre P et P_1 .

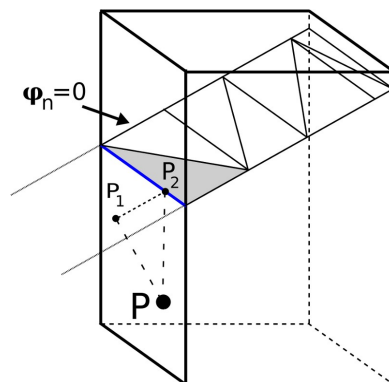


Figure 4.3.11-c: exemple de cas où la méthode de projection décrite au [§44] donne de mauvais résultats.

Dans le cas où le projeté est en dehors du triangle mais à l'intérieur du volume de la structure, la correction apportée par la méthode de projection est correcte et donne de bons résultats. Un exemple est donné figure 4.3.11-d . Si on considère les triangles rouge et vert, on voit que les points P_1 et P_2 , projetés du point problématique M sur le plan de chaque triangle, sont tous les deux en dehors des triangles. La méthode de projection ramène donc chaque point sur l'arête la plus proche des triangles et le point P_3 est obtenu. La distance entre M et P_3 est effectivement la bonne distance entre le point problématique et la surface iso-zéro.

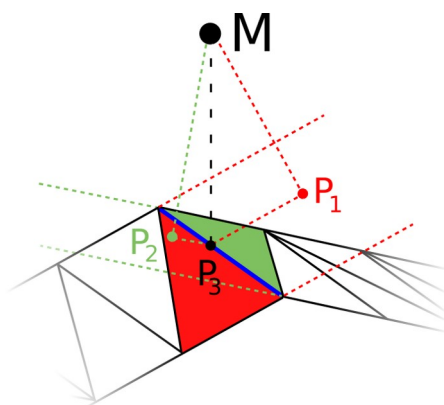


Figure 4.3.11-d: exemple de projection d'un point problématique en présence d'un angle de bifurcation important de la fissure. Du fait de cet angle, l'iso-zéro de la level set présente elle-même un angle. On arrive à la même situation suite à la triangulation géométrique de la surface iso-zéro, avec une grille auxiliaire pas trop raffinée ou pour une iso-zéro peu régulière du point de vue géométrique.

Pour éviter des erreurs, après chaque projection du point problématique M sur un des triangles de l'iso-zéro, il faut vérifier si le point projeté (point P_1 de la figure 4.3.11-c , points P_1 ou P_2 de la figure 4.3.11-d) est en dehors du triangle avant la correction apportée par la méthode de projection. Dans ce cas, si le point projeté est aussi en dehors du volume de la structure (cas de la figure 4.3.11-c), on le sélectionne comme projeté du point M , sinon on sélectionne le point obtenu après la correction (point P_3 en figure 4.3.11-d).

Pour distinguer ces deux cas, il suffit de vérifier si l'arête sur laquelle le projeté est ramené par la correction est partagée par deux triangles. Sur la figure 4.3.11-c , l'arête (en bleu) appartient seulement au triangle gris, et donc elle est sur la surface libre de la structure. Dans ce cas, le projeté P_1 avant correction est en dehors du volume de la structure. Sur la figure 4.3.11-d , la même arête, toujours en bleu, appartenant aux triangles rouge et vert, n'est donc pas sur la surface libre de la structure. Dans ce cas, le projeté P_1 ou P_2 avant correction est dans le volume de la structure.

Pour appliquer cette solution, les triangles qui forment la surface iso-zéro doivent être uniquement définis, ce qui n'est pas le cas car on a décidé de retenir tous les triangles que l'on pouvait former avec les points d'intersection entre éléments et iso-zéro. Cependant, en utilisant cette idée comme point de départ, on peut différencier les situations des figures 4.3.11-c et 4.3.11-d , même si les triangles ne sont pas uniquement définis.

Tout d'abord, le cas de la figure 4.3.11-d peut être résolu en vérifiant si les deux points définissant l'arête bleue sont partagés entre deux éléments de la grille auxiliaire. On reprend figure 4.3.11-e le cas de la figure 4.3.11-d et on suppose pour simplifier que la grille auxiliaire est composée de deux seuls éléments. L'arête sur laquelle le point problématique a été projeté est représentée en bleu. Les deux points (en bleu) qui la définissent sont partagés entre les deux éléments, ce qui permet de conclure que l'arête est elle-même partagée.

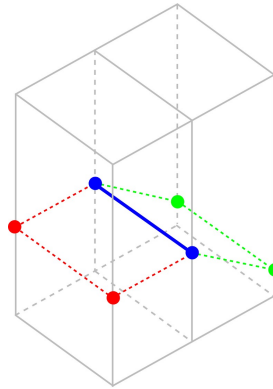


Figure 4.3.11-e: on reprend le cas de figure 4.3.11-d et , pour simplifier la discussion, on fait l'hypothèse que la grille auxiliaire est formée seulement de deux éléments. Les points de chaque élément sont colorés en rouge et en vert. Les points bleus sont communs aux deux éléments. L'arête bleu est celle à laquelle la projection du point problématique de la figure 4.3.11-d appartient.

Dans le cas où les deux points de l'arête ne sont pas communs aux deux éléments, on ne peut rien conclure. En reprenant le cas de la figure 4.3.11-e , si on suppose que l'arête sur laquelle le point a été projeté appartient seulement à l'élément de gauche (points rouges), on peut avoir les deux situations de la figure 4.3.11-f .

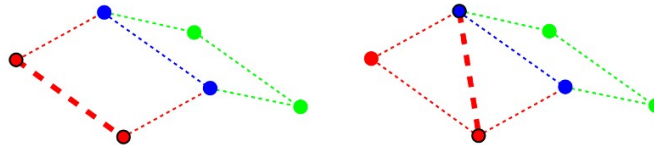


Figure 4.3.11-f: cas de la figure 4.3.11-e. On suppose que l'arête sur laquelle le point a été projeté appartient seulement à l'élément à gauche (points rouges). Deux situations sont possibles: l'arête est sur la surface libre de l'élément (situation à gauche) ou l'arête est à l'intérieur de l'élément (situation à droite).

Pour distinguer ces deux cas, on peut utiliser la propriété de convexité des éléments de la grille auxiliaire. La situation est représentée figure 4.3.11-g , où un plan normal à la surface iso-zéro et passant par l'arête de projection a été dessiné. Dans le cas où l'arête est sur la surface libre, tous les points de l'élément sont du même côté du plan (cas à gauche de la figure). Sinon, on trouve des points des deux côtés du plan (cas à droite de la figure).

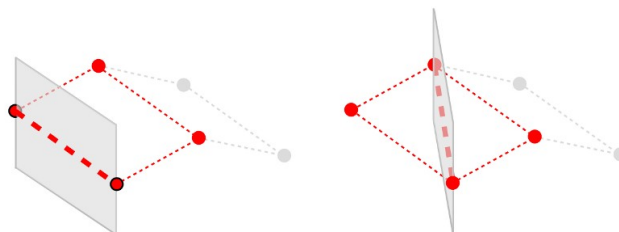


Figure 4.3.11-g: représentation du plan normal à la surface iso-zéro passant par l'arête sur laquelle le point problématique a été projeté.

Pour vérifier dans quelle situation de la figure 4.3.11-g on est, on peut utiliser le triangle sélectionné par la méthode de projection (figure 4.3.11-h) selon la procédure suivante:

- on calcule la normale au triangle sélectionné n_t : $n_t = \frac{\vec{12} \wedge \vec{13}}{|\vec{12} \wedge \vec{13}|}$,
- on calcule la normale au plan défini par la normale au triangle n_t et l'arête a : $n_p = \frac{\vec{a} \wedge \vec{n}_t}{|\vec{a} \wedge \vec{n}_t|}$,
- on initialise la variable qui contient le nombre de points du côté du plan orienté par le vecteur n_p : $c=2$, incluant déjà les points 1 et 2.
- boucle sur les points P_i de l'élément (points rouges de la figure 4.3.11-h)
- si $P_i \neq 1$ et $P_i \neq 2$:
 - on calcule le produit scalaire entre n_p et l'arête reliant les points 1 et P_i : $p = \vec{1P_i} \cdot \vec{n}_p$
 - si $p > 0$, le point P_i est du même côté du plan que n_p : $c = c + 1$
 - fin de boucle
- si c est égal à 2 ou au nombre de points d'intersection de l'élément avec la level set (points rouges de la figure 4.3.11-h), alors on conclut que tous les points sont du même côté du plan et que l'arête est sur la surface libre. Sinon, l'arête n'est pas sur la surface libre.
- si c est égal à 2, tous les points de l'élément sont de l'autre côté du plan dans la direction opposée à celle pointée par n_p . On invertit donc cette normale de façon telle qu'elle pointe dans le volume de la structure : $n_p = -1 \cdot n_p$. Cette propriété sera utile dans la suite de la discussion.

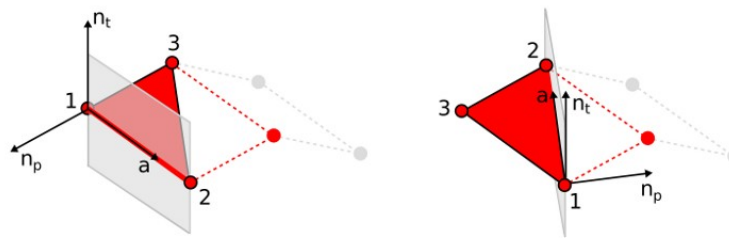


Figure 4.3.11-h: la méthode de projection donne en sortie un triangle, un point projeté et une arête à laquelle ce point appartient. On suppose que dans les deux cas de la figure 4.3.11-h le triangle est celui coloré en rouge et que l'arête est celle identifiée par les points 1 et 2 (vecteur a). La position du point projeté sur cette arête n'a pas d'importance. Le point 1 est pris comme origine des vecteurs utilisés pour le calcul.

Dans la discussion ci-dessus on a supposé que le point projeté, après correction, appartient à une arête du triangle. Cela permet de comprendre si le point projeté, avant correction, est en dehors du volume de la structure et de bien calculer la distance entre le point problématique et la surface iso-zéro. Toutefois il peut arriver que le point projeté, après correction, coïncide avec un des trois points qui définissent le triangle (un exemple est donné figure 4.3.11-i). Ce cas ne peut pas être traité selon la procédure décrite ci-dessus. Cependant on peut toujours utiliser la propriété de convexité des éléments de la grille auxiliaire pour vérifier si le point projeté, avant correction, est à l'extérieur du volume de la structure.

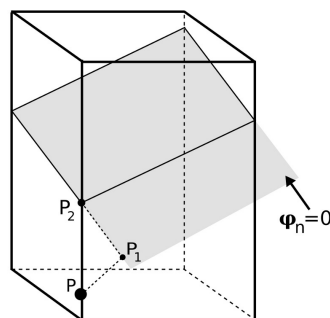


Figure 4.3.11-i: exemple de projection d'un point P donnant un point projeté P_2 , après correction, qui coïncide avec un des points utilisés pour la discrétisation de l'iso-zéro de la level set.

La procédure à suivre est la suivante (figure 4.3.11-j). On fait l'hypothèse que le point projeté, après correction, coïncide avec le point 1 :

- on calcule la normale n_i au triangle sélectionné comme déjà décrit
- on initialise la variable booléenne qui est mise à vrai si le point projeté, avant correction, est à l'extérieur du volume: $e=false$
- boucle sur les arêtes qu'on peut former avec le point 1 (arêtes $\vec{13}$, $\vec{12}$ et $\vec{14}$ de la figure 4.3.11-j)
- on calcule la normale n_p au plan défini par la normale au triangle n_i et l'arête comme déjà détaillé ci-dessus (figure 4.3.11-h)
- Si l'arête est sur la surface libre:
- on oriente la normale au plan de façon telle qu'elle pointe dans le volume de la structure. Cela a été déjà détaillé ci-dessus (figure 4.3.11-h)
- on calcule la distance du point P_1 (point projeté avant correction) au plan: $d = \vec{1P_1} \cdot n_p$
- si $d < 0$ et $|d| > \text{tolérance}$, le point projeté, avant correction, est à l'extérieur du volume: $e=true$
- fin de boucle
- si $e=true$, alors on conclut que le point projeté, avant correction, est à l'extérieur du volume

Dans Code_Aster une tolérance égale à 1% de la longueur de la plus petite arête du maillage est utilisée.

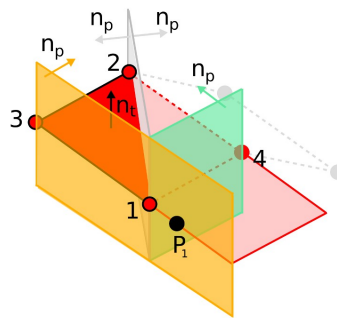


Figure 4.3.11-j: cas de la figure 4.3.11-e. On suppose que le point projeté, après correction, coïncide avec le point 1 et que le triangle sélectionné est le triangle rouge. Le point projeté, avant correction, est le point P_1 de la figure. Les plans orange et vert sont sur la surface libre et leur normale pointe toujours vers l'intérieur du volume. Le plan gris est dans le volume et sa normale ne peut pas être orientée.

L'algorithme décrit ci-dessus est pertinent dans la plupart des cas, sauf en présence de grands angles de bifurcation de la fissure et pour une surface fissurée qui n'est pas normale à la surface de la structure. Dans ce dernier cas la présence d'une discontinuité de la normale à la surface de la fissure entraîne que la distance d'un point à la surface n'est pas uniquement définie si son projeté coïncide avec le point où on a la discontinuité. Un exemple est donné figure 4.3.11-k (bib69,bib69).

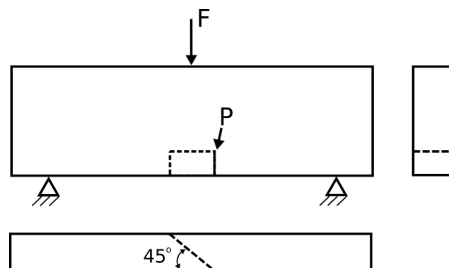


Figure 4.3.11-k: exemple de propagation 3D en mode mixte d'une fissure qui n'est pas normale à la surface libre de l'éprouvette. Suite à la propagation, la surface de la fissure au point P (et au point symétrique de l'autre côté de l'éprouvette) présente une discontinuité de normale.

La configuration de la surface de la fissure après un premier pas de propagation est visible sur la figure 4.3.11-l. Au point P l'angle de propagation est presque de 60 degrés (bib69) et la surface de la fissure n'est pas normale à la surface libre de l'éprouvette.

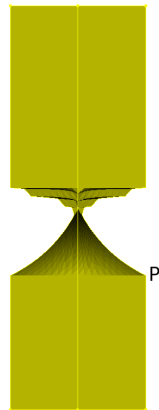


Figure 4.3.11-l: iso-zéro de la level set normale (le domaine de calcul est localisé [§46]) après un premier pas de propagation pour l'éprouvette de la figure 4.3.11-k. La fissure se propage avec un angle très important en surface (au point P)

Les points sur la surface libre de l'éprouvette qui sont à droite de l'iso-zéro de la level set normale de figure la 4.3.11-l sont des points problématiques pour la méthode upwind où les conditions aux limites sont imposées suivant l'algorithme décrit ci-dessus. Les iso-surfaces de la level set normale sont montrées figure 4.3.11-m. La présence d'un angle de propagation très important en surface pose des problèmes: les conditions aux limites calculées pour les points noirs de la figure 4.3.11-m ne sont pas les bonnes.

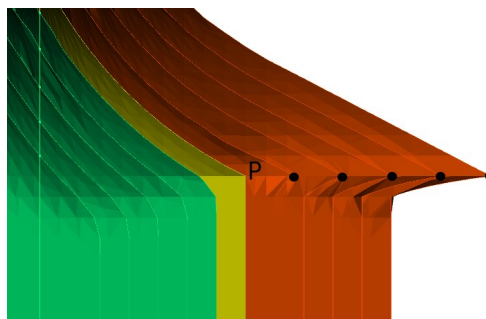


Figure 4.3.11-m: exemple d'iso-surfaces pour la level set normale. Pour celles en rouge (à droite de l'iso-zéro), les conditions aux limites décrites dans ce paragraphe ont été utilisées. Pour celles en vert, aucune condition aux limites n'a été utilisée. Les petits points noirs montrent les points où les conditions aux limites calculées ne sont pas bonnes.

Figure 4.3.11-n, on simule la projection du point A qui est un des points problématiques marqués avec un point noir sur la figure 4.3.11-m.

- Si on considère l'iso-zéro avant la propagation (lignes en rouge), le point de cette surface le plus proche du point A est le point P . Il existe un triangle $TRI1$ contenant P , pour lequel le point A se projette au point C . Le point C étant en dehors du volume de l'éprouvette, l'algorithme de projection retient le point C comme point projeté.
- Si on considère l'iso-zéro après propagation (lignes en bleu), le point de cette surface le plus proche du point A est de nouveau le point B . Il existe un triangle $TRI2$ contenant P , pour lequel le point A se projette au point B . Le point B étant en dehors du volume de l'éprouvette, l'algorithme de projection retient le point B comme point projeté.

Dans les deux cas, c'est le point P qui est le plus proche du point A . Les deux triangles de projection peuvent être choisis indifféremment, les points C ou B étant alors aussi choisis indifféremment. La distance du point A à sa projection sur l'un des triangles dépendra donc du choix de ce dernier ainsi que la distance associée.

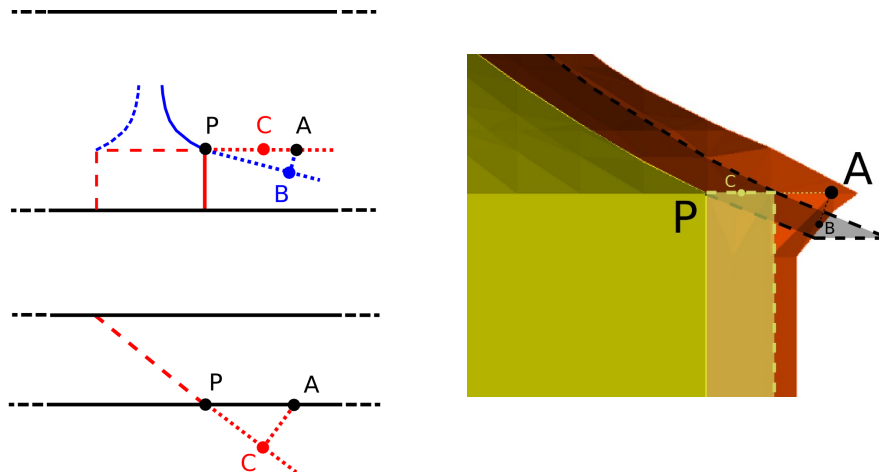


Figure 4.3.11-n: projection d'un point A problématique de la figure 4.3.11-m sur l'iso-zéro de la level set normale selon l'algorithme décrit dans ce paragraphe. La projection est montrée à gauche sur les vues frontale et haute de la figure 4.3.11-k et à droite en 3D. Les surfaces transparentes sont obtenues par extension de l'iso-zéro à l'extérieur du volume de l'éprouvette.

Si on considère un point A^* sur la même surface libre de l'éprouvette que le point A mais en dessous (figure 4.3.11-p), on peut voir que le point de l'iso-zéro le plus proche (point P^*) est toujours sur l'iso-zéro d'avant la propagation (lignes en rouge) indépendamment de l'ordre dans lequel les triangles sont élaborés. Le point C^* est donc toujours pris comme projection du point A^* . Il est évident que $\overline{A^*C^*} = \overline{AC}$.

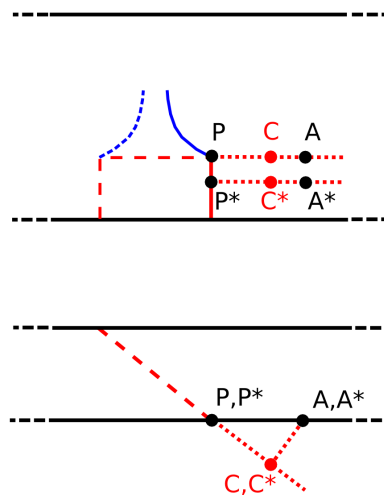


Figure 4.3.11-o: différence entre la projection du point A de la figure et le point A^* qui est sur la même surface libre de l'éprouvette mais en dessous du point A .

Ces observations permettent de comprendre ce qu'il se passe dans le cas de figure 4.3.11-m: le triangle $TRI2$, correspondant à la nouvelle iso-zéro créée par la propagation (en bleu en figure), a été sélectionné comme triangle le plus proche pour le point A et le point B a été pris comme point projeté. La distance du point A à l'iso-zéro a été donc calculée comme $\overline{AB}$. Par contre, pour le premier point en dessous du point A (point A^*), le point C^* a été sélectionné comme point projeté et sa distance à l'iso-zéro a été calculée comme $\overline{A^*C^*}$. Visiblement $\overline{A^*C^*} = \overline{AC} > \overline{AB}$ et donc la valeur de la level set affectée au point A^* ($\overline{A^*C^*}$) est supérieure à celle affectée au point A ($\overline{AB}$). Cela entraîne que l'iso-zéro qui passe par le point A^* dévie fortement à droite à proximité du point A .

Pour éviter ce problème on peut décider d'utiliser pour le calcul de la distance le point projeté après correction (point P) au lieu de celui avant correction (point B) et d'affecter au point A la distance $\overline{AP}$. Toutefois dans ce cas la distance calculée est plus élevée que la distance attendue pour avoir continuité avec la distance calculée au point A^* ($\overline{AP} > \overline{A^*C^*}$) et la level set au point A dévie alors à gauche (figure 4.3.11-q).

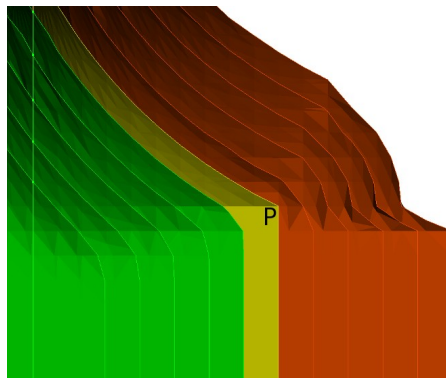


Figure 4.3.11-p: iso-surfaces de la level set normale obtenues en utilisant le point projeté après correction pour le calcul de la condition aux limites.

En considérant que la déformation anormale de la level set normale est limitée aux points sur la surface du domaine de calcul qui sont proches au point P et que la définition de la distance à l'iso-zéro pour ces points est arbitraire à cause de la présence du fort angle de propagation au point P , on décide dans ce cas d'évaluer la valeur de la level set normale comme moyenne des valeurs calculées en utilisant les points projetés avant et après correction. Le résultat qu'on obtient est satisfaisant (figure 4.3.11-q).

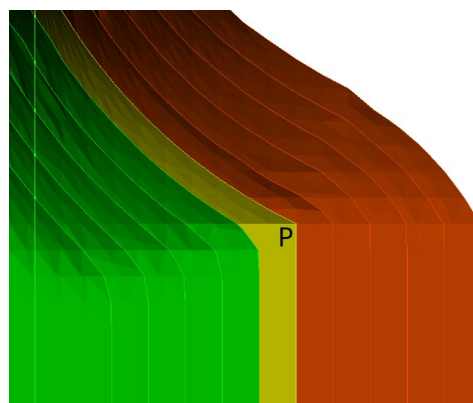


Figure 4.3.11-q: iso-surfaces de la level set normale obtenues en utilisant pour le calcul de la condition aux limites la valeur moyenne des valeurs utilisées pour les figure 4.3.11-m et figure 4.3.11-p.

Pour décider si on doit utiliser cette valeur moyenne pour un point problématique à la place de la valeur issue par l'algorithme décrit ci-dessus, il suffit de vérifier si le point projeté après correction est coïncidant avec un des points extrémité du fond de la fissure (point P de la figure 4.3.11-k). Il faut remarquer que si l'angle de propagation au point P n'est pas trop grand, le problème décrit ci-dessus disparaît.

En 2D tous les calculs détaillés ci-dessous pour établir si le point projeté, avant correction, est à l'extérieur du volume de la structure, y compris sa correction en présence d'un fort angle de propagation, ne sont pas nécessaires. La situation est en effet plus simple. La méthode de projection d'un point sur l'iso-zéro ne change pas parce que des triangles virtuels sont utilisés (voir [§44]). Ces triangles sont toujours dans le plan de la structure ($X-Y$), ce qui fait que le point projeté, avant correction, est dans l'une des deux situations suivantes (figure 4.3.11-r):

- soit le point est sur une des arêtes d'un triangle (point A_1 de la figure 4.3.11-r),
- soit le point coïncide avec un des points définissant la surface iso-zéro, qui se réduit à une ligne dans le plan de la structure en 2D (points B_2 et C_2 de la figure 4.3.11-r).

Dans le premier cas, on ne fait rien. Dans le second cas, on peut regarder si le point projeté, avant correction (B_1 et C_1), est en dehors du volume de la structure en vérifiant simplement si le point projeté, après correction (B_2 et C_2), est partagé avec un autre élément de la grille auxiliaire, ce qui est équivalent à la méthode utilisée dans le cas 3D et illustré figure 4.3.11-d. Dans le cas où le point est partagé (B_2), le point projeté après correction (B_2) est celui à utiliser pour le calcul de la distance. Dans l'autre cas (C_2) on doit utiliser le point projeté avant correction (C_1).

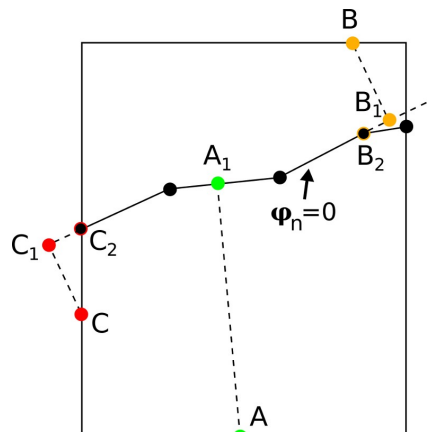


Figure 4.3.11-r: exemple de projection en 2D de trois points problématiques A , B et C . Le point projeté avant projection est noté avec l'indice 1 (A_1 , B_1 et C_1), celui après correction avec l'indice 2 (B_2 et C_2).

Remarque

L'injection des bonnes valeurs des level sets comme condition aux limites aux points problématiques est efficace seulement si le domaine de calcul est localisé (voir paragraphe [§46]), ce qui est toujours le cas. Toutefois on peut utiliser la méthode upwind sans localisation de domaine, même si cela augmente les temps de calculs et rend la méthode moins robuste si la fissure est proche des bords du domaine. Dans ce cas, le traitement des conditions aux limites proposé ici conduit à un nombre d'itérations de convergence pour la méthode upwind beaucoup plus élevé que le nombre strictement nécessaire pour une bonne mise à jours des level sets juste autour du fond de la fissure (bib69).

Dans le cas où l'utilisateur de Code_Aster choisit de ne pas utiliser la localisation du domaine de calcul, la condition aux limites suivante est utilisée:

$$\nabla \varphi = 0$$

où la composante problématique de la dérivée de la level set $D_i \varphi^n(x_k)$ ([§ 24]) est nulle. Ce choix, similaire à un lissage aux frontières, permet d'obtenir une très bonne

convergence de la méthode même si les level sets obtenues sont légèrement déformées à proximité des points où la condition aux limites est imposée (les iso-zéro tendent à devenir normales au bord du domaine là où la condition aux limites est imposée). Normalement ces déformations ne posent pas de problèmes parce qu'elles sont toujours éloignées du fond de la fissure, ce qui n'est pas le cas si la localisation du domaine est utilisée.

L'utilisation de la localisation du domaine est cependant fortement conseillée dans tous les cas afin de limiter les temps de calculs.

4.4 Calcul des level sets à proximité des iso-zéros

Les méthodes présentées précédemment ne donnent de résultats exacts que si les iso-zéros des level sets passent effectivement par des nœuds. Lorsque l'iso-zéro passe entre deux nœuds, elle a tendance à être légèrement déformée lors de la réinitialisation. Un algorithme de calcul des iso-zéros est donc mis en place. Celui-ci calcule géométriquement les distances signées sur les nœuds des mailles coupées ou tangentées par l'iso-zéro de la level set et fixe la valeur calculée pour la réinitialisation.

Concernant la réorthogonalisation, les valeurs de lst situées sur la surface ($lsn=0$) ne doivent pas être modifiées ; on profitera donc de l'algorithme de calcul de lsn pour calculer les valeurs de lst orthogonales sur les éléments coupés ou tangentés par ($lsn=0$) et figer ces valeurs pendant la phase de réorthogonalisation. Pour cela, la valeur de lst est interpolée sur les triangles IJK que l'on peut former avec les points d'intersection $(lsn=0) \cap arêtes$. Si pour un nœud M , la projection orthogonale de M est à l'intérieur de IJK , on interpole la valeur de $lst(M)$ comme lst du point projeté. Sinon, $lst(M)$ est calculée comme lst du projeté ramené à l'intérieur du triangle IJK (Figure 4.4-a).

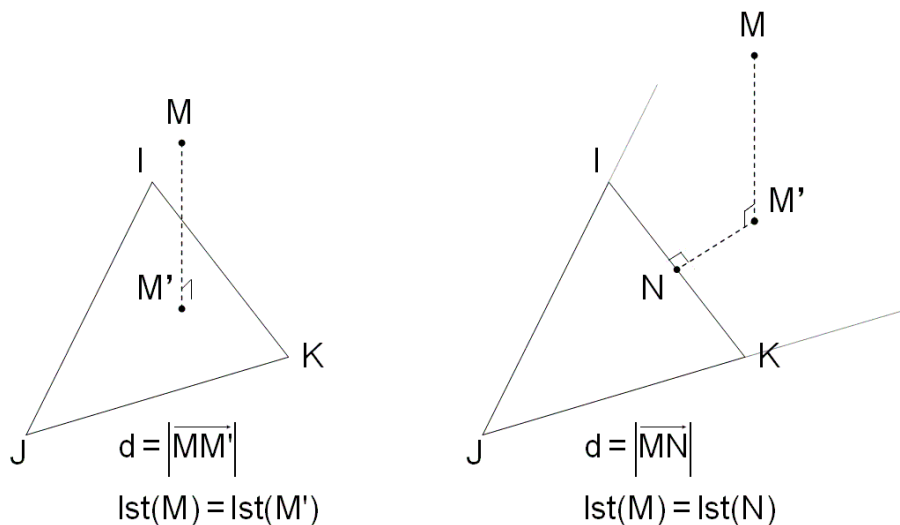


Figure 4.4-a : Calcul de distance et interpolation de lst sur les éléments coupés par $lsn=0$.

En 3D, l'algorithme mis en place après la propagation de lsn et avant sa réinitialisation est le suivant :

- Boucle sur les mailles
 - Si au moins 3 nœuds de la maille sont sur $lsn=0$, on la repère comme maille coupée
 - Si la maille a une arête $[AB]$ telle que $lsn(A) \cdot lsn(B) < 0$, on repère la maille coupée
- Fin de boucle
- Boucle sur les nœuds
 - Si le nœud appartient à une maille coupée, il sera à calculer
- Fin de boucle
- Boucle sur les nœuds P à calculer
 - Boucle sur les mailles coupées dont le nœud est sommet
 - Boucle sur les arêtes $[AB]$ de la maille
 - Si $lsn(A) \cdot lsn(B) \leq 0$, on interpole le point I_{AB} :

$$\overrightarrow{AI}_{AB} = s \cdot \overrightarrow{AB} \quad \text{avec} \quad s = \frac{|lsn(A)|}{|lsn(A)| + |lsn(B)|}$$
 et $lst(I_{AB}) = lst(A) + s \cdot (lst(B) - lst(A))$
 - Fin de boucle
 - Boucle sur tous les triangles IJK que l'on peut former avec les points I_{AB}
 - On calcule le projeté M de P sur le triangle IJK
 - On repère si le point projeté M est à l'intérieur du triangle
 - On stocke la distance $d = PM$
 - On interpole linéairement $lst(M)$ à partir de $lst(I)$, $lst(J)$ et $lst(K)$
 - Fin de boucle
- Fin de boucle
- Parmi les distances PM calculées on recherche la plus petite distance avec M à l'intérieur de IJK . Si aucune projection à l'intérieur de IJK n'a été trouvée, on prend juste la plus petite distance calculée et la valeur de lst correspondante.
- On stocke les valeurs de d et lst correspondantes pour le nœud P .
- Fin de boucle
- On remplace lsn et lst sur les nœuds à calculer par les nouvelles valeurs stockées de d et lst .

Les nœuds repérés comme « nœuds calculés » ont des level sets qui sont alors figées pendant les phases de réinitialisation de lsn et de réorthogonalisation de lst .

L'algorithme lancé après les phases de propagation et de réorthogonalisation de lst et avant sa réinitialisation est le même, mais il calcule uniquement les distances lst , de la même manière que lsn précédemment, sans toucher aux valeurs de lsn .

L'algorithme décrit ci-dessus est valide seulement en 3D car en 2D on ne peut pas former des triangles. En effet deux points d'intersection seulement peuvent être détectés pour chaque élément. Pour utiliser en 2D le même algorithme que pour le cas 3D, on peut utiliser la solution suivante. On forme un triangle utilisant les deux points d'intersection (A et B) et un point virtuel (C). Celui-ci est choisi de façon à ce que le triangle formé soit dans un plan orthogonal au plan du modèle (plan $X-Y$). Pour éviter des problèmes numériques, les coordonnées du point virtuel (C) sont calculées de la manière suivante:

- X, Y = coordonnées du point milieu de l'arête reliant les deux points d'intersection (A et B),
- Z = distance entre les deux points d'intersection (A et B).

5 Localisation du domaine de calcul des level sets

Au paragraphe [§27] on a dit que, dans le cas de l'utilisation d'une grille auxiliaire pour la méthode upwind, les valeurs des level sets sont mises à jour, réinitialisées et réorthogonalisées pour tous les points de la grille mais que seulement les valeurs dans une région autour du fond de la fissure sont projetées sur le maillage de la structure. Cela montre que la mise à jour des level sets, leur réinitialisation et leur réorthogonalisation sont nécessaires uniquement dans la région autour du fond de la fissure, indépendamment de l'utilisation ou non d'une grille auxiliaire, et que le domaine déjà défini pour la projection entre grille auxiliaire et maillage de la structure ([§27]) peut aussi être utilisé pour localiser le domaine de calcul. Un avantage de cette localisation est évidemment que le temps de calcul nécessaire pour la mise à jour, la réinitialisation et la réorthogonalisation des level sets est inférieur par rapport au cas où tous les nœuds et éléments du maillage ou de la grille auxiliaire sont utilisés. De plus, la localisation permet aussi d'améliorer la robustesse des méthodes simplexe et upwind (bib69). En fait, dans le cas où les level sets sont mises à jour en des nœuds/points qui sont très loin du fond de la fissure en présence d'oscillations sur l'angle et la vitesse de propagation entre des points successifs du fond de la fissure, les iso-zéros de la level set normale peuvent présenter des irrégularités très marquées à cause de la variation linéaire imposée à la vitesse normale ([§9]). Ces irrégularités peuvent rendre difficile la convergence des phases de réinitialisation et réorthogonalisation qui suivent la mise à jour. Un exemple illustratif est décrit figure 5-a et figure 5-b. On y montre une fissure 3D se propageant en mode mixte selon un angle de propagation qui change de façon linéaire le long du fond de la fissure. Des petites perturbations ont été superposées à la valeur théorique attendue, ce qui fait que la variation de l'angle de propagation avec la position sur le fond n'est pas parfaitement linéaire (figure 5-a). Ces perturbations simulent la présence de petites erreurs dans un calcul numérique de valeurs d'angle.

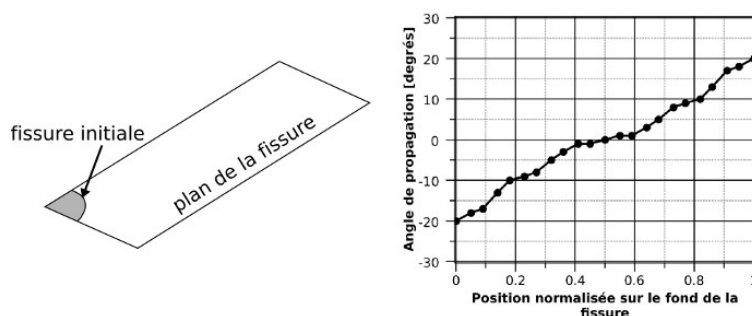


Figure 5-a: configuration initiale (à gauche) de la simulation d'une propagation hors plan d'une fissure 3D selon un angle de propagation qui change de façon presque linéaire par rapport à la position considérée sur le fond de la fissure (graphe à droite).

Même si les perturbations de l'angle de propagation, et donc de la vitesse normale, sont petites, elles sont amplifiées par la variation linéaire imposée à cette composante de la vitesse dans le domaine de calcul, ce qui est très clair si on visualise l'iso-zéro de la level set normale après la seule phase de mise à jour (figure 5-b). La localisation du domaine de calcul permet de limiter cet effet (figure 5-c).

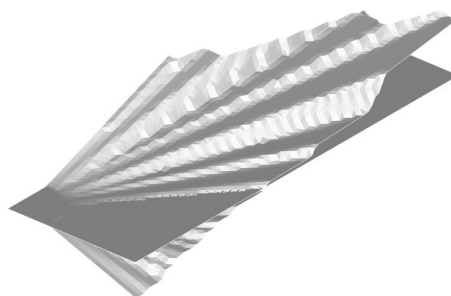


Figure 5-b: iso-zéro de la level set normale après la seule phase de mise à jour. L'angle de propagation de la figure 5-a a été imposé. Le plan de la fissure initiale a été superposé pour comparaison.

Figure 5-c: iso-zéro de la level set normale obtenue avec une restriction du domaine de mise à jour à comparer à celle de la figure 5-b. Il faut remarquer que pour cette figure le plan de la fissure initiale n'a pas été représenté, au contraire de ce qui a été fait pour la figure 5-b. On peut donc en conclure que la valeur de la level set n'a pas été changée pour les points en dehors du domaine de calcul.

5.1 Sélection des nœuds/points et des éléments du domaine de calcul localisé

Le domaine à utiliser pour la localisation est celui déjà défini pour la projection entre la grille auxiliaire et le maillage de la structure ([§27]). En effet ce domaine est étendu à tous les points utilisés pour les calculs mécaniques et il permet de préserver la surface de la fissure existante, comme déjà montré (figure 4.3.9-b). Les méthodes simplexe et upwind seront appliquées seulement aux nœuds et aux éléments contenus dans ce domaine, qui est défini par le cercle 3 de figure 4.3.9-a. En 3D ce domaine coïncide avec un tore construit autour du fond de la fissure et de rayon égal à celui du cercle 3. Dans tout les cas le rayon du domaine localisé est donc égal à :

$$R_{loc} = \Delta a + R_{calcG}$$

où Δa est l'avancée de la fissure à imposer (choisie par l'utilisateur) et R_{calcG} est le rayon du domaine utilisé par CALC_G.

On peut donc tout d'abord sélectionner les nœuds de la grille ou du maillage qui ont une distance au fond de la fissure inférieure ou égale à R_{loc} (figure 5.1-a à gauche). Cette distance peut être calculée de façon très rapide parce que pour chaque nœud on connaît déjà son projeté sur le fond de fissure ([§ 5]). Puis on prend tous les éléments qui ont au moins un des nœuds sélectionnés précédemment (éléments gris de la figure 5.1-a de droite). Cela permet de sélectionner correctement tous les éléments qui sont coupés par la frontière du domaine et qui sont donc partiellement contenus dans le domaine. De cette façon, toute la surface (ou le volume en 3D) du domaine est correctement recouvert par les éléments sélectionnés. Enfin la liste des nœuds du domaine doit être mise à jour en ajoutant les nœuds des éléments partiellement contenus dans le domaine (nœuds marqués avec un petit cercle sur la figure 5.1-a de droite). La façon la plus rapide et facile de le faire est de reconstruire la liste en prenant tous les nœuds des éléments sélectionnés. Une fois que la liste des nœuds a été mise à jour, la valeur du rayon R_{loc} du domaine discrétisé, c'est-à-dire du domaine constitué par les nœuds et les éléments sélectionnés, doit être mise à jour en prenant la valeur au nœud du domaine discrétisé le plus loin du fond. Cette opération est nécessaire pour la partie qui va suivre ([§49]).

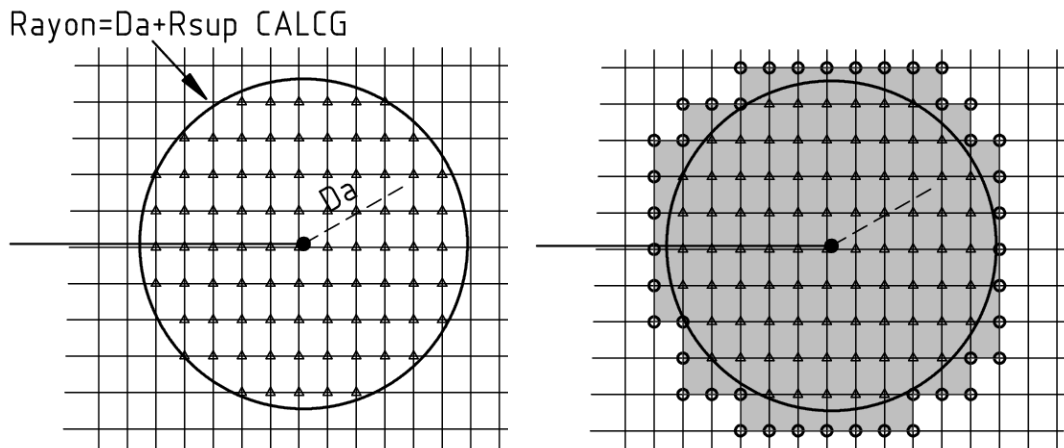


Figure 5.1-a: exemple de sélection des nœuds et des éléments du domaine de localisation. On suppose que ce domaine coïncide avec le cercle de la figure. La première étape (à gauche) est la sélection des nœuds à une distance inférieure ou égale à R_{loc} . Les nœuds sélectionnés sont marqués avec un petit triangle. Ensuite les éléments qui contiennent au moins un des ces nœuds dans leur définition sont sélectionnés (éléments gris à droite). Enfin on sélectionne tous les nœuds des éléments qui ont été pris: les nœuds marqués avec des petits cercles viennent s'ajouter à ceux qui ont été sélectionnés au début.

5.2 Déplacement du domaine localisé en suite à la propagation de la fissure

La mise à jour, la réinitialisation et la réorthogonalisation des level set sont faites exclusivement aux nœuds définissant le domaine localisé (nœuds marqués avec des triangles et des cercles sur la figure 5.1-a de droite). Cela signifie que les valeurs des level set aux nœuds à l'extérieur du domaine localisé ne sont pas changées, ce qui pose des problèmes au cours des pas de propagation successifs. La situation est illustrée figure 5.2-a pour un point P_1 dans le plan $t-n$ (plan e_1-e_2 de la figure 2.1-2). Au début de la propagation le fond de la fissure coïncide avec le point P_1 et le domaine localisé avec le cercle Ω_1 . Pour simuler la propagation, les level set sont mises à jour, réinitialisées et réorthogonalisées seulement dans ce cercle. La nouvelle position du fond après le premier pas de propagation coïncide avec P_2 . Au deuxième pas de propagation, le fond de la fissure coïncide avec le point P_2 et un nouveau domaine Ω_2 est construit autour de ce point pour la mise à jour, la réinitialisation et la réorthogonalisation des level sets. La valeur des level sets n'a pas été mise à jour au pas de propagation précédent pour tous les points de Ω_2 : seulement les level sets des points de Ω_2 en commun avec Ω_1 sont à jour (points de la surface grise de la figure 5.2-a). Pour tous les autres points, la valeur des level sets est celle que l'on avait au début de la propagation. Les bonnes valeurs en ces autres points doivent donc être calculées au début du deuxième pas de propagation.

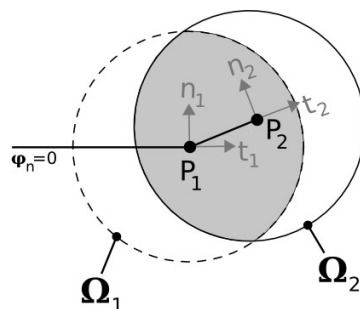


Figure 5.2-a: déplacement du domaine localisé entre deux pas de propagation successifs. Le fond de la fissure se déplace de P_1 à P_2 .

Le calcul des bonnes valeurs des level sets pour tous les points dans la région blanche du domaine Ω_2 de la figure 5.2-a peut être fait en utilisant la définition de distance signée des level sets (bib69). Pour chaque point M dans cette région (figure 5.2-b), les level set peuvent être calculées comme ceci:

$$\varphi_n(M) = \overrightarrow{MP_2} \cdot \mathbf{n}_2$$

$$\varphi_t(M) = \overrightarrow{MP_2} \cdot \mathbf{t}_2$$

Ces équations permettent de calculer la bonne valeur des level sets avec le bon signe. Il faut remarquer que, dans le cas 3D, le point P_2 est le point projeté du point M sur le fond de la fissure. Pour chaque point M , son projeté et la base $\mathbf{n}_2 - \mathbf{t}_2$ ont déjà été calculés pendant la phase d'extension de la vitesse du fond de la fissure aux points du domaine de calcul ([§5]). Tous les ingrédients du calcul sont donc déjà disponibles et pour chaque point M l'évaluation de la bonne level set est limitée au calcul de deux produits scalaires, ce qui est très rapide à faire.

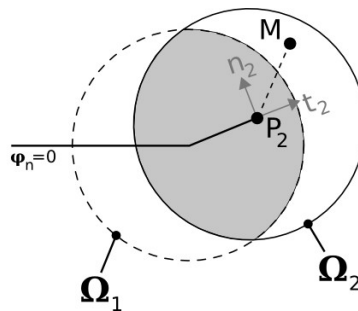


Figure 5.2-b: calcul des bonnes valeurs des level sets aux points de Ω_2 qui sont à l'extérieur du domaine Ω_1 du pas de propagation précédent. Le calcul est fait tout au début de la propagation courante, c'est-à-dire avant la mise à jour, la réinitialisation et la réorthogonalisation des level sets.

Il faut remarquer que même si les valeurs des level sets calculées par les équations ci-dessus ne sont pas très précises, la phase de mise à jour ne sert qu'à l'initialisation de ces dernières et les bonnes level sets seront récupérées par les phases de réinitialisations et de réorthogonalisation.

5.3 Limitations sur la valeur du rayon du domaine localisé entre deux propagation successives

Les équations données au paragraphe 48 pour le calcul des bonnes valeurs des level sets peuvent être appliquées seulement aux points pour lesquels la base $\mathbf{n}_2 - \mathbf{t}_2$ est valide pour le calcul de la distance signée. Cela signifie que le point M doit appartenir au demi-plan montré figure 5.3-a, qui contient la base $\mathbf{n}_2 - \mathbf{t}_2$, le fond de la fissure P_2 et qui a pour bord une droite normale à l'arête reliant P_1 et P_2 parallèle à $\mathbf{n}_2$.

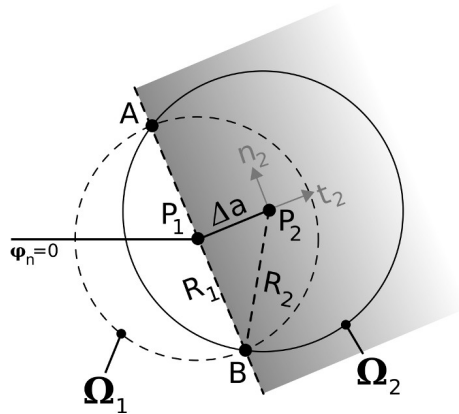


Figure 5.3-a: définition du demi-plan auquel le point M de la figure 5.2-b doit appartenir pour que les équations de calcul des level sets du paragraphe 48 soient valides.

Cette condition sur la position du point M se traduit par une condition sur la valeur maximale du rayon R_2 du domaine Ω_2 : la valeur maximale est celle pour laquelle les deux points d'intersection (A et B) entre les deux domaines Ω_1 et Ω_2 sont sur le bord du demi-plan de validité des équations, comme indiqué figure 5.3-a. Ces deux points sont toujours symétriques par rapport à l'arête reliant P_1 et P_2 . La valeur maximale du rayon R_2 peut donc être calculée facilement:

$$R_2 \leq \sqrt{\Delta a^2 + R_1^2}$$

La valeur du rayon du domaine actuel est donc bornée par les valeurs de l'avancée de la fissure et du rayon du domaine utilisés au pas de propagation précédent. Normalement il n'y a aucun intérêt à changer la valeur du rayon du domaine entre deux propagations successives. Dans ce cas, comme R_1 est égal à R_2 , l'inégalité sur R_2 est nécessairement vérifiée quelle que soit la valeur de l'avancée choisie.

Il faut toutefois remarquer que la valeur du rayon du domaine peut changer par rapport à la valeur théorique donnée par l'utilisateur du fait de la discrétisation du domaine ([§ 47] et figure 5.2-a de droite). Donc, même si on donne $R_2 = R_1$ au pas de propagation courant, la valeur effective du rayon du domaine sera la suivante:

$$R_{2\text{eff}} = R_2 + \Delta R_2 = R_1 + \Delta R_2$$

où ΔR_2 est l'incrément du rayon R_2 du fait de la discrétisation du domaine, qui est de l'ordre de la longueur moyenne des arêtes des éléments autour de la frontière du domaine localisé (voir figure 5.2-a).

Pour la même raison, la valeur effective du rayon R_1 au pas de propagation précédent peut être différente de la valeur théorique donnée par l'utilisateur:

$$R_{1\text{eff}} = R_1 + \Delta R_1$$

où ΔR_1 est l'incrément du rayon R_1 du fait de la discrétisation du domaine, qui est de l'ordre de la longueur moyenne des arêtes des éléments autour de la frontière du domaine localisé au pas de propagation précédent.

Ainsi l'inégalité ci-dessus peut être écrite dans la forme suivante:

$$R_1 + \Delta R_2 \leq \sqrt{\Delta a^2 + (R_1 + \Delta R_1)^2}$$

Dans le cas où la longueur des arêtes de la grille auxiliaire utilisée est uniforme dans le domaine, $\Delta R_2 \approx \Delta R_1$, et l'inégalité est toujours vérifiée. Par contre, si cette longueur change dans le domaine, on pourrait avoir $\Delta R_2 > \Delta R_1$ et l'inégalité pourrait ne pas être vérifiée.

Dans ce dernier cas on doit forcément raffiner le maillage (ou la grille auxiliaire) pour réduire la différence entre ΔR_1 et ΔR_2 ou bien utiliser une avancée de fissure Δa plus élevée.

6 Calcul direct des nouvelles level sets après avancée

Une méthode alternative aux méthodes de mise à jour des level sets est celle décrite dans (bib 70) . Elle est dérivée de l'analyse des équations de mise à jour, de réinitialisation et de réorthogonalisation des level sets décrites aux paragraphes 11 et 14 : on arrive à montrer que le processus de mise à jour des level sets peut toujours être ramené à un processus 2D en mode $I + II$ dans le plan formé par la tangente et la normale à la surface de la fissure en fond de fissure (plan $n_p - t_p$ de la figure 2.3-a) . Cela permet de simplifier énormément le calcul direct des nouvelles valeurs des level sets après avancée. Ce calcul a été effectivement déjà explicité au paragraphe 48 pour ce qui concerne l'évaluation des level sets pour les points hors domaine de calcul qui sont rajoutés. Les équations qui y sont décrites peuvent être utilisées dans le cas où la nouvelle position du fond est connue. En sachant que le problème de mise à jour des level sets peut toujours être ramené à un problème 2D, la nouvelle position du fond peut être facilement calculée à partir de l'avancée de la fissure et de l'angle de propagation au point du fond considéré.

6.1 Méthode de calcul des level sets

Ces idées sont reprises figure 6.1-a , qui montre la configuration des iso-surfaces des deux level sets autour d'un point P du fond d'une fissure 3D obtenue par le processus de mise à jour, de réinitialisation et de réorthogonalisation décrit aux paragraphes 11 et 14 .

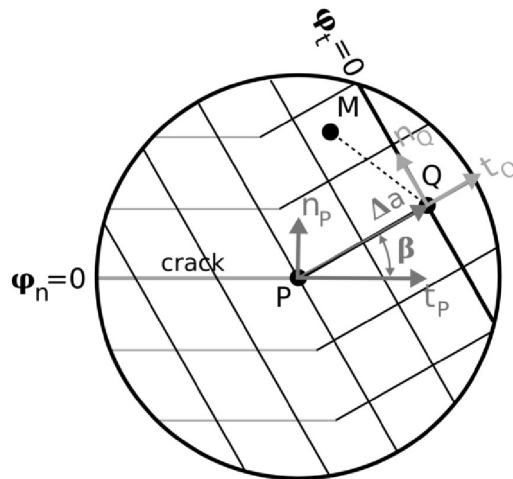


Figure 6.1-a: iso-surfaces des deux level sets obtenues par le processus standard de mise à jour, avec réinitialisation et réorthogonalisation. P est un point du fond d'une fissure 3D. Q est la nouvelle position de ce point après une avancée de Δa avec un angle de propagation β .

On considère un point M du domaine de calcul où on doit calculer la nouvelle valeur des deux level sets. On projette ce point sur le fond de la fissure en utilisant le même algorithme que celui décrit au paragraphe 5 et on obtient le point P (figure 6.1-a) . La base locale (t_p, n_p) et la vitesse d'avancée V_p au point P peuvent être calculées comme décrit aux paragraphes 5 , 6 et 8 . Pour l'angle de propagation, on utilise la même interpolation linéaire que celle utilisée pour la vitesse (voir paragraphe 5 et figure 2.2-b) :

$$\beta = (\beta_J - \beta_I) \cdot s_{min} + \beta_I$$

L'avancée de la fissure au point P du fond peut donc être aisément calculée :

$$\Delta \vec{a} = \|V_p\| \cdot \Delta t \cdot (\sin \beta \cdot n_p + \cos \beta \cdot t_p)$$

où Δt est le temps ou, en termes équivalents, le nombre de cycles de fatigue à simuler au pas de propagation courant (voir discussion du paragraphe 11).

La nouvelle position Q du point P du fond peut donc être calculée comme ceci:

$$\overrightarrow{OQ} = \overrightarrow{OP} + \Delta \vec{a}$$

où O est le point origine de la base globale dans laquelle les coordonnées des points sont exprimées. Enfin la base au point Q peut être calculée par une rotation d'angle β de la base au point P autour de l'axe normal au plan $t_P - n_P$:

$$\begin{aligned} t_Q &= \cos \beta t_P + \sin \beta n_P \\ n_Q &= -\sin \beta t_P + \cos \beta n_P \end{aligned}$$

Tous les ingrédients du calcul des level sets sont maintenant connus. On peut donc utiliser les mêmes équations que celles utilisées au paragraphe 48:

$$\begin{aligned} \varphi_n(M) &= \overrightarrow{QM} \cdot n_Q \\ \varphi_t(M) &= \overrightarrow{QM} \cdot t_Q \end{aligned}$$

Afin de préserver la surface existante de la fissure (iso-surfaces en gris de la figure 6.1-a), le calcul de la level set normale avec l'équation ci-dessus est fait uniquement aux points du domaine où $\varphi_t(M) > 0$.

6.2 Traitement des points problématiques

Les équations écrites au paragraphe précédent s'appliquent si le vecteur $\overrightarrow{PM}$ est dans le plan $t_P - n_P$ (figure 6.1-a) (bib70). Cela n'est pas le cas pour tous les points qui n'ont pas un point de projection normale sur le fond de la fissure. Dans ce cas un point extrémité du fond est sélectionné comme point projeté P par l'algorithme de projection décrit au paragraphe 5.

Ces points problématiques peuvent être traités par une rotation de la base (t_P, n_P) qui permet de rétablir la condition de coplanarité du plan $t_P - n_P$ et du vecteur $\overrightarrow{PM}$ ou $\overrightarrow{QM}$ (bib 70).

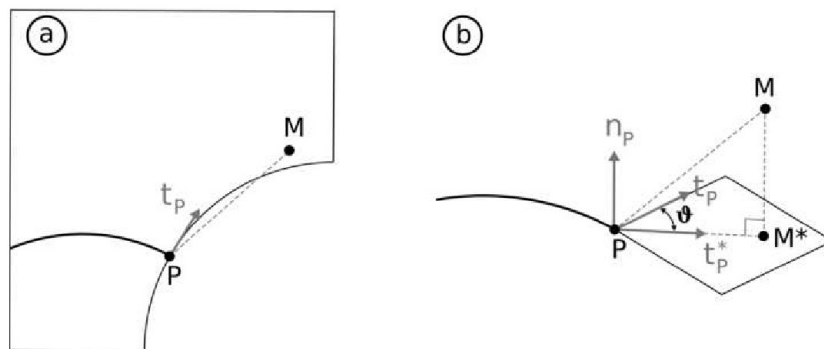


Figure 6.2-a: a) exemple de point problématique. b) rotation de la base au point P pour rétablir la condition de coplanarité entre le plan $t_P - n_P$ et les vecteurs $\overrightarrow{PM}$ ou $\overrightarrow{QM}$.

Un exemple de point problématique est visible sur la figure 6.2-a a. Pour rétablir la condition de coplanarité, on peut faire une rotation θ de la base (t_P, n_P) autour du vecteur n_P de telle manière que le plan $t_P^* - n_P$ contienne le point M (figure 6.2-a b). En effet, de cette façon la nouvelle position du point P après propagation (point Q de la figure 6.1-a) est contenue dans le plan $t_P^* - n_P$, ce qui permet de vérifier la condition de coplanarité.

Le nouveau vecteur t_P^* peut être calculé à partir du projeté M^* du point M sur le plan normal à l'axe n_P (figure 6.2-a b). On calcule d'abord :

$$\overrightarrow{M^*M} = (\overrightarrow{PM} \cdot \mathbf{n}_p) \cdot \mathbf{n}_p$$

ce qui permet d'obtenir le nouveau vecteur:

$$\mathbf{t}_p^* = \frac{\overrightarrow{PM} - \overrightarrow{M^*M}}{\|\overrightarrow{PM}\| - \|\overrightarrow{M^*M}\|}$$

Pour prendre en compte correctement le signe de la level set tangente, le vecteur $\mathbf{t}_p^*$ doit être réorienté suivant la direction donnée par $\mathbf{t}_p$ (bib 70) :

$$\mathbf{t}_p' = \frac{\mathbf{t}_p^* \cdot \mathbf{t}_p}{\|\mathbf{t}_p^* \cdot \mathbf{t}_p\|} \cdot \mathbf{t}_p^*$$

Si les deux vecteurs $\mathbf{t}_p^*$ et $\mathbf{t}_p$ sont orthogonaux, l'équation ci-dessus ne peut pas être appliquée mais on peut écrire tout de suite que:

$$\mathbf{t}_p' = \mathbf{t}_p^*$$

L'utilisation du vecteur $\mathbf{t}_p'$ à la place de $\mathbf{t}_p$ dans les équations du paragraphe 52 permet de bien calculer les level sets pour les points problématiques.

6.3 Discontinuité de la level set normale et limitations de calcul

Le fait de calculer la level set normale seulement aux points M pour lesquels $\varphi_t(M) > 0$ entraîne une discontinuité de la level set normale, comme démontré dans (bib 70). La situation est illustrée figure 6.3-a. La level set normale n'a pas été recalculée dans la zone colorée en gris impliquant une discontinuité des iso-surface entre la zone en gris et la zone où la level set a été recalculée.

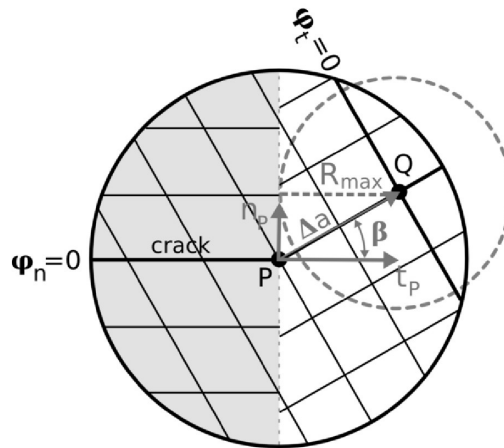


Figure 6.3-a: discontinuité des iso-surface de la level set normale et rayon maximal $R_{\max}$ du domaine de calcul de CALC_G à utiliser.

Comme démontré dans (bib 70), cette discontinuité ne cause aucun problème pour les calculs X-FEM (identification des nouvelles mailles coupées notamment). De plus, la continuité de l'iso-zéro de la level set normale est préservée, ce qui assure la continuité de la surface de la fissure. La seule condition à respecter pour gérer cette discontinuité porte sur la valeur maximale du rayon de calcul de CALC_G

(voir U4.82.03). Cette condition a pour but d'éviter que le domaine de calcul de `CALC_G` contienne la discontinuité (cercle gris de la figure 6.3-a) :

$$R_{max} = \Delta a_{max} \cdot \cos \beta_{max}$$

où on prend la valeur maximale de Δa et β le long du fond de la fissure.

Si elle restreint l'angle de propagation à des valeurs strictement inférieures à $\pi/2$ (sous peine d'un raffinement de plus en plus important à mesure que l'on se rapproche de $\pi/2$), cette condition n'est pas très restrictive parce qu'il est conseillé de respecter une relation similaire même dans le cas de mise à jour des level sets par la méthode simplexe ou upwind.

6.4 Localisation du domaine

Comme montré dans (bib70), la localisation du domaine décrite au paragraphe 46 n'est plus nécessaire si on utilise le calcul direct des level sets.

Tout d'abord, la localisation du domaine n'améliorerait pas la robustesse de l'algorithme. Dans (bib69), il est montré que de faibles erreurs dans le calcul de l'angle de propagation (ce qui est toujours le cas lors de simulations numériques) sont amplifiées aux points situés loin du front de fissure (du fait de l'interpolation linéaire utilisée pour créer le champ de vitesse V_N). Les iso-surfaces de la level set normale ne sont alors plus régulières et ceci peut entraîner l'échec des phases de réinitialisation et de réorthogonalisation des level sets (dans la méthode upwind par exemple) comme il est fait mention au paragraphe 46. La localisation du domaine permet ainsi de réduire le domaine de mis à jour des level sets à un domaine englobant le front de fissure ce qui conduit à une limitation de l'effet d'amplification. Cependant la méthode géométrique gère avec succès les iso-surfaces non régulières puisqu'elle est basée sur des équations polynomiales explicites et non sur des équations différentielles. De plus, la méthode géométrique n'utilise pas de processus de réinitialisation et de réorthogonalisation des level sets (qui sont calculés à chaque pas de propagation). Pour toutes ces raisons, la localisation du domaine n'affecte pas la robustesse de la méthode géométrique.

La localisation du domaine dans le cadre de l'utilisation de la méthode géométrique ne réduirait pas non plus forcément le temps de calcul nécessaire à la mise à jour des level sets. Il s'avère que le calcul des level sets en chaque point du domaine est rapide et que le temps nécessaire pour ce calcul peut être plus faible que le temps nécessaire pour sélectionner les points du domaine localisé. L'algorithme utilisé pour la sélection des points appartenant au domaine restreint est basé sur le calcul de la distance entre un point M du domaine et son projeté P sur le front de fissure. Comme le point P doit être calculé également pour le calcul des level sets, le coût de calcul supplémentaire pour l'algorithme de sélection se réduit au coût du calcul de la distance $\|\overrightarrow{PM}\|$ et la complexité de l'algorithme est de $O(n)$. Même si la complexité de l'algorithme de mise à jour des level sets par la méthode géométrique est de $O(n)$, il apparaît clair que la somme des calculs effectués par l'algorithme de sélection sera moindre que celle effectuée par l'algorithme de mise à jour des level sets. Ceci conduit à penser à première vue que la localisation du domaine serait avantageuse en terme de temps de calcul par rapport au calcul des level sets dans tout le domaine. Cependant, la sélection (basée sur la distance $\|\overrightarrow{PM}\|$) des points à l'intérieur du domaine restreint n'est pas suffisante. En effet, pour le calcul de la base locale en chaque point, il s'agit de déterminer le gradient des level sets ce qui nécessite la sélection des éléments contenant les points sélectionnés (les fonctions de forme des éléments étant utilisées pour calculer les gradients). Même si l'algorithme de sélection ne change pas nécessairement, le nombre d'opérations effectuées augmente et ceci peut rendre le temps de calcul de la sélection plus élevé que le calcul des level sets par la méthode géométrique dans tout le domaine. A noter que, dans le même temps, le temps de calcul des gradients des level sets est aussi plus élevé si la localisation du domaine n'est pas prise en compte.

Finalement, l'avantage de la localisation du domaine en terme de temps de calcul dépend fortement des algorithmes et implémentations choisis et du nombre de points du domaine de calcul. La localisation du domaine pourrait ainsi être avantageuse dans le cas de maillages éléments finis « très gros ». Dans ce cas, la limitation de la mise à jour des level sets à un groupe de nœuds (fixé et défini par l'utilisateur) définissant la surface potentielle de propagation de la fissure peut être un moyen plus efficace de réduire le temps de calcul plutôt que le choix d'un algorithme de localisation du domaine pour lequel un nouveau domaine restreint doit être calculé à chaque pas de propagation afin de suivre le déplacement du front de fissure.

7 Méthode maillage pour la mise à jour des level sets

La méthode la plus intuitive pour mettre à jour les valeurs des level sets à chaque pas de propagation de la fissure consiste à utiliser leur propriété de distance signée. Ceci implique une représentation explicite de la fissure (lèvres de la fissure et fond de fissure) propagée afin de calculer les distances des nœuds à la fissure. Cette méthode utilise ainsi deux maillages distincts (fissure et structure) et seul le maillage de la fissure se trouve modifié au cours de la propagation. La mise à jour du maillage de la fissure à chaque pas de propagation consiste concrètement en la création d'une nouvelle rangée de mailles surfaciques pour représenter l'avancée de la fissure et d'une nouvelle rangée de mailles linéiques pour représenter le nouveau fond. Une fois ce maillage mis à jour, on recalcule les nouveaux level sets par calcul direct de la distance à la fissure (projection orthogonale sur le maillage de la fissure).

7.1 Présentation de la méthode

Le maillage de la fissure est réalisé dans *Code_Aster* par la commande `PROPA_FISS`. La donnée d'entrée est la valeur du vecteur propagation sur le front de fissure, déterminé à partir des facteurs d'intensité des contraintes et de la loi de Paris. Le principe de la méthode maillage est de calculer les nœuds du nouveau fond à partir des vitesses de propagation et des nœuds du fond actuel. Il se résume aux 3 étapes suivantes :

- calcul par interpolation linéaire du vecteur propagation à chaque nœud du front de fissure discrétisé. Le vecteur propagation est orthogonal au front de fissure : la direction de propagation en un nœud correspond à la moyenne des vecteurs normaux des deux éléments linéaires se partageant ce nœud. Cette étape donne la position de chaque point propagé.
- ajout d'un nœud en chacun de ces points, création d'une nouvelle surface (ajout d'éléments linéaires pour le nouveau front de fissure et d'éléments quadrangles pour la nouvelle surface) et processus de maillage de cette surface créée.
- traitement spécifique des nœuds du bord qui, selon les cas, peuvent « sortir » de la structure. Ils sont alors déplacés sur les frontières de la structure.

Pour les modélisations 2D, la méthode est même plus simple. Seul un élément linéaire doit être ajouté à chaque pas de propagation et aucune correction des points du bord n'est nécessaire.

7.2 Algorithme

L'algorithme mis en place pour la propagation par la méthode maillage est le suivant (comme décrit dans (bib70) et sur la figure 7.2-a). A l'itération i :

- récupération du maillage de la fissure,
- récupération des points extrémités du front de fissure,
- correction du maillage de la fissure :
 - on modifie les coordonnées des deux nœuds extrémités du front de fissure pour qu'ils correspondent aux points extrémité du front de fissure,
 - on modifie les coordonnées de tous les autres nœuds du fond pour qu'ils soient équidistribués entre les deux extrémités (nombre de nœuds du fond $NbNf$).
- boucle sur les nœuds du fond : $1 \leq k \leq NbNf$
 - calcul de $\Delta G(k)$ par interpolation linéaire,
 - calcul de l'angle de bifurcation β à partir de $K_I(k)$ et $K_{II}(k)$,
 - calcul du vecteur de propagation par interpolation linéaire :
$$\Delta a(k) = C(\Delta G(k))^{m/2} \text{ si } C \text{ et } m \text{ sont renseignés,}$$
$$\Delta a(k) = DA_{MAX}(\Delta G(k)/\max_k(\Delta G(k)))^{m/2} \text{ si } DA_{MAX} \text{ et } m \text{ sont renseignés,}$$
 - calcul de la position du propagé du nœud N_k ,
 - ajout d'un nouveau nœud.
- fin de boucle,
- boucle entre 1 et $NbNf - 1$
 - ajout d'une nouvelle maille SEG2 entre chacun des nœuds du fond et le nœud suivant,

- ajout d'une maille QUA4 entre deux nœuds successifs du fond à l'itération $i-1$ et leurs propagés,
- en 2D : ajout d'une maille POI pour le fond et SEG2 entre le fond précédent et le nouveau fond.
- fin de boucle,
- création des mailles *FOND* (nouveau fond) et *LEVRE* (lèvre précédente + nouvelles mailles surfaciques),
- réactualisation du maillage.

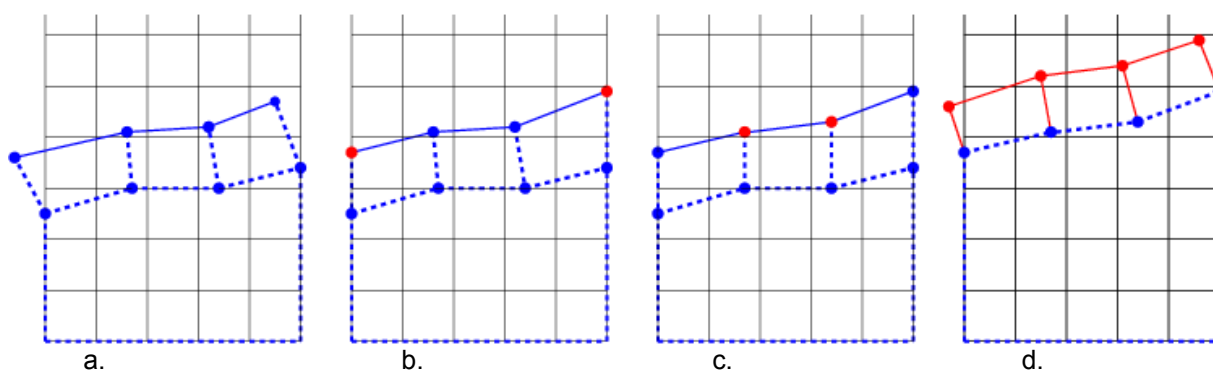


Figure 7.2-a: a. maillage en entrée de la routine, b. étape de correction des points du bord, c. étape d'équirépartition de tous les autres nœuds du front, d. propagation du front discrétisé et création des nouvelles mailles (SEG2 et QUA4).

8 Séparation et fusion des fronts de fissure

Différentes méthodes ont été présentées pour suivre la propagation d'une fissure avec X-FEM (ie pour mettre à jour les valeurs des level sets à chaque pas de propagation) :

- la méthode maillage (ou par projection), voir paragraphe 56 ,
- les méthodes de résolution directe des équations d'évolution d'Hamilton-Jacobi (simplexe ou upwind), voir paragraphes 15 et 24 ,
- la méthode géométrique, voir paragraphe 52 .

Il s'agit d'expliquer ici les modifications à apporter à ces différentes méthodes pour rendre compte du comportement d'un front de fissure lorsqu'il rencontre un obstacle (un trou par exemple). Comment gérer la séparation de ce front de fissure en deux, voire en plusieurs autres morceaux ? De même qu'en est-il de la fusion de deux fronts de fissure indépendants en un seul ? Ce sont les level sets qui déterminent le fond de fissure et donc la présence ou non de plusieurs fonds de fissure. Mais il s'agit dans certains cas de procéder à des améliorations des méthodes précédentes pour qu'elles s'adaptent à la séparation du front de fissure (modification de la création du maillage de la fissure pour la méthode maillage, continuité des level sets dans le trou pour les méthodes simplexe et upwind avec utilisation d'une grille auxiliaire).

Selon les méthodes utilisées pour la gestion de la mise à jour des level sets, la stratégie mise en place pour rendre compte de la séparation ou de la fusion de fronts de fissure diffère.

8.1 Méthode maillage

Pour la méthode maillage, une première option est d'étendre ce qui est fait au paragraphe 56 au cadre de plusieurs fonds de fissure. Les level-sets donnent des informations sur le nombre et la position des extrémités des fonds de fissure (intersection entre le maillage de la structure et l'iso-zéro des deux level-sets). Tout en conservant les nœuds qui constituent les extrémités déjà existantes à l'instant $i - 1$, les nœuds constituant les nouvelles extrémités sont sélectionnés (figure 8.1-a a.).

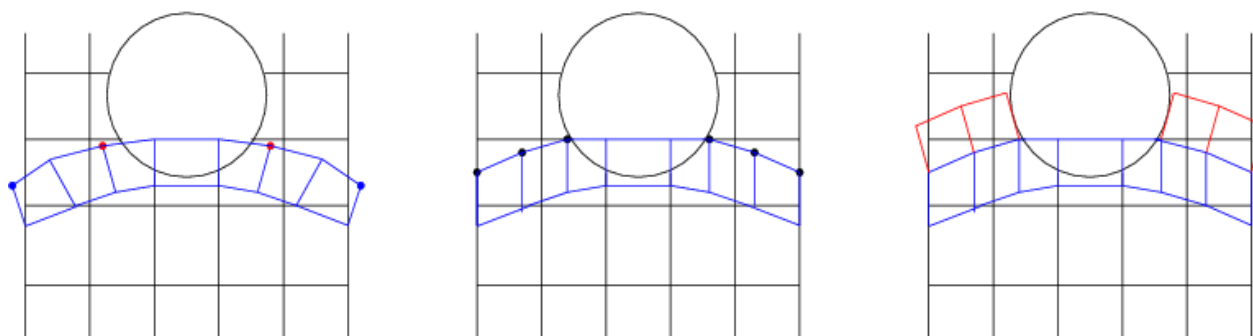


Figure 8.1-a: a. nœuds déterminant les extrémités des fonds de fissure (en bleu : nœuds sélectionnés pour les extrémités déjà existantes, en rouge : nœuds sélectionnés pour les nouvelles extrémités), b. repositionnement des nœuds sélectionnés sur les différents fonds, c. calcul des nœuds propagés et formation des nouvelles mailles.

Comme au paragraphe 56 une procédure de repositionnement des nœuds des différents fonds (figure 8.1-a b.) est effectuée. Il ne reste plus qu'à calculer les nœuds propagés et former les mailles (figure 8.1-a c.).

Bien que cette méthode fonctionne correctement, elle présente un désavantage : la perte de nœuds à la séparation de deux fonds. Les mailles peuvent de plus se déformer de façon importante lors de la phase de repositionnement des nœuds extérieurs à la structure. Ceci peut notamment conduire à une mauvaise discrétisation de la fissure en propagation hors-plan.

Pour pallier ces difficultés, l'idée a été d'augmenter le nombre de nœuds sur chaque fond de fissure. Reste la question de la création de ces nœuds et du maillage. En remarquant que les mailles représentant la fissure ne servent qu'à effectuer des projections, il est possible de créer une rangée de mailles pour l'avancée de fissures non conforme à la rangée de mailles précédente (figure 8.1-b).

De cette manière, on peut augmenter le nombre de nœuds entre deux pas de propagation sans modifier la discrétisation induite par les vitesses de propagation.

L'algorithme se résume donc comme suit : recherche des nœuds les plus proches des extrémités du fond, repositionnement sur le bord, création de nouveaux nœuds sur le fond actuel positionnés de façon équidistante les uns des autres, calcul des nœuds du nouveau fond. Comme il y a le même nombre de nœuds entre les deux fonds, des quadrangles respectant les surfaces du fond sont créés.

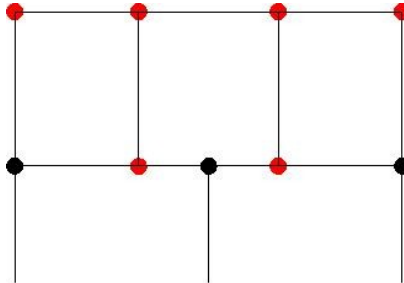


Figure 8.1-b: maillage non conforme (en rouge; nœuds créés pour la nouvelle rangée de mailles).

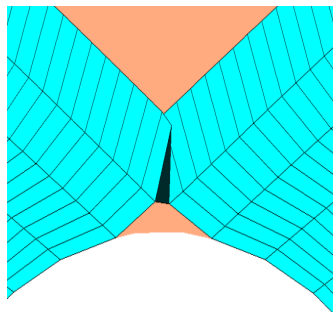
Le nombre total N de nœuds sur tous les fonds de fissure doit rester constant entre deux pas de propagation. Pour cela, on répartit les N_i nœuds par fond proportionnellement au rapport entre la longueur curviligne s_i du fond i et la longueur curviligne totale des n fonds :

$$N_i = E \left(\frac{s_i}{\sum_{j=1}^n s_j} N \right)$$

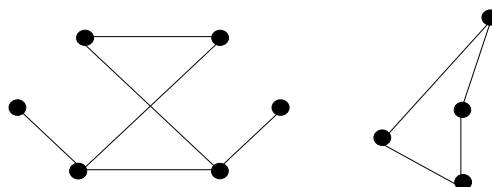
où $E(x)$ représente la partie entière de x . Comme il peut y avoir des cas pour lesquels la somme des N_i est inférieure à N , on distribue les nœuds restants aux fonds ayant la plus grande partie décimale.

Remarque

La méthode maillage peut être problématique quand la fissure est concave. Ce cas peut être notamment rencontré lors de la réunion de deux fonds de fissure (figure 8.1-c a.) et est absolument à éviter car des problèmes importants sont créés par le calcul des level sets (valeurs fausses, apparition de faux fonds de fissure).



a.



b.

Figure 8.1-c: cas problématiques : a. front de fissure concave lors de la réunion de deux fonds de fissures, b. création de mailles pathologiques.

La création de mailles problématiques (concaves par exemple) peut s'avérer être une source de difficultés (voir figure 8.1-c b.). Elle peut apparaître lorsque la vitesse de propagation pour les

nœuds du bord est imposée tangente à la surface. Cela peut également créer des erreurs dans le calcul de la level set tangente et impliquer la création de faux fonds de fissure. Pour éviter la concavité d'une maille dans le cas d'un avancement maximal D_{max} de la fissure trop important, on diminue la valeur de ce dernier pour le calcul suivant. Soit le quadrangle représenté dans le plan (ABC) figure 8.1-d (si le quadrangle n'est pas plan alors le point D représente la projection du nœud propagé à partir de C sur le plan (ABC)). On calcule les coordonnées du point M d'intersection entre (AB) et (CD) et on compare les distances entre AM et AB , CM et CD . On pose $\left(\vec{E}_1 = \frac{\vec{AB}}{\|\vec{AB}\|}, \vec{E}_2 = \frac{\vec{AC} - (\vec{AC} \cdot \vec{E}_1) \vec{E}_1}{\|\vec{AC} - (\vec{AC} \cdot \vec{E}_1) \vec{E}_1\|} \right)$ une base orthonormée, et $\vec{AD}(d_1, d_2)$, $\vec{AC}(c_1, c_2)$ et $\vec{AM}(x, 0)$ dans le repère $(A, \vec{E}_1, \vec{E}_2)$. Par définition de M ,

$$\det(\vec{CM}, \vec{CD}) = \begin{vmatrix} x - c_1 & d_1 - c_1 \\ -c_2 & d_2 - c_2 \end{vmatrix} = 0.$$

$$\text{On en déduit } x = \frac{d_2 c_1 - d_1 c_2}{d_2 - c_2} = \frac{(\vec{AD} \cdot \vec{E}_2)(\vec{AC} \cdot \vec{E}_1) - (\vec{AD} \cdot \vec{E}_1)(\vec{AC} \cdot \vec{E}_2)}{\vec{CD} \cdot \vec{E}_2}.$$

On introduit le coefficient $Crit = \min\left(\frac{AM}{AB}, \frac{CM}{CD}\right)$ et l'avancement maximal rectifié D_{max}' vérifie :

$$\text{si } Crit < 1.2 \quad \text{alors } D_{max}' = \frac{D_{max} \cdot Crit}{1.2}$$

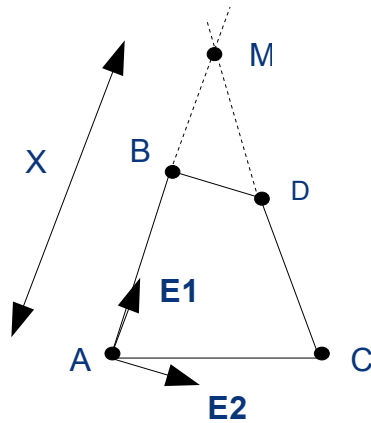


Figure 8.1-d: représentation d'une maille pour laquelle on vérifie que l'avancement n'est pas trop grand.

Le coefficient 1,2 est un coefficient de sécurité afin d'assurer une propagation non nulle ($Crit=1$ conduisant à un point M confondu avec le(s) point(s) B et/ou D). Bien qu'avec une telle stratégie, l'avancement maximal risque de tendre vers 0, elle permet cependant de résoudre une partie des cas problématiques.

8.2 Méthode upwind et simplexe: introduction d'un fond virtuel

Comme remarqué dans le paragraphe 26, l'utilisation de la méthode upwind s'accompagne très généralement de l'utilisation d'une grille auxiliaire (dont le maillage est régulier) sur laquelle s'effectue

la mise à jour des level sets qui sont par la suite projetées sur le maillage physique de la pièce (généralement beaucoup moins régulier). Cependant une partie de la grille peut être située à l'extérieur du maillage de la pièce, comme par exemple dans le cas d'une structure contenant un trou.

Les nœuds de la grille situés à l'extérieur du maillage de la pièce sont alors projetés sur le même point du front de fissure lors du calcul de la vitesse de propagation des level sets. Tous ces nœuds se voient donc attribuer les mêmes vitesses et angles de propagation ainsi que les mêmes bases locales (comme décrit au paragraphe 6). Il en résulte l'apparition d'une discontinuité des level sets sur la grille au niveau du bord libre de la pièce (voir figure 8.2-a).

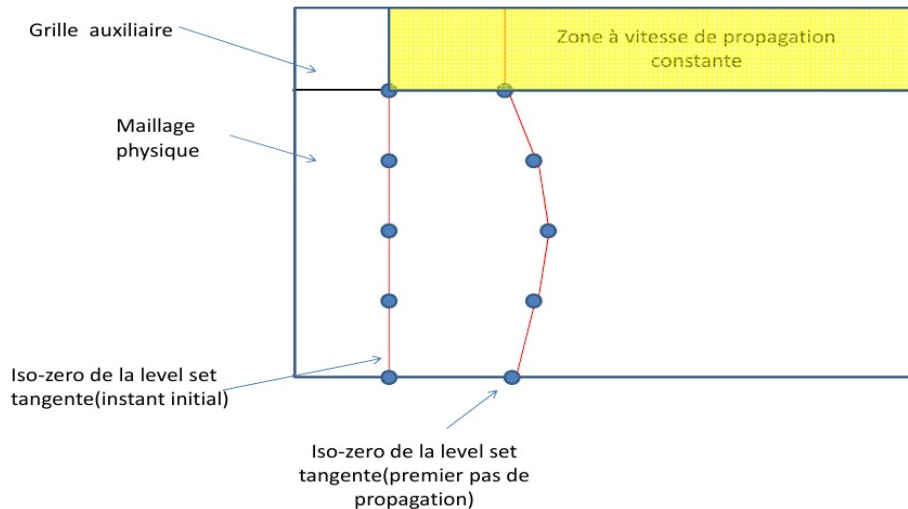


Figure 8.2-a : exemple de discontinuité des level sets sur la grille au niveau du bord libre.

Cette discontinuité, bien que située hors du maillage physique, implique une mauvaise définition du gradient des level sets sur le bord libre et donc une mauvaise définition de la base locale lors du pas de propagation suivant. Elle peut également influencer sur la définition des level sets à l'intérieur du maillage, notamment lors de la propagation de fissures dans des pièces ayant une section irrégulière et/ou des trous. Pour éviter ces problèmes et définir de manière cohérente les level sets sur la grille au voisinage du maillage physique, une interpolation du front de fissure à l'extérieur du maillage par une droite tangente au front de fissure est proposée. Pour cela, le dernier segment du front de fissure est prolongé par un segment de droite à l'extrémité duquel est introduit un nouveau nœud du front de fissure, qualifié de virtuel. Ce point est alors considéré comme un point du front de fissure. La même base locale que celle du dernier point du front de fissure physique lui est attribuée. La vitesse et l'angle de propagation sont calculés par interpolation linéaire à partir des vitesses et angles de propagation des deux derniers points (figure 8.2-b).

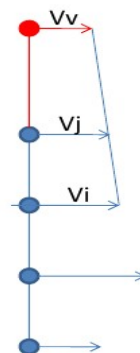


Figure 8.2-b : représentation de l'approximation linéaire de la vitesse au point virtuel V_v (en rouge) à partir des vitesses des deux derniers points du front de fissure V_i et V_j (en bleu).

Pour créer les points du fond virtuel, on procède de la manière suivante (figure 8.2-c) :

- on pose P_i et P_j deux points situés à l'extrémité du fond de fissure et d la distance du point virtuel P_v au dernier point du fond de fissure P_j ,
- on détermine la droite prolongeant le dernier segment du front de fissure au point virtuel : $\vec{P}_{jv} = d \cdot \vec{b}_j$ où $\vec{P}_{jv}$ est le vecteur reliant P_j à P_v et $\vec{b}_j$ est le vecteur unitaire tangent au front de fissure au point P_j ,
- on calcule les dérivées spatiales de la vitesse et de l'angle de propagation sur le dernier segment du fond virtuel : $\frac{dV}{ds} = \frac{V_j - V_i}{\|\vec{P}_{ij}\|}$, $\frac{d\beta}{ds} = \frac{\beta_j - \beta_i}{\|\vec{P}_{ij}\|}$
- on en déduit la vitesse et l'angle de propagation au fond virtuel :

$$V_v = V_j + d \cdot \frac{dV}{ds}, \quad \beta_v = \beta_j + d \cdot \frac{d\beta}{ds}$$

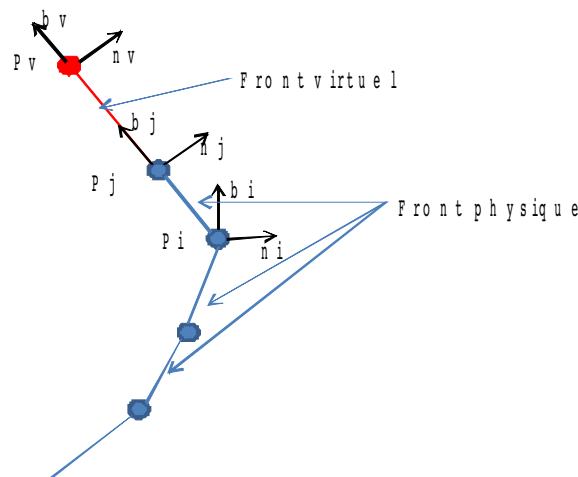


Figure 8.2-c : création et emplacement du front de fissure virtuel.

On place le point virtuel à une distance d du front physique égale au rayon du tore utilisé pour la propagation des level sets (voir paragraphe 46). On choisit cette distance de façon à obtenir des level sets lissées sur l'ensemble de la zone de propagation. Si la localisation n'est pas utilisée, le point virtuel est placé à une distance de 7 fois l'avancement maximal de la fissure. Pour des vitesses et des angles de propagation (définis sur les points virtuels) trop importants ou trop faibles, on rapproche le point virtuel du front physique de manière à satisfaire les conditions imposées. On note par exemple que l'on a une importante discontinuité des level sets si l'angle de propagation est supérieur (resp. inférieur) à 90 degrés (resp. -90 degrés). On note également que si la distance du point virtuel au front de fissure physique est inférieure au rayon de localisation, on peut conserver une discontinuité des gradients. Néanmoins, cette discontinuité est située loin du maillage de la pièce. Elle n'influe donc pas sur la définition de la base locale sur le bord libre.

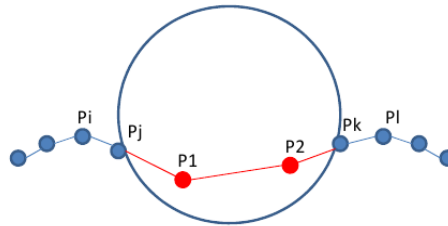


Figure 8.2-d: création d'un front virtuel (en rouge) à l'intérieur d'un trou circulaire. P_1 est créé par prolongement de $[P_i P_j]$ et P_2 par prolongement de $[P_l P_k]$. Le dernier segment du fond virtuel relie P_1 et P_2 et connecte les fronts de fissure.

La méthode du fond virtuel peut alors être immédiatement étendue au cas de trous. Si la pièce présente un trou, il convient en effet également de bien définir les level sets à l'intérieur de celui-ci. On procède alors comme décrit précédemment en définissant un front de fissure virtuel à l'intérieur du trou par la création de deux points virtuels P_1 et P_2 , un pour chaque fond de fissure délimité par le trou (figure 8.2-d). Deux des trois segments du front virtuel sont des extensions des deux fronts de fissures physiques. Le troisième segment relie quant à lui les deux points virtuels afin de connecter les fronts de fissure. Il ne sert pas à interpoler les level sets mais à assurer leur continuité dans le trou. Les deux points virtuels ayant des vitesses de propagation différentes, deux nœuds voisins situés au centre du trou peuvent être projetés sur deux nœuds du fond de fissure différents et avoir des vitesses différentes.

Les angles et vitesses de propagation ainsi que la base locale sur le segment $[P_1 P_2]$ sont calculés par interpolation linéaire (comme décrit au paragraphe 8) à partir des valeurs attribuées aux deux points virtuels.

Dans le cas de la création d'un fond virtuel dans un trou, on doit s'assurer qu'il n'y a pas de croisement des fonds de fissure. La distance d'un point virtuel au fond de fissure ne doit pas dépasser le tiers de la distance entre les deux fonds $d \leq \frac{\|P_k - P_j\|}{3}$.

De plus, la vitesse de propagation au point virtuel V_v ne doit pas être trop petite ou trop importante. On impose qu'elle soit au moins égale aux deux tiers de la vitesse V_j du dernier fond de fissure et qu'elle ne doive pas dépasser quatre tiers de cette valeur. Ainsi, on impose que $\frac{2V_j}{3} \leq V_v \leq \frac{4V_j}{3}$.

Si $V_v < \frac{2V_j}{3}$ (resp. $V_v > \frac{4V_j}{3}$), alors on impose $V_v = \frac{2V_j}{3}$ (resp. $V_v = \frac{4V_j}{3}$) et on en

déduit l'expression de la distance d dans ces deux cas : $d = \left| \frac{V_j}{3 \frac{dV}{ds}} \right|$.

Néanmoins, la création d'un fond virtuel n'est pas suffisante. Les valeurs des level sets définies à l'intérieur du trou, issues de la projection sur le segment milieu du fond virtuel, peuvent dans certains cas influencer la définition des level sets dans le maillage. De plus, l'approximation de la vitesse est plus précise si les level sets sont initialement droites et si l'intersection des deux iso-zéros est confondue avec le front virtuel. Dans le but de garantir le redressement des level sets dans le trou lors de la fusion des fronts de fissure après le passage de celui-ci, on recalcule les level sets pour les nœuds projetés sur un segment virtuel du front de fissure, et ce avant la propagation des level sets. On calcule les level sets de manière géométrique (voir paragraphe 52) en projetant le nœud sur le front de fissure (figure 8.2-e). On pose XM un nœud du maillage. On définit alors son projeté XN sur le fond de fissure, les vecteurs $\vec{ip}$ et $\vec{np}$ définissant la base locale en front de fissure en XN .

On obtient les level sets normale et tangente aux nœuds XM simplement par :
 $lsn(XM) = \overrightarrow{XM} \cdot \overrightarrow{XN} \cdot \vec{n}_p$ et $lst(XM) = \overrightarrow{XM} \cdot \overrightarrow{XN} \cdot \vec{t}_p$

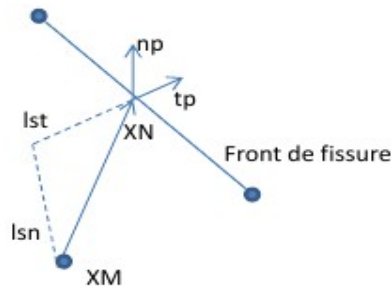


Figure 8.2-e: Calcul des levels sets aux nœuds projetés sur le fond de fissure.

Ce calcul des level sets permet de corriger l'influence de la définition non physique des level sets au centre du trou lors de la fusion des deux fonds de fissure. Une fois le fond virtuel créé et les level sets corrigées, le calcul des vitesses de propagation par projection sur le front de fissure s'effectue comme décrit au paragraphe 6, sans faire de distinction entre le front physique et le fond virtuel.

Remarque

L'introduction d'un fond virtuel nécessite une stratégie de vérification de la numérotation des fronts de fissure ainsi que de leur sens de parcours ce qui n'était jusqu'alors pas nécessaire : la connexion des fronts de fissure par un fond virtuel ne les rend plus indépendants les uns des autres. La méthode suivante a été adoptée pour vérifier si les nœuds des différents fronts de fissure sont bien numérotés (figure 8.2-f) :

a) on construit un nouveau front de fissure sur la grille auxiliaire qui par définition de la grille n'est constitué que d'un seul morceau avec un unique sens de parcours. On projette ensuite les extrémités E_1^i et E_2^i de chacun des morceaux i du front de fissure réelle sur le front de fissure continu associé à la grille. Ces extrémités sont repérées sur le front de fissure de la grille par leur abscisse curviligne respective. A titre d'exemple, aux extrémités E_1^i et E_2^i correspondent les abscisses curvilignes $s_{P_1^i}$ et $s_{P_2^i}$ (où P_1^i et P_2^i sont respectivement les projetés de E_1^i et E_2^i sur le front de fissure associé à la grille auxiliaire) ;

b) la séquence des abscisses curvilignes ainsi obtenue permet une renumérotation des morceaux des fronts de la fissure réelle ;

c) on évalue le produit scalaire $(E_2^i E_1^i) \cdot (P_2^i P_1^i)$ pour vérifier que chacun des morceaux est parcouru dans le même sens que le front de fissure sur la grille. Dans le cas contraire, on réoriente le(s) morceau(x) mal orienté(s).

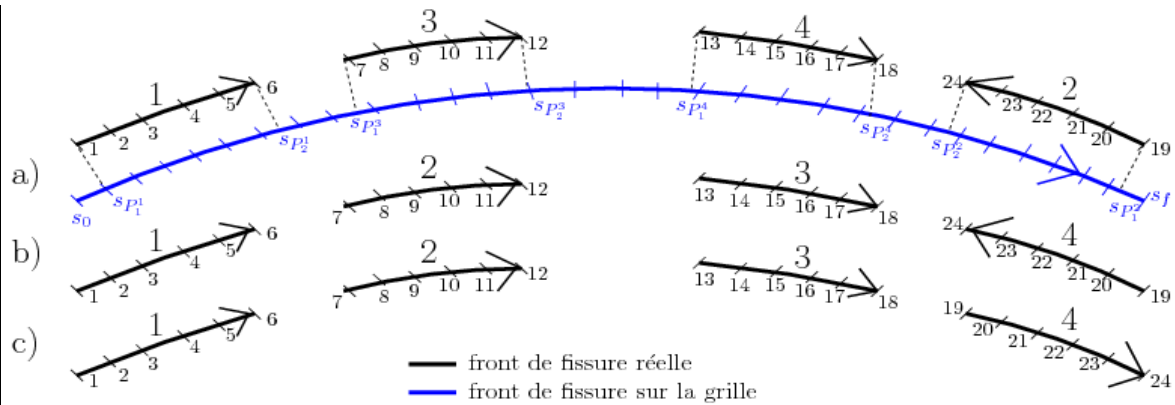


Figure 8.2-f: a. exemple de mauvaise numérotation et orientation du front de fissure réelle (noire) et front de fissure sur la grille (bleu) , b. renumérotation des morceaux du front de fissure réelle, c. réorientation des morceaux du front de fissure réelle

8.3 Méthode géométrique

La méthode géométrique ne nécessitant aucune grille auxiliaire, sa gestion de la séparation et de la fusion de fronts de fissure est automatique et ne demande aucune modification de la procédure décrite au paragraphe 52 .

9 Routines informatiques :

•op0010.f :

Opérateur correspondant à la commande `PROPA_XFEM`. Assure l'enchaînement des principales étapes du processus de propagation:
Calcul du nombre de cycles de fatigue pour obtenir l'avance maximale, donné par l'utilisateur, de la fissure qui propage le plus vite et récupération des données G et β de la fissure à propager (`xprcyc`)
Calcul des `cham_no_s` des vitesses de propagation (appel à la routine `xprvit`)
Ajustement de V_N (`xpraju`)
Calcul de la condition CFL pour les phases de réinitialisation et de réorthogonalisation des level sets (`xprcfl`)
Propagation des level sets (`xprls`)
Réinitialisation de `lsn` (`xprini`, `xprcnu`, `xprrei`, `xprupw`)
Réorthogonalisation de `lst` (`xprreo`, `xprupw`)
Réinitialisation de `lst` (`xprrei`, `xprupw`)
Enrichissement de la structure de données `fiss_xfem` (`xajuls`, `xprmil`, `xnrch3`, `xbaslo`, `xlagsp`)

•xprvit.f :

Xfem PROpagation : calcul des champs de **VIT**esse de propagation.
Récupération des données G et β de la fissure à propager, calcul en chaque point du fond de fissure des vitesses normale et tangentielle V_N et V_T , et projection orthogonale des valeurs à tous les nœuds du maillage.

•xprcfl.f :

Xfem PROpagation : calcul des conditions **CFL**.
Calcule la longueur de la plus petite arête du maillage en utilisant l'option de calcul `CFL_XFEM`. Cette valeur représente la valeur de la condition CFL pour les phases de réinitialisation et réorthogonalisation.

•xpraju.f :

Xfem PROpagation : **AJU**stement de la vitesse normale.
Ajustement du champ de vitesse normale d'après l'équation explicitée au paragraphe [§2.5].

•xprls.f :

Xfem PROpagation des **Level Sets**.
Calcule les level sets résultantes d'une avancée selon les champs de vitesses calculés par un schéma de Runge-Kutta d'ordre 2 (paragraphe [§3]).

•xprini.f :

Xfem PROpagation : **INI**tialisation des paramètres nécessaires aux routines `xprrei` et `xprreo`
Calcul des paramètres $|T|$ et $\vec{n}_i$ nécessaires aux routines de réinitialisation et réorthogonalisation avec la méthode « simplexe » par l'appel à l'option de calcul `XFEM_SMPLEX_INIT` (ou duplique les structures de données calculées lors d'une propagation antérieure). Repère les nœuds sommets afin de les distinguer des nœuds milieux (non physiques), et les nœuds sur lesquelles il faudra estimer les résidus locaux (tore autour du fond de fissure).

•xprrei.f :

Xfem PROpagation : **REI**nitialisation d'une level set par la méthode simplexe.
Calcule géométriquement les level sets sur les nœuds des éléments coupés par l'iso-zéro de la level set à réinitialiser (appel à `xprls0`). Réinitialise cette level set en cherchant à atteindre l'état stationnaire de l'équation de réinitialisation exprimé en fonction d'un temps virtuel (paragraphe [§4.2]) par la méthode « simplexe » avec les option de calcul `MOY_NOEU_S` et `XFEM_SMPLEX_CALC`.

•xprls0.f :

Xfem PROpagation : Calcul géométrique des level sets sur les éléments coupés par (**LS=0**).

Calcule géométriquement les level sets sur les nœuds des éléments coupés par l'iso-zéro d'une level set.

•xprreo.f:

Xfem Propagation : **RE**Orthogonalisation d'une level set par la méthode simplexe.
Réorthogonalise une level set par rapport à une autre en recherchant l'état stationnaire de l'équation de réorthogonalisation (paragraphe [§4.2]) par la méthode « simplexe » avec l'option de calcul XFEM_SMPLX_CALC.

•xprupw.f:

Xfem Propagation : réinitialisation et réorthogonalisation des level sets avec la méthode **UP**Wind
Réinitialise une level set ou réorthogonalise une level set par rapport à une autre en recherchant l'état stationnaire des équations du paragraphe [§26] par la méthode « upwind ».

•xprcnu.f:

Xfem Propagation : création de la table de Connexion des Nœuds pour la méthode Upwind
Calcule la base locale de la grille utilisée par la méthode «upwind» et arrange les nœuds selon la direction des axes de cette base (paragraphe [§26]).

•xprpfi.f:

Xfem Propagation: calcul de la **P**rojection d'un point sur la surface de la **F**issure
Projette un point P sur l'iso-zéro de la level set normale ou tangentielle et calcule la valeur des deux level set au point P . Cette routine est utilisée par la méthode upwind pour affecter les conditions aux limites aux points problématiques (paragraphe [§33]).

•xprali.f:

Xfem Propagation: **A**rete Libre
L'iso-zéro des deux level sets est représentée et stockée comme une liste de triangles. Cette routine détermine si l'arête de l'un de ces triangles est sur la surface libre du modèle. Cela est utilisé par la méthode upwind pour affecter correctement les conditions aux limites aux points problématiques (paragraphe [§33]).

•xprpls.f:

Xfem Propagation **P**rojection des **L**evel **S**ets: projection des level sets sur le maillage physique
Projette les level sets sur le maillage physique à partir de la grille auxiliaire et calcule le gradients des level sets projetées sur le maillage physique.

•xprcyc.f:

Xfem Propagation: calcule du nombre de **C**YCles de fatigue
Sélectionne la fissure qui propage le plus vite parmi les fissures du modèle et calcule le nombre de cycles de fatigue qui sont nécessaires pour avoir l'avance maximale donnée par l'utilisateur. En plus on lit les valeurs de G et β dans les tableaux issus par CALC_G et on stocke celles de la fissure à propager. Ces valeurs sont récupérées et utilisées par *xprvit.f*. Les paramètres affectant la vitesse de propagation des fissures (TEST_MAIL, NB_POINT_FOND et CHAR_LINE) sont gérés aussi par cette routine fortran.

•xprdis.f:

Xfem Propagation: **D**ISTance maximale entre deux fonds de fissure
Calcule la distance maximale entre deux fonds de fissure du modèle. Cette routine est utilisée pour vérifier si le maillage physique et/ou level sets est suffisamment raffiné pour bien représenter les level sets (voir opérande TEST_MAIL dans la documentation [U4.82.11]).

•te0118.f:

Option de calcul « CFL_XFEM » : Calcul des longueurs caractéristiques par éléments.

Option de calcul « MOY_NOEU_S » : Calcul de la moyenne d'un champ nodal sur les nœuds sommets (comme un *cnsces* ignorant les nœuds milieux).

Option de calcul « XFEM_SMPLX_INIT » : Calcul des quantités nécessaires à la méthode simplexe constantes par rapport au temps virtuel τ (i.e. $|T|$ et $\vec{n}_i$).

Option de calcul « XFEM_SMPLX_CALC » : Calcul des quantités nécessaires à la méthode simplexe variables par rapport au temps virtuel τ (i.e. $\delta\varphi_T$ et α_i^T).

• xprgeo.f:

Xfem PROpagation: calcul des level set par la méthode GÉOmétrique

Routine qui recalcule les level sets après propagation par la méthode géométrique (méthode de calcul directe des level sets).

• xprvir.f:

Xfem PROpagation: création du fond de fissure **VIR**tuel

Cette routine est utilisée dans le cas de l'utilisation de la méthode upwind (ou simplexe) avec utilisation d'une grille auxiliaire afin de résoudre les problèmes liés à la projection des level sets pour les nœuds extérieurs au maillage de la pièce (typiquement une structure contenant un trou).

• xprfon.f:

X fem PROpagation: renumérotation du **FON**d virtuel s'il est mal orienté

Cette routine permet de vérifier (et, le cas échéant, de modifier) la numérotation des nœuds du front de fissure dans le cas de la création d'un fond virtuel (créé par la routine xprvir.f). Elle est utilisée uniquement dans le cadre des méthodes upwind et simplexe avec utilisation d'une grille auxiliaire.

10 Bibliographie

- 1 GRAVOUIL A., MOËS N., BELYTSCHKO T., "Non-planar 3D crack growth by the extended finite element and level sets - Part II: Level set update", International Journal for Numerical Methods in Engineering, vol. 53, pp. 2569-2586, 2002
- 2 OSHER S., SETHIAN J. A., "Fronts propagations with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations", Journal of Computational Physics, vol. 79, pp. 12-49, 1988
- 3 SUKUMAR N., CHOPP D.L., MORAN B., "Extended finite element method and fast marching method for three-dimensional fatigue crack propagation", Engineering Fracture Mechanics, vol. 70, pp. 29-48, 2003
- 4 ERDOGAN G., SIH G.C., "On the crack extension in plates under plane loading and transverse shear", Journal of Basic Engineering, vol. 85, pp. 519-27, 1963
- 5 BARTH T.J., SETHIAN J.A., "Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains", Journal of Computational Physics, vol. 145, pp. 1-40, 1998
- 6 CRANDALL M.G., LIONS P.L., "Two approximations of solutions of Hamilton-Jacobi equations", Mathematics of Computation, vol. 43, pp. 1-19, 1984
- 7 DECONINCK H., STRUIJS R., ROE P.L., "Compact Advection Schemes on Unstructured Grids", Technical Report, VKI, VKI LS 1993-04, Computational Fluid Dynamics, 1993
- 8 ROE P.L., "Linear Advection Schemes on Triangular Meshes", Technical Report CoA 8720, Cranfield Institute of Technology, 1987
- 9 ROE P.L., "Optimum Upwind Advection on Triangular Mesh", ICASE 90-75, 1990
- 10 PENG D., MERRIMAN B., OSHER S., ZHAO H., KANG M., "A PDE-based fast local level set method", Journal of Computational Physics, vol. 155, pp. 410-438, 1999
- 11 ADALSTEINSSON D., SETHIAN J.A., "A fast level set method for propagating interfaces", Journal of Computational Physics, vol. 118, pp. 269-277, 1995
- 12 PRABEL B., COMBESCURE A., GRAVOUIL A., MARIE S., "Level set X-FEM non matching meshes: application to dynamic crack propagation in elastic-plastic media", International Journal for Numerical Methods in Engineering, vol. 1, pp. 1-15, 2006
- 13 DUFLOT M., "A study of the representation of cracks with level sets", International Journal for numerical methods in engineering, vol. 70, pp. 1261-1302, 2007
- 14 COLOMBO, PATRICK MASSIN, « Fast and robust level set update for 3D non-planar X-FEM crack propagation modelling », Computer Methods in Applied Mechanics and Engineering, 200 (2011), 2160-2180
- 15 SUKUMAR N., CHOPP D.L., BÉCHET E., MOËS N., « Three-dimensional non-planar crack growth by a coupled extended finite element and fast marching method », International Journal Numerical Methods in Engineering, 76 (2008), 727-748
- 16 SHI J., CHOPP D., LUA J., SUKUMAR N., BELYTSCHKO T., « Abaqus implementation of extended finite element method using a level set representation of three-dimensional fatigue crack growth and life predictions », Engineering Fracture Mechanics, 77 (2010), 2840-2863
- 17 CITARELLA R., BUCHHOLZ F.-G., Comparison of crack growth simulation by DBEM and FEM for SEN-specimens undergoing torsion or bending loading, Engineering Fracture Mechanics, 75 (2008), 489-509

- 18 COLOMBO D., « An implicit geometrical approach to level sets update for 3D non planar X-FEM crack propagation », submitted to Computer Methods in Applied Mechanics and Engineering
- 19 GALENNE E., « Propagation automatique de fissures 3D avec X-FEM en fatigue : méthode de projection » Compte rendu AMA, 2008

11 Description des versions

Indice document	Version Aster	Auteur(s) Organisme(s)	Description des modifications
A	11	D. COLOMBO (University of Manchester), M. GUITON, M. SIAVELIS (IFPen), S. GENIAUT, V.X. TRAN EDF/R&D AMA, P.MASSIN EDF/R&D LaMSID	Texte initial