
eXtended Finite Element Method : Généralités

Résumé :

Ce document présente la méthode X-FEM (eXtended Finite Element Method) qui permet principalement de considérer des fissures ne respectant pas le maillage pour traiter les problèmes de fissures 2D et 3D. La fissure est définie par la commande `DEFI_FISS_XFEM` [U4.82.08] et est utilisable pour des calculs en statique linéaire et non-linéaire.

Ce document présente les généralités autour de X-FEM , ainsi que la technique utilisée pour représentée les fissures (méthode des Level Sets). D'autres documents dédiés à des problématiques spécifiques sont disponibles :

- contact-frottement sur les lèvres de la fissure en petits glissements avec X-FEM [R5.03.54],
- contact-frottement sur les lèvres de la fissure en grands glissements avec X-FEM [R5.03.53] ,
- propagation de fissures avec X-FEM [R7.02.13]

Table des Matières

1	Introduction.....	3
2	Représentation de la fissure par des « level sets ».....	5
2.1	Aspect théorique des level sets.....	5
2.2	Calcul des level sets.....	7
2.3	Base locale au fond de fissure.....	16
2.4	Détermination du fond de fissure.....	16
3	Problème de fissuration avec X-FEM.....	24
3.1	Problème général.....	24
3.2	Enrichissement de l'approximation du déplacement.....	25
3.3	Sous-découpage.....	38
3.4	Intégration de la rigidité.....	46
3.5	Intégration des seconds membres surfaciques.....	50
4	Thermique linéaire avec X-FEM.....	52
4.1	Introduction.....	52
4.2	Restrictions.....	52
4.3	Problème traité.....	53
4.4	Approximation X-FEM du champ de température.....	54
4.5	Intégration des matrices et des vecteurs élémentaires.....	55
5	Calcul énergétique des Facteurs d'Intensité des Contraintes.....	57
5.1	Méthode G-thêta pour le calcul de G.....	57
5.2	Méthode G-thêta pour le calcul de KI, KII et KIII avec les level sets.....	60
5.3	Méthode G-thêta avec X-FEM.....	66
6	Bibliographie.....	67
7	Description des versions.....	71

1 Introduction

La simulation numérique en mécanique de la rupture est essentiellement basée sur l'utilisation de la méthode des éléments finis (MEF). Classiquement pour un problème de propagation de fissures, un premier calcul est réalisé sur un maillage initial, puis un nouveau maillage est déterminé, prenant en compte l'avancée de la fissure (suivant la loi de propagation choisie). Un nouveau calcul s'en découle, et le procédé est itéré pour chaque pas de propagation.

Un inconvénient majeur et immédiat est la nécessité de remailler à chaque pas de propagation. Le processus de remaillage peut être facilement automatisé en 2D [bib67], et dans certains cas en 3D [bib67], mais un remaillage 3D de qualité s'avère coûteux en temps (supervision humaine) et en argent. En effet avec un mailleur automatique, un raffinement local approprié au niveau de la zone de fissuration entraîne souvent un nombre excessif d'éléments partout sur le reste de la structure. Un processus de raffinement par couches est généralement utile, comme l'introduction d'un tore en fond de fissure, solution performante (facilitant aussi la mise en place d'éléments de Barsoum [bib67]) mais nécessitant une intervention humaine coûteuse ; et ce d'autant plus que la forme géométrique de la fissure est très complexe (fissures hélicoïdales par exemple [bib67]). Le problème devient quasi-inconcevable en multi-fissuration 3D. En plus de ces difficultés pratiques, la projection de grandeurs (contraintes, variables internes) d'un maillage à l'autre pose des problèmes théoriques fondamentaux (vérification des équations de conservation de l'énergie, de la quantité de mouvement, de la masse) [bib67].

Parallèlement aux difficultés liées à la propagation, les méthodes avec maillage s'avèrent peu efficaces pour des études paramétriques où l'on s'intéresse à l'influence de la position et de la forme de la fissure.

Les méthodes dites « Meshless » [bib67] ont été proposées pour s'affranchir des contraintes liées au maillage. Les meshless sont basées sur une discrétisation uniquement nodale, sans connectivité, et les fonctions de forme sont construites à partir de la configuration nodale. Initialement introduites à la fin des années 70' pour les problèmes sans frontières (méthode Smoothed Particle Hydrodynamics dans le domaine de l'astrophysique), les méthodes Meshless ont été étendues par la suite aux problèmes mécaniques et se déclinent aujourd'hui sous diverses variantes : méthodes MLS (Moving Least Square), Diffuse Element Method (DEM), Element Free Galerkin (EFG), Reproducing Kernel Particle Method (RKPM), hp clouds, et bien d'autres. L'élément commun de toutes ces méthodes est la notion de Partition de l'Unité, qui est un ensemble de fonctions dont la somme est égale à un en chaque point du domaine considéré. Le principal inconvénient de ces méthodes est qu'elles requièrent un effort numérique plus important que celles basées sur un maillage (temps CPU notamment). En particulier, l'évaluation des fonctions de forme est loin d'être aussi triviale, les schémas d'intégration numériques sont généralement plus riches donc plus coûteux et le système global d'équations résultant a une largeur de bande supérieure comparé à une MEF [bib67]. Même dans les récentes versions des méthodes Meshless combinant des supports physiques et mathématiques [bib67], l'imposition de conditions aux limites reste problématique.

D'autres méthodes se basent sur une partition de l'unité dans le cadre des éléments finis standards, où seule la définition des fonctions de formes diffère. Ce choix de partition de l'unité évite les problèmes d'intégration de la rigidité, considérablement chère pour les méthodes Meshless (EFG, DEM, RKPM notamment). De plus, l'utilisation d'une partition de l'unité basée sur les éléments finis permet une implémentation facile des conditions aux limites de type Dirichlet (contrairement aux techniques utilisant une partition de l'unité basée sur les moindres carrés). Ces méthodes, que l'on retrouve d'abord dans leurs versions maillées sous le nom de Partition of Unity Finite Element Method (PUFEM) ou de Generalized Finite Element Method (GFEM) [bib67], permettent d'enrichir facilement l'espace des fonctions de forme, grâce à la connaissance *a priori* des propriétés de la solution du problème. Combinant la GFEM et la *hp-cloud*, Duarte *et al.* [bib67] propose une partition de l'unité mixte (éléments finis et Shepard), qui permet de considérer une fissure non maillée, avec des fonctions de formes enrichies.

Encore plus proche du cadre éléments finis classique, la méthode des éléments finis étendus (X-FEM) a suscité un des plus vifs intérêt si on s'en réfère à l'évolution du nombre de publications sur ce sujet depuis son apparition et à la place qui lui est réservée dans les conférences internationales. Cette méthode utilise la partition de l'unité pour enrichir la base des fonctions de forme afin de représenter

un saut du champ de déplacement au niveau des lèvres de la fissure, ainsi que la singularité en fond de fissure. Deux enrichissements sont alors introduits : un enrichissement par une fonction saut qui permet de gérer la discontinuité au travers des lèvres de la fissure, et un enrichissement par des fonctions asymptotiques, qui permet une représentation fidèle des phénomènes physiques ayant lieu au niveau du fond de fissure. Historiquement, les signes précurseurs sont dus à Belytschko et Black [bib67], qui appliquent la partition de l'unité [bib68] à la mécanique de la rupture en incorporant les formules analytiques des champs asymptotiques dans l'approximation du déplacement. L'ajout de la fonction Heaviside généralisée [bib68] permet d'écrire l'approximation enrichie du déplacement sous sa forme finale, expression qui donne naissance à la méthode des éléments finis étendus. Comparée à GFEM, elle offre moins de dépendance vis-à-vis de la connaissance de la forme des solutions, ce qui offre une plus grande flexibilité [bib67]. Le champ d'application de X-FEM ne cesse de s'élargir, cette approche ayant été utilisée dans des cadres très variés : plaques de Reissner-Mindlin, coques rupture en 3D, multi-fissuration, zones cohésives, modélisation de trous et de bi-matériaux, formulations incompressibles et grandes transformations, nucléation des fissures, la fissuration sous contact, rupture dynamique, plasticité...

De plus, l'utilisation de la méthode des level sets couplée avec X-FEM a grandement facilité le traitement des fissures en 2D ([bib67], [bib67] et [bib68]) et en 3D [bib68]. Initialement introduite dans le cadre de la mécanique des fluides pour représenter l'évolution d'interfaces, la méthode des level sets considère l'interface comme l'iso-zéro d'une fonction distance. Cette méthode s'avère particulièrement efficace pour la propagation d'une fissure en 3D [bib67], couplée avec l'utilisation de la Fast Marching Method [bib67].

Ce document s'articule autour de 4 sections, dont cette introduction qui tient lieu de section 3.

La section 2 est consacrée à l'utilisation de la méthode des level sets pour la fissuration. Après un bref rappel théorique, on précise le calcul des level sets, qui s'avèrent pratiques pour déterminer la position de la fissure en 3D et du fond de fissure, et qui permettent de définir une base locale en fond de fissure.

La section 3 présente le problème de fissuration traité avec X-FEM. On introduit au paragraphe [§3.2] l'approximation du déplacement écrit dans une base de fonctions de forme enrichie. L'ajout de fonctions discontinues au travers de l'interface conduit à une procédure de sous-découpage détaillée au paragraphe [§3.3], préalable à la phase d'intégration des termes de rigidité et du second membre, dont la mise en place est expliquée au paragraphe [§3.4].

La section 5 s'intéresse au post-traitement du taux de restitution d'énergie et des facteurs d'intensité de contraintes en mécanique de la rupture linéaire. La méthode G -thêta permet de calculer le taux de restitution d'énergie local. Pour cela, un champ thêta est introduit, représentant une extension virtuelle de la fissure. Le G local est alors solution d'une équation variationnelle, utilisant l'intégrale J sous forme d'intégrales de domaine. Les choix de discrétisations de G et du champ thêta amènent à un système linéaire, dont la résolution conduit aux valeurs de G le long du fond de fissure [bib69]. La méthode G -thêta permet aussi de déterminer les facteurs d'intensité de contraintes le long du fond de fissure. Au lieu d'utiliser l'intégrale J , on utilise la forme bilinéaire de G , qui conduit à des intégrales de domaine mêlant champs solutions et champs asymptotiques analytiques (appelées aussi intégrales d'interaction). Ainsi, de même que pour G , les K sont alors solutions d'une équation variationnelle, utilisant des intégrales d'interaction. Les choix de discrétisations des K et du champ thêta amènent à un système linéaire, dont la résolution conduit aux valeurs des K le long du fond de fissure [bib69].

2 Représentation de la fissure par des « level sets »

La méthode des « Level sets » a été introduite dans le cadre de la mécanique des fluides pour représenter l'évolution d'interfaces. L'idée principale est de considérer l'interface comme l'iso-zéro d'une fonction distance. Le choix de la fonction distance importe peu ici, car seule la connaissance de l'iso-zéro est utile et importante.

2.1 Aspect théorique des level sets

Soit une interface Γ délimitant un ouvert Ω de \mathfrak{R}^n . L'idée est de définir une fonction $\varphi(x, t)$ régulière (au moins Lipchitzienne) telle que le sous-espace $\varphi(x, t) = 0$ représente l'interface.

La level set a les propriétés suivantes :

$$\begin{aligned} \varphi(x, t) > 0 & \text{ pour } x \in \Omega \\ \varphi(x, t) < 0 & \text{ pour } x \notin \overline{\Omega} \\ \varphi(x, t) = 0 & \text{ pour } x \in \partial\Omega = \Gamma(t). \end{aligned}$$

Cette méthode s'applique aisément aux problèmes de fissuration 2D, notamment dans le cadre des approches où la fissure n'est pas maillée. ([bib67] [bib67] en 2D). L'extension est possible pour le traitement des fissures en 3D.

Ainsi, dans le cas de la fissuration, il est nécessaire d'introduire deux level sets ([bib68] en 2D et [bib68] en 3D) :

- une level set normale (l_{sn}) qui représente la distance à la surface de la fissure (surface étendue par prolongement à tout le domaine),
- une level set tangente (l_{st}) qui représente la distance au fond de fissure.

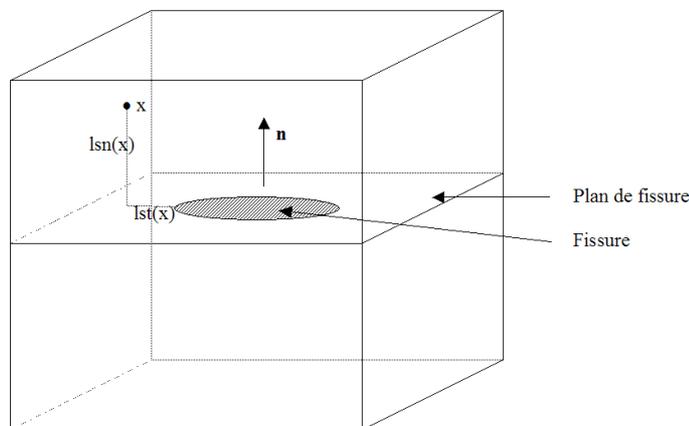


Figure 2.1-1 : Level sets et distance à la fissure

L'iso-zéro de la level set normale définit la surface de la fissure, étendue par continuité à tout le domaine. L'intersection des iso-zéros des deux level sets définit le fond de fissure. De plus, le signe de la level set tangente est choisi de telle sorte que la surface de la fissure Γ_{cr} corresponde à l'espace engendré par $(l_{sn} = 0) \cap (l_{st} < 0)$. Le signe de la level set normale est choisi arbitrairement grâce à la convention d'orientation de la normale au plan de fissure, définie explicitement au paragraphe [§2.2.1]. Les points x pour lesquels $l_{sn}(x)$ est négatif sont dits « au-dessous » de la fissure, et ceux pour lesquels $l_{sn}(x)$ est positif sont dits « au-dessus » de la fissure (voir Figure 2.1-2 et Figure 2.1-3).

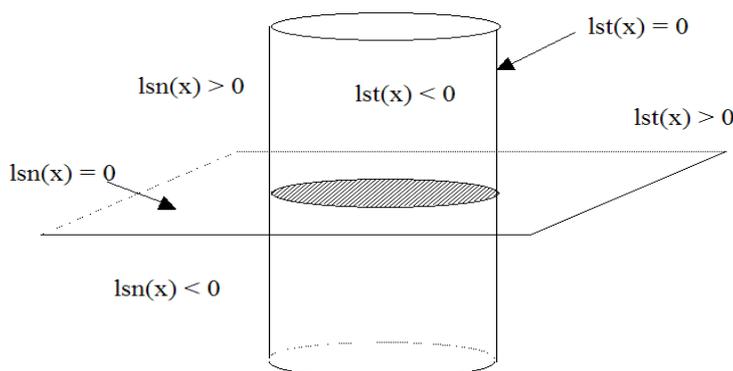


Figure 2.1-2 : Level sets pour la représentation d'une fissure 3D

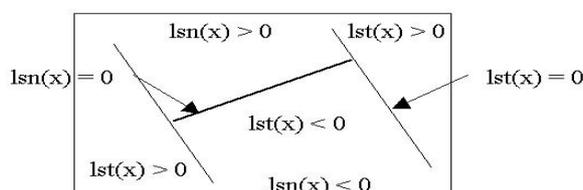


Figure 2.1-3 : Level sets pour la représentation d'une fissure 2D

La phase de propagation de la fissure se traduit simplement par la propagation des level sets. La propagation d'une level set nécessite trois étapes successives [bib67] :

- extension de la vitesse connue sur l'iso-zéro 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.

La propagation d'une fissure représentée par 2 level sets présente quelques particularités. Le vecteur vitesse est connu seulement sur le front de fissure, c'est-à-dire une courbe. La fissure ne peut que grandir, pas se déplacer. Deux fonctions level sets doivent être propagées et on souhaite que leurs gradients restent orthogonaux. L'enchaînement des étapes peut être résumer comme ceci :

- propagation de lsn et réinitialisation de lsn ,
- propagation de lst ,
- orthogonalisation du gradient de lst versus gradient de lsn ,
- réinitialisation de lst .

Ces étapes se ramènent toutes à la résolution d'équation de type Halimton-Jacobi [bib67] :

$$\frac{\partial \dots}{\partial t} + F |\nabla \dots| = f$$

où F et f sont des fonctions qui dépendent de l'étape.

La Fast Marching Method [bib67] est une technique alternative bien adaptée à la propagation strictement monotone de fronts. Cette méthode sépare les nœuds du maillage suivant leur éloignement de l'interface, et à chaque itération l'équation de propagation est résolue pour les seuls nœuds immédiatement adjacents à l'interface, en utilisant un schéma de différences finies d'ordre 2.

Cette partie est plus détaillée dans le document [R7.02.13].

2.2 Calcul des level sets

Le calcul des level sets (champs scalaires) est effectué pour chaque fissure. Il peut se faire de deux manières. Soit par la donnée de leurs expressions analytiques et dans ce cas, une simple évaluation de ces fonctions aux nœuds du maillage fournit les champs scalaires recherchés. Soit la fissure est maillée et dans ce cas, il est nécessaire de donner des mailles surfaciques correspondants à une lèvres (`GROUP_MA_FISS`) et les mailles linéiques correspondants au fond de fissure (`GROUP_MA_FOND`). Dans le cas 2D, on donnera des mailles linéiques (pour `GROUP_MA_FISS`) et des mailles points (pour `GROUP_MA_FOND`). Les distances sont ensuite calculées par un algorithme de projections orthogonales inspiré de celui utilisé pour le contact [bib68], explicité aux paragraphes [§2.2.1] et [§2.2.2].

2.2.1 Calcul de la level set normale (lsn)

Pour chaque nœud du maillage, on cherche la maille de `GROUP_MA_FISS` la plus proche de ce nœud. Pour cela, on utilise les algorithmes de projection sur un triangle (voir le paragraphe [§2.3.2] de [bib68]) et sur un quadrangle (voir le paragraphe [§2.3.3] de [bib68]). La valeur de *lsn* est alors la distance normale de ce point à la maille.

2.2.1.1 Calcul de la level set normale en 3D

En 3D, il faut faire attention au sens de la normale. En effet, les mailles de `GROUP_MA_FISS` étant intérieures à la structure, ce ne sont pas des mailles de bord, et l'orientation automatique des normales (mot-clé `ORIE_PEAU_3D` de l'opérateur `MODI_MAILLAGE`) est alors impossible. Pour s'assurer de toujours choisir le même sens pour les normales, on prend la normale de la première maille de `GROUP_MA_FISS` comme référence et on « propage » la direction de cette normale à toutes les autres mailles du groupe pour chacun des éléments adjacents. Pour chaque maille on stocke un *indicateur d'orientation* de la maille qui vaut « +1 » si l'orientation de sa normale est cohérente avec celle de la maille de référence construite de proche en proche et « -1 » sinon. Cela permet d'affecter le bon signe de la valeur de la level set normale en multipliant sa valeur non corrigée par l'indicateur d'orientation (voir algorithme de calcul de *lsn* en 3D ci-dessous).

Algorithme de calcul de l'*indicateur d'orientation* de la maille:

- création du vecteur I « indicateur d'orientation » et du vecteur C contenant le numéro de couche à laquelle chaque élément de `GROUP_MA_FISS` appartient. La taille des deux vecteurs est égale au nombre d'éléments n de `GROUP_MA_FISS`
- on met à zéro tous les éléments des deux vecteurs I et C , sauf le premier élément auquel on affecte la valeur +1
- boucle i sur le numéro de la couche (vecteur C), de 1 à n
 - boucle j sur les éléments du vecteur C , de 1 à n
 - on récupère le numéro de la couche à laquelle l'élément j appartient:
 $n_{couche} = C(j)$
 - si $n_{couche} = i$, c'est-à-dire l'élément j appartient à la couche courante i :
 - on calcule la normale \vec{n}_j à l'élément j (on calcule la normale au triangle formé par les trois premiers nœuds définissant l'élément)
 - boucle n_{el} sur les éléments qui ont au moins un nœud en commun avec l'élément j
 - si l'élément n_{el} n'appartient à aucune couche ($C(n_{el})=0$), on lui assigne le numéro de la couche courante+1 ($C(n_{el})=i+1$), on en calcule la normale $\vec{n}_{n_{el}}$ et l'indicateur d'orientation:
 $I(n_{el}) = \text{sign}(I(j) \cdot \vec{n}_{n_{el}} \cdot \vec{n}_j)$
 - fin boucle
 - fin si
 - fin boucle
- fin boucle

Algorithme de calcul de lsn en 3D :

- boucle sur tous les nœuds P du maillage
 - initialisation de $dmin$
 - boucle sur les mailles triangulaires de `GROUP_MA_FISS` (avec subdivision des quadrangles en triangles)
 - soient A , B et C les sommets du triangle
 - calcul de la normale à la maille : $\vec{N} = \vec{AB} \wedge \vec{AC}$
 - calcul de M , projeté de P dans le plan ABC
 - si M en dehors du triangle ABC , on ramène M sur une des droites (AB) (AC) (BC)
 - si M encore en dehors du triangle ABC , on ramène M en A B ou C
 - on récupère l'indicateur d'orientation I de la normale de la maille
 - si $PM < dmin$ alors $dmin = PM$ et $lsn(P) = I \cdot \vec{PM} \cdot \vec{N}$
 - fin boucle
- fin boucle

La subdivision d'un quadrangle en triangles permet de se ramener à un calcul approché de projection dans un cas linéaire (car les fonctions de forme du quadrangle sont bilinéaires alors que celles du triangle sont linéaires). De plus, il existe 2 façons de réaliser un tel découpage (suivant la diagonale choisie). On engendre donc tous les cas possibles, comme il est fait au paragraphe [§2.3.3] de [bib68].

Le calcul de la projection dans le cadre de la recherche d'appariement des nœuds de contact pour la méthode continue n'utilise pas ce genre d'algorithme, mais résout le problème non-linéaire de

projection sur un quadrangle par la méthode de Newton. On pourrait s'en inspirer pour la calcul des level sets par projection.

2.2.1.2 Calcul de la level set normale en 2D

En 2D, on commence par réorganiser les mailles de `GROUP_MA_FISS` pour avoir une série de mailles contiguës. Deux vecteurs vont être stockés, la liste des mailles ordonnées et la liste de leur orientation, qu'on appellera ici `LIMA` et `ORI`.

Algorithme de tri de `GROUP_MA_FISS` :

Initialisation de la première maille `M`, première maille de `GROUP_MA_FISS`. On a $LIMA(1)=M$ et $ORI(1)=1$. On initialise le booléen `FINFIS` qui teste si on est en fin de fissure, `FINFIS` initialisé à `FALSE`.

```

•boucle de recherche de la maille suivante  $LIMA(k)$  tant que  $FINFIS = FALSE$ 
  si  $ORI(k-1)=1$ , calcul de  $N$  nœud 2 de  $LIMA(k-1)$ 
  si  $ORI(k-1)=0$ , calcul de  $N$  nœud 1 de  $LIMA(k-1)$ 
   $FINFIS = TRUE$ 
  •boucle sur les mailles  $M$  de GROUP_MA_FISS, tant que  $FINFIS = TRUE$ 
    Calcul du nœud 1 et du nœud 2 de  $M$ ,  $N1$  et  $N2$ 
    si  $N1=N$ ,  $ORI(k)=1$  et  $LIMA(k)=M$ ,  $FINFIS = FALSE$ 
    si  $N2=N$ ,  $ORI(k)=0$  et  $LIMA(k)=M$ ,  $FINFIS = FALSE$ 
    (sortie de boucle quand on trouve une maille suivante)
  •fin boucle
  (sortie de boucle si on n'a pas trouvé de maille suivante car on est en fin de fissure)
•fin de boucle
    
```

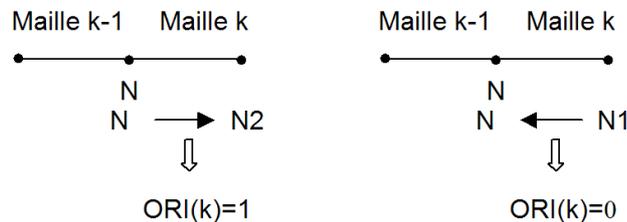


Figure 2.2.1.2-1 : Orientation de la maille suivante

La première maille identifiée pouvant être quelconque n'importe quelle maille à l'intérieur de la fissure, on effectue un décalage de la liste trouvée du nombre total de maille `NBMAF` moins le nombre de mailles trouvées k de façon à avoir une liste qui termine sur la maille de fin de fissure trouvée. On obtient donc un vecteur avec des composantes non affectées en tête qui représentent les mailles non encore répertoriées et une partie remplie sur la fin du vecteur correspondant aux mailles orientés précédemment stockées.

```

•boucle de recherche de la maille précédente  $LIMA(k)$  tant que  $FINFIS = FALSE$  ( $k$ 
  décroissant)
  si  $ORI(k+1)=1$ , calcul de  $N$  nœud 1 de  $LIMA(k+1)$ 
  si  $ORI(k+1)=0$ , calcul de  $N$  nœud 2 de  $LIMA(k+1)$ 
   $FINFIS = TRUE$ 
  •boucle sur les mailles  $M$  de GROUP_MA_FISS, tant que  $FINFIS = TRUE$ 
    Calcul du nœud 1 et du nœud 2 de  $M$ ,  $N1$  et  $N2$ 
    Si  $N1=N$ ,  $ORI(k)=0$  et  $LIMA(k)=M$ ,  $FINFIS = FALSE$ 
    Si  $N2=N$ ,  $ORI(k)=1$  et  $LIMA(k)=M$ ,  $FINFIS = FALSE$ 
    (sortie de boucle quand on trouve une maille précédente)
  •fin boucle
  (sortie de boucle si on n'a pas trouvé de maille précédente)
•fin de boucle
    
```

- fin boucle
(sortie de boucle si on n'a pas trouvé de maille précédente, car on est en fin de fissure)
- fin de boucle

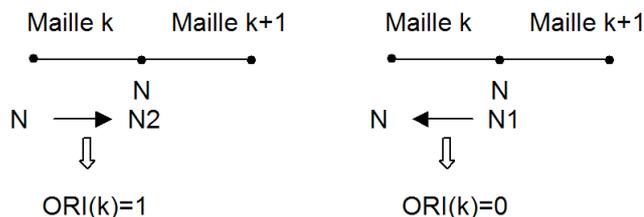


Figure 2.2.1.2-2 : Orientation de la maille précédente

Algorithme de calcul de lsn en 2D :

- boucle sur tous les nœuds P du maillage
 - initialisation de $dmin$
 - boucle sur les mailles linéiques de LIMA
 - soient A , B les sommets du segment si $ORI=1$ et B , A si $ORI=0$
 - calcul de M , projeté de P sur la droite AB
 - si M en dehors du segment AB , on ramène M en A ou B
 - si $PM < dmin$
 - $dmin = PM$ et
 - $lsn = PM_p \cdot \text{sign}(\vec{AB}, \vec{AP})$
 - où M_p est le projeté de P sur AB (éventuellement en dehors de AB)
 - si $ORI=0$, $lsn = -lsn$
 - fin si
 - fin boucle
 - fin boucle

2.2.2 Calcul de la level set tangente (l_{st})

Pour chaque nœud du maillage, on cherche la maille de `GROUP_MA_FOND` la plus proche de ce nœud. Pour cela, on utilise l'algorithme de projection sur un segment (voir le paragraphe [§2.3.1] de [bib68]). La valeur de *l_{st}* est alors la distance normale de ce point au segment.

De même que précédemment, la détermination de la normale au segment n'est pas évidente a priori. Pour la calculer, il faut d'abord trouver la maille surfacique de `GROUP_MA_FISS` qui le borde.

Algorithme de calcul de *l_{st}* en 3D :

```

•boucle sur tous les nœuds P du maillage
  initialisation de dmin
  •boucle sur les segments de GROUP_MA_FOND
    soit A et B les deux extrémités du segment
    •boucle sur toutes les mailles de GROUP_MA_FISS
      •si A et B appartiennent à cette maille alors
        soit C un nœud de la maille autre que A et B
        calcul de la normale à cette maille
        (càd la normale au plan de fissure) :  $\vec{N} = \vec{AB} \wedge \vec{AC}$ 
        calcul de la normale au fond de fissure dans le plan de la fissure :
           $\vec{N}' = \vec{AB} \wedge \vec{N}$ 
        Vérification du sens de  $\vec{N}'$ 
        Soit M la projection de P sur (AB)
        Si M est en dehors de [AB], on le ramène en A ou B
        Si  $PM < dmin$  alors  $dmin = PM$  et  $l_{st}(P) = \vec{PM} \cdot \vec{N}'$ 
      •fin si
    •fin boucle
  •fin boucle
•fin boucle

```

Remarque :

Lors de la projection de *P* sur les segments *AB* du fond de fissure, il est possible que tous les projetés *M* se retrouvent hors du segment considéré. Dans ce cas, le fait de ramener *M* sur le bord n'est pas un critère suffisant pour déterminer le segment le plus proche (voir Figure 2.2.2-2). On propose de choisir le segment le plus proche comme celui où on a le moins rabattu le projeté. Avec les notations ci-dessus, si on appelle *M'* le point rabattu (point *B* sur la Figure 2.2.2-2), alors on cherche le segment pour lequel l'angle $\alpha = \angle MPM'$ est le plus petit. On note que dans un cas sans rabattement (quand le projeté tombe dans le segment), cet angle est nul. Ce test supplémentaire est introduit pour le choix du segment le plus proche.

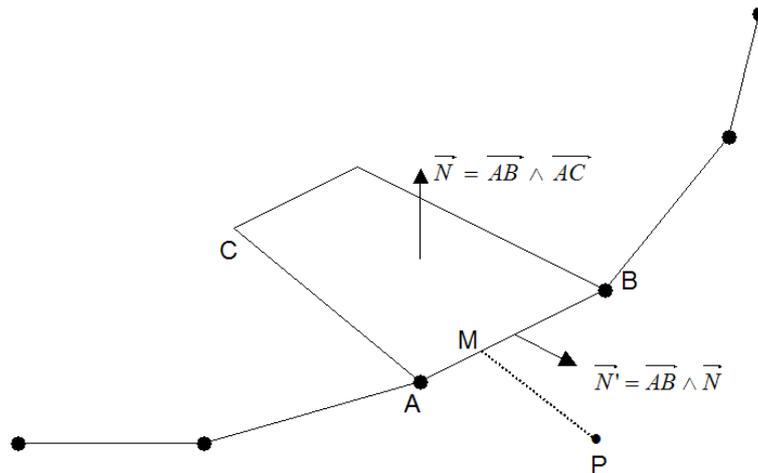


Figure 2.2.2-1 : Calcul de la level set tangente

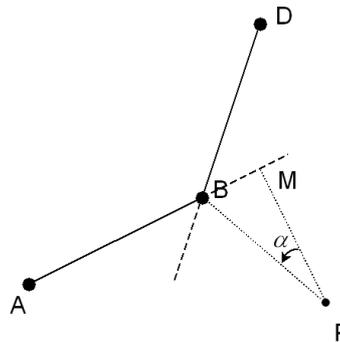


Figure 2.2.2-2 : Critère supplémentaire pour le choix du segment le plus proche

Algorithme de calcul de lst en 2D:

```

•boucle sur tous les nœuds  $P$  du maillage
  initialisation de  $dmin$ 
  •boucle sur les mailles point A de GROUP_MA_FOND
    •boucle sur toutes les mailles de GROUP_MA_FISS
      •si  $A$  appartient à cette maille alors
        soit  $B$  le deuxième nœud de la maille
        soit  $M$  la projection de  $P$  sur  $(AB)$ 
        calcul de  $\varepsilon$  tel que  $\vec{AM} = \varepsilon \times \vec{AB}$ 
        si  $AM < dmin$  alors  $dmin = AM$  et  $lst = -\varepsilon \times AB$ 
      •fin si
    •fin boucle
  •fin boucle
•fin boucle
  
```

2.2.3 Approximation des level sets

Quelque soit la méthode de calcul utilisée (par des fonctions analytiques ou par projection), les champs des level sets sont interpolés par les fonctions de forme linéaires utilisées pour l'approximation du champ de déplacement [bib68] :

$$l_{sn}(x) = \sum_i \phi_i(x) l_{sn_i}$$

$$l_{st}(x) = \sum_i \phi_i(x) l_{st_i}$$

où ϕ_i sont les fonctions de forme linéaires classiques et l_{sn_i} et l_{st_i} les valeurs nodales des champs level sets. Sur un tétraèdre à 4 nœuds la surface de la fissure sera donc un plan, mais sur un hexaèdre à 8 nœuds, elle pourra être légèrement courbe.

L'erreur de discrétisation des level sets est directement liée à la taille du maillage et à la courbure de la level set. Examinons l'exemple suivant, mettant en jeu une fissure courbe en 2D :

Soit la level set est l'ellipse d'équation :

$$x^2 + \left(\frac{y}{0.5}\right)^2 = 1$$

La distance à l'ellipse est calculée en chacun des 4 nœuds du quadrangle défini par $(x, y) \in [0.4, 0.9] \times [0, 0.5]$. Soit P un point donné de l'espace de coordonnées (x_p, y_p) . Soit H sa projection sur l'ellipse précédemment définie. H a pour coordonnées $(a \cos \theta, b \sin \theta)$. L'équation de la droite (HP) est la suivante :

$$y = \frac{a \sin \theta}{b \cos \theta} x + b \sin \theta \left(1 - \frac{a^2}{b^2}\right)$$

On résout numériquement cette équation d'inconnue θ pour chaque nœud du quadrangle et on en déduit la distance à l'ellipse :

$$dist = \sqrt{(x_p - a \cos \theta)^2 + (y_p - b \sin \theta)^2}$$

L'approximation de la level set s'écrit alors sur ce quadrangle :

$$l_{sn}^h(x, y) = -0.44347 \phi_1(x, y) - 0.1 \phi_2(x, y) + 0.22227 \phi_3(x, y) + 0.040806 \phi_4(x, y)$$

où ϕ_i $i=1,4$ sont les fonctions de forme associées aux nœuds du quadrangle. Ces fonctions s'exprime à l'aide des N_i , fonctions de forme classiques sur le quadrangle de référence, et des changements de variables entre les coordonnées réelles (x, y) et les coordonnées de référence (s, t) .

$$\phi_1(x, y) = N_1(s(x, y), t(x, y)) = \frac{1}{4}(1-s)(1-t)$$

$$\phi_2(x, y) = N_2(s(x, y), t(x, y)) = \frac{1}{4}(1+s)(1-t)$$

$$\phi_3(x, y) = N_3(s(x, y), t(x, y)) = \frac{1}{4}(1+s)(1+t)$$

$$\phi_4(x, y) = N_4(s(x, y), t(x, y)) = \frac{1}{4}(1-s)(1+t)$$

avec les changements de variables suivants :

$$s(x, y) = 4x - 2.6$$

$$t(x, y) = 4y - 1$$

Sur la Figure 2.2.3-1, on a représenté en pointillé la projections sur l'ellipse de chaque nœud . On observe que l'iso-zéro de la level set interpolée avec les fonctions de forme du quadrangle est assez éloignée de l'ellipse initiale.

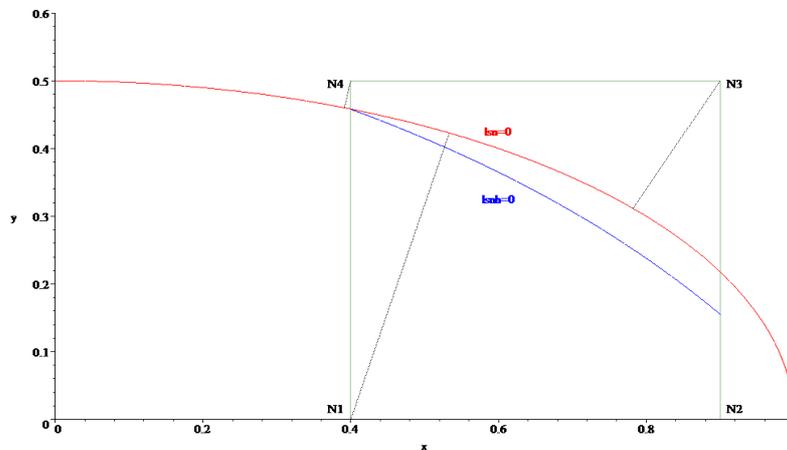


Figure 2.2.3-1 : Erreur de discrétisation de la level set

2.2.4 Réajustement des level sets

Afin de limiter les problèmes d'intégration lorsque la fissure passe « près » d'un nœud, une procédure de réajustement de la level set normale est mise en place. Si sur une arête du maillage, l_{sn} s'annule trop « près » d'un nœud extrémité, la valeur de l_{sn} en ce nœud est mise à zéro. Le critère utilisé pour l'instant est 1% de la longueur de l'arête. Cette valeur est celle utilisée par le logiciel développé par l'équipe de Nicolas Moës au GeM (voir le paragraphe [§1.3.2.3] de [bib68]).

Le réajustement de la level set du nœud milieu n'est pas systématique sur une arête dont les level set normales des extrémités ont été mises à zéro. Le critère est fixé pour le moment à 0.0001 (il fera l'objet d'une recherche plus approfondie car on peut montrer que lors du traitement du contact avec des éléments quadratiques, l'algorithme de réajustement (avec différents critères) et la suppression des ddl en trop sont insuffisants à l'obtention d'un bon conditionnement de la matrice).

Algorithme de réajustement de la level set normale :

```

•boucle sur les mailles
  •boucle sur les arêtes de la maille, d'extrémités A et B
    si lsn(A) ≠ lsn(B)
      
$$d = \frac{lsn(A)}{lsn(A) - lsn(B)}$$

      si |d| ≤ 0.01 alors lsn(A) = 0 fin si
      si |d - 1| ≤ 0.01 alors lsn(B) = 0 fin si
    fin si
    si élément quadratique
      si lsn(A) = 0 et lsn(B) = 0 alors
        
$$d = lsn(M)$$

        si |d| ≤ 0.0001 alors lsn(M) = 0 fin si
      fin si
    fin si
  fin boucle
fin boucle

```

2.2.5 Notion de vraie distance signée

Si les level sets sont calculées par des fonctions analytiques, le choix des fonctions n'est pas unique pour une même géométrie de fissure. En effet, n'importe quelle fonction à valeurs positives d'un côté et à valeurs négatives de l'autre est valable. Cependant, si on désire que la level set représente la vraie distance signée, alors le choix est unique. Cette notion est importante par la suite lorsque l'on définit les coordonnées dans la base locale au fond de fissure à l'aide des valeurs des level sets. Il faut donc veiller à donner la formule exacte de la distance à la fissure (expression qui n'est pas aisée pour des géométries de fissures complexes).

Un des intérêts de la méthode par projection est qu'elle fournit un champ qui est la vraie distance signée. On pourrait donc aussi envisager de combiner les deux méthodes. La donnée d'une fonction analytique simple permettrait dans un premier temps de déterminer l'iso-zéro ; puis de créer un maillage simplifié de cette surface, qui servirait de support à la méthode par projection. L'inconvénient est de créer un maillage 2D virtuel de la fissure qui est disjoint du maillage réel 3D.

Sinon, on pourrait aussi envisager une phase d'orthogonalisation qui transforme une level set quelconque en une vraie distance [bib68].

2.2.6 Multi-fissuration

Les level sets sont définies pour chaque fissure par l'opérateur `DEFI_FISS_XFEM`. La liste des fissures est nécessaire pour la création des éléments finis X-FEM (opérateur `MODI_MODELE_XFEM`). Pendant cette phase, des champs « concaténés » sont créés. Par exemple, on crée un champ de level set normale global à toutes les fissures. Pour chaque nœud (champs aux nœuds) ou chaque maille (champs par élément), on va chercher l'information associée à la fissure la plus proche.

Les champs aux nœuds « concaténés » créés sont :

- Level set normale,
- Level set tangente,
- Statut des nœuds (voir §3.2.5.1),
- Base locale au fond de fissure (voir §2.3).

Les champs par éléments « concaténés » créés sont :

- Ceux liés au sous-découpage (voir § 3.3),

Ceux liés aux structures de données pour le contact (voir document R5.03.54).

Pour ce qui concerne la propagation [R7.02.13], on peut définir plusieurs fissures sur le modèle et on peut donner la liste des fissures qui se propagent: à chaque appel de l'opérateur `PROPA_FISS`, toutes les fissures données se propage.

Restrictions :

Les fissures doivent être suffisamment espacées l'une de l'autre (3 mailles sans fissures doivent au moins les séparer). *A fortiori*, les fissures ne doivent pas non plus se croiser. Sinon, l'introduction d'enrichissements spéciaux est nécessaire [bib68].

2.3 Base locale au fond de fissure

Les gradients des level sets peuvent être utilisés pour définir la base locale au fond de fissure [bib68] [bib67].

Les gradients sont déterminés grâce aux dérivées des fonctions de formes et des valeurs nodales des level sets.

$$\begin{cases} \nabla lsn_j^{elt} = \sum_i \phi_{i,j} lsn_i \\ \nabla lst_j^{elt} = \sum_i \phi_{i,j} lst_i \end{cases} \quad j=1,3$$

où les $\phi_{i,j}$ sont les dérivées des fonctions de forme par rapport à la direction j .

On détermine ainsi un champ de gradients par élément. Les valeurs sont calculées aux nœuds des éléments, pour chaque élément indépendamment des autres ; puis pour calculer la valeur nodale on moyenne sur les valeurs obtenues par éléments aux nœuds.

La base locale au fond de fissure $\{e_1, e_2, e_3\}$ se calcule alors en tout point grâce aux champs de gradients nodaux :

$$e_1 = \frac{\sum_i \phi_i \nabla lst_i}{\|\sum_i \phi_i \nabla lst_i\|}, \quad e_2 = \frac{\sum_i \phi_i \nabla lsn_i}{\|\sum_i \phi_i \nabla lsn_i\|}, \quad e_3 = e_1 \times e_2$$

où ∇lsn_i et ∇lst_i sont les valeurs nodales des gradients.

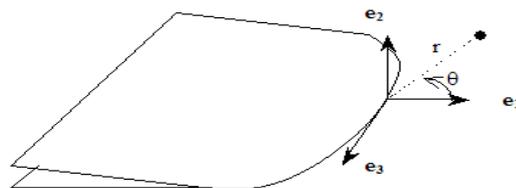


Figure 2.3-1 : Base locale au fond de fissure

En 2D on aura uniquement :

$$e_1 = \frac{\sum_i \phi_i \nabla lst_i}{\|\sum_i \phi_i \nabla lst_i\|}, \quad e_2 = \frac{\sum_i \phi_i \nabla lsn_i}{\|\sum_i \phi_i \nabla lsn_i\|}$$

2.4 Détermination du fond de fissure

Le fond de fissure est défini par l'intersection des iso-zéros des deux level sets. Pour le calcul des facteurs d'intensité de contraintes (Stress Intensity Factors en anglais), il est pratique de définir des points appartenant au fond de fissure, qui serviront de base pour l'interpolation des SIFs (voir le paragraphe [§5.1.3]).

Les points choisis sont les intersections des faces des éléments avec la courbe $lsn=0 \cap lst=0$. Ces points seront ensuite ordonnés de manière à définir une abscisse curviligne le long du fond de fissure.

2.4.1 Recherche des points du fond de fissure

La recherche des points du fond de fissure se fait de la manière suivantes. On se restreint aux éléments où la level set normale change de signe et où tous les nœuds sont de statuts « Crack-Tip » (cette notion est définie au paragraphe [§3.2.5.1]).

Soit une face d'un tel élément, si cette face est un quadrangle, on se ramène à deux triangles.

Soit le triangle ABC

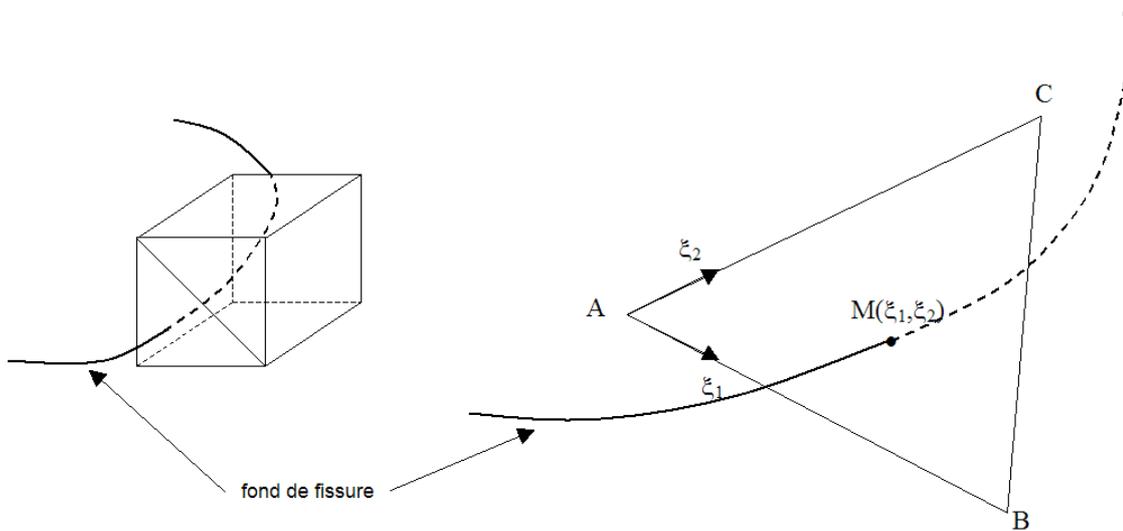


Figure 2.4.1-1 : Face d'un élément intersectée par le fond de fissure

On cherche le point M solution du système suivant :

$$\begin{cases} lsn(M)=0 \\ lst(M)=0 \end{cases}$$

On écrit le point M dans le repère (A, \vec{AB}, \vec{AC})

$$M = A + \xi_1 \vec{AB} + \xi_2 \vec{AC}$$

Le système s'écrit alors

$$\begin{cases} (1 - \xi_1 - \xi_2) lsn(A) + \xi_1 lsn(B) + \xi_2 lsn(C) = 0 \\ (1 - \xi_1 - \xi_2) lst(A) + \xi_1 lst(B) + \xi_2 lst(C) = 0 \end{cases}$$

$$\Leftrightarrow \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} lsn(B) - lsn(A) & lsn(C) - lsn(A) \\ lst(B) - lst(A) & lst(C) - lst(A) \end{bmatrix}^{-1} \begin{bmatrix} -lsn(A) \\ -lst(A) \end{bmatrix}$$

Le point M est retenu sous réserve qu'il appartienne au triangle ABC .

En 2D, on utilise le même procédé mais directement sur les mailles et non sur les faces des mailles (toujours des triangles ou bien des quadrangles découpés en triangles).

La recherche des points du fond de fissure s'accompagne d'une stratégie d'élimination des points M redondants. En effet, si la face n'est pas une face du bord géométrique de la structure, elle est alors

commune à deux mailles de la liste des mailles du fond, et un point du fond peut ainsi être détecté (au moins) deux fois. Pour vérifier si le nouveau point M détecté n'appartient pas déjà à la liste des points P du fond, on procède à une caractérisation du point M suivant qu'il se situe sur un nœud sommet de la face, une arête de la face ou bien à l'intérieur de la face. On effectue ensuite une vérification suivant les cas, via une boucle sur les points P de même statut que M :

- si M est situé sur un nœud sommet de numéro N^M et que $N^M = N^P$,
- si M est situé sur une arête caractérisée par les numéros de sommets N_1^M et N_2^M et que

$$\begin{cases} N_1^M + N_2^M = N_1^P + N_2^P \\ N_1^M \cdot N_2^M = N_1^P \cdot N_2^P \end{cases},$$

- si M est situé à l'intérieur d'une face caractérisée par NNF sommets et que

$$\begin{cases} \sum_{i=1}^{NNF} N_i^M = \sum_{i=1}^{NNF} N_i^P \\ \prod_{i=1}^{NNF} N_i^M = \prod_{i=1}^{NNF} N_i^P \\ \sum_{i,j \in (1,NNF), i \neq j} N_i^M \cdot N_j^M = \sum_{i,j \in (1,NNF), i \neq j} N_i^P \cdot N_j^P \end{cases},$$

alors le point M n'est pas conservé. Dans le cas contraire, il est ajouté à la liste des points P du fond. Pour s'affranchir des erreurs numériques qui pourraient engendrer qu'un point M normalement situé sur un nœud sommet se voit caractérisé comme un point intérieur à la surface, une procédure «fit to vertex» est associée à la stratégie précédente. Ainsi, si M est détecté comme un point situé sur une arête ou à l'intérieur d'une face, l'algorithme suivant est utilisé :

- calcul de la distance de chaque nœud sommet de la face au point M ,
- chacune de ces valeurs est placée dans le vecteur $Dist$ qui est ensuite trié de la plus petite valeur à la plus grande,
- si $Dist(1) < 10^{-4} \cdot Dist(2)$ alors M est replacé sur le nœud sommet. Sinon, il conserve son statut précédent.

Une procédure équivalente est mise en place pour replacer (si besoin est) un point intérieur à la face sur une arête.

Remarque :

Même lorsqu'on se restreint aux éléments où la level set normale change de signe et où tous les nœuds sont de type « Crack-Tip », on ne peut pas être sûr que le système soit inversible. On pourrait déjà se limiter aux faces triangulaires où les level sets changent de signe au sens large (0 inclus), mais cela n'éliminerait pas tous les cas de déterminant nul. En effet, lorsque la trace du fond de fissure sur la face triangulaire est une courbe, le système admet une infinité de solution. Ce cas n'est pas détectable a priori, et seul un test sur la non-nullité du déterminant permet de s'affranchir de tels cas. Ainsi, si sur une face le déterminant est nul numériquement, on ne détermine pas de point du fond de fissure sur cette face. S'il en existe (forcément une infinité), les deux points solution sur les bords de la face sont alors déterminés par une autre face de l'élément en question sur laquelle le déterminant est non-nul.

2.4.2 Orientation du fond de fissure

L'orientation du fond de fissure n'est nécessaire qu'en 3D.

2.4.2.1 Explication de la méthode d'orientation du fond de fissure

Lors de la recherche des points du fond de fissure par la méthode mentionnée au paragraphe [§17], les points ne sont pas forcément trouvés dans un ordre permettant la définition immédiate d'un chemin ordonné et d'une abscisse curviligne. Pourtant, la définition d'un fond de fissure ordonné et d'une abscisse curviligne le long du fond de fissure est indispensable au calcul de G par la méthode G-thêta.

La méthode d'orientation utilisée est basée sur le fait que deux points consécutifs du fond appartiennent forcément à une même maille 3D.

Pour expliquer cette méthode, on va se baser sur l'exemple d'un fond de fissure défini par une droite traversant une structure parallélépipédique. On a illustré sur la figure suivante la vue du dessus du maillage de cette structure en 3D :

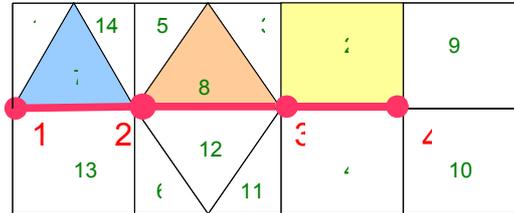


Figure 2.4.2.1-1 : Vue de dessus d'un maillage d'une structure parallélépipédique

La numérotation des mailles est arbitraire.

On peut voir sur la figure 2.4.2.1-1 le fond de fissure en rouge constitué de 4 points. L'ensemble des mailles de cette structure ont leur level set normale qui change de signe. Les 14 mailles représentées sont dites connectées au fond de fissure et possèdent chacune au moins un des quatre points du fond. La liste ordonnée des points de ce fond est 1-2-3-4.

Pour rechercher les points du fond, on fait une boucle sur les 14 mailles connectées au fond. Cette boucle étant faite dans l'ordre croissant des indices des mailles, on trouve d'abord le point 1 puis les points 3 et 4 (ou 4 puis 3) et enfin le point 2. On obtient donc la liste des points suivante : 1-4-3-2. Il reste à ordonner cette liste.

Étape 1 : Recherche des points du fond par maille

Le principe de l'orientation est basé sur le fait qu'une maille connectée au fond contient forcément 1 ou 2 points du fond. Si par exemple le fond passe par un seul sommet d'une maille, cette dernière ne contiendra qu'un point du fond. En rentrant dans une maille 3D par une face, le fond doit forcément ressortir par un autre point de cette même maille. Celle-ci contient alors deux points du fond.

Remarque :

Si une ou plusieurs mailles possèdent plus de deux points du fond, la procédure échoue et le fond n'est pas orienté. Un élément hexaédrique `HEXA8` peut contenir par exemple trois points du fond si celui-ci coupe nettement deux faces opposées et rase une autre face. Si l'orientation du fond est nécessaire pour la suite des calculs, il faut raffiner le maillage pour n'avoir que des mailles contenant au maximum deux points du fond.

La première étape de cette orientation consiste à associer pour chaque maille connectée au fond, un couple d'indices de points du fond leur appartenant.

Dans l'exemple, la maille 1 ne possède que le point 1, on lui associe le couple $(1,0)$. La maille 2 a les points 3 et 4, on lui associe le couple $(4,3)$, etc.

Soit $NMAFON$ le nombre de mailles connectées au fond.

On crée une liste $LISTPT$ de taille $2 \times NMAFON$, contenant les indices des points du fond pour chacune des $NMAFON$ mailles connectées au fond.

Dans notre exemple, la liste $LISTPT$ contient 14 couples d'indices et pourrait se décrire comme suit :

Indice de la maille	1	2	3	4	5	6	7	8	...
---------------------	---	---	---	---	---	---	---	---	-----

Couple associé à la maille	(1,0)	(4,3)	(3,0)	(3,4)	(2,0)	(2,0)	(2,1)	(3,2)	...
----------------------------	-------	-------	-------	-------	-------	-------	-------	-------	-----

Remarques :

- Les couples (4,3) et (3,4) sont équivalents.
- Les couples contenant un 0 ne seront pas considérés. En effet, un point appartiendra toujours à au moins un couple d'indices non nuls. Par exemple le point 1 appartient au couple (2,1).
- On détecte les points extrémités en regardant à combien de couples d'indices non nuls appartiennent les points. Le point 3 se trouve dans les deux couples différents d'indices non nuls (3,4) et (2,3). Ainsi, on sait automatiquement que le point 3 est situé entre le point 2 et le point 4. Le point 1 fait seulement partie du couple d'indices non nuls (2,1). Le point 1 est donc obligatoirement une extrémité du fond.

Étape 2 : Recherche des extrémités du fond

Comme on vient de le mentionner, les extrémités du ou des fonds de fissure sont repérables car un seul couple d'indices tous non nuls dans *LISTPT* contient leur indice. Grâce à cette indication, on crée la liste *PTEXTR* comprenant l'ensemble des points extrémités.

Dans l'exemple, le point 1 n'appartient qu'au couple (2,1) et le point 4 qu'au couple (4,3). Ce sont alors les deux extrémités de notre fond.

Étape 3 : Ordonnement des points du fond

Cette étape correspond à l'orientation du fond à proprement parler. Pour commencer l'ordonnement on part d'un point extrémité. Celui-ci est le premier point de la liste *PTEXTR*. Si le fond est fermé (fond sans extrémité), le premier point est celui ayant l'indice 1.

Dans notre exemple, le premier point est le point 1. Pour connaître le point suivant, il suffit de chercher le couple possédant un 1 et une deuxième valeur non nulle. Dans notre cas, il n'existe qu'un couple avec le 1 à savoir le (2,1), donc le deuxième point du fond est le 2. Ensuite, le point 2 ne se trouve que dans le couple (2,3), donc le point suivant est le 3. On continue cette démarche jusqu'à ce que tous les points soient ordonnés.

On a alors obtenu la liste des points 1-2-3-4.

Remarque :

- Si la première extrémité trouvée avait été le point 4, on aurait ordonné le fond dans l'autre sens. On aurait obtenu la liste ordonnée des points 4-3-2-1.

2.4.2.2 Cas d'un fond de fissure multiple

En 3D, le fond de fissure est une ligne soit fermée (fissure non débouchante), soit ouverte (fissure débouchante). Dans la plupart des cas, le fond de fissure est une ligne continue, comme celui de la fissure circulaire représentée sur la Figure 2.4.2.2-1.

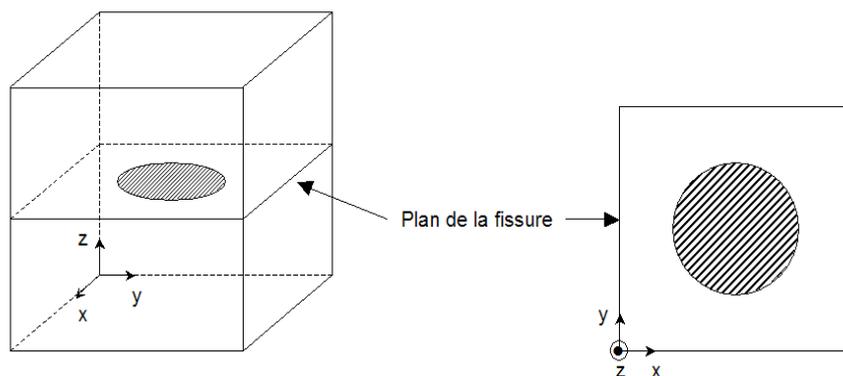


Figure 2.4.2.2-1 : Cas d'un fond de fissure continu

Cependant, il peut arriver que le fond de fissure soit en fait composé de plusieurs morceaux discontinus. C'est le cas par exemple de la fissure circulaire représentée sur la Figure 2.4.2.2-2. Dans ce cas, on parle toujours **du** fond de fissure, comme de l'ensemble des morceaux du fond. On dit que le fond de fissure est un fond multiple. Sur l'exemple de la Figure 2.4.2.2-2, le fond de fissure est composé des lignes courbes (BC) , (DE) , (FG) et (HA) .

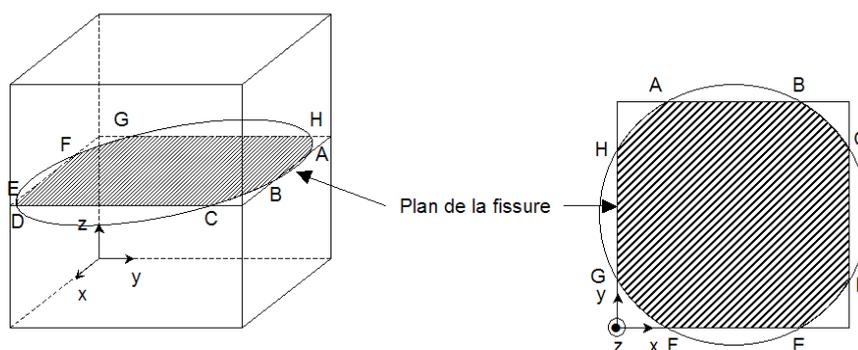


Figure 2.4.2.2-2 : Cas d'un fond de fissure multiple

La Figure 2.4.2.2-3 illustre la discrétisation d'un fond de fissure multiple (comprenant deux morceaux). Le premier morceau est composé des points 1 à 5 et le second morceau est composé des points 6 à 10.

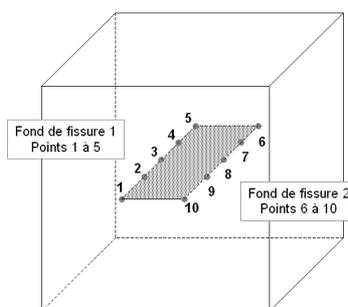


Figure 2.4.2.2-3 : Fond de fissure multiple

Dans le cas d'un fond multiple, les deux premières étapes d'ordonnancement des points sont inchangées par rapport au cas d'un fond continu.

Lors de la troisième étape, les fissures ayant un fond multiple sont détectées lorsque l'on atteint une deuxième extrémité, sans avoir ordonné l'ensemble des points du fond. On recherche alors le départ du fond suivant en prenant un point de la liste *PTEXTR*.

Dans le cas illustré sur la figure 2.4.2.2-3, *PTEXTR* contient les points 1, 5, 6 et 10. On commence par ordonner les points de 1 à 5. Puis, comme on sait qu'il reste cinq points à ordonner, on prend le point 6 de *PTEXTR* pour continuer l'orientation du fond.

Remarque :

| En 2D, chaque point du fond de fissure représente à lui seul un fond.

2.4.2.3 Algorithme utilisé pour l'orientation du fond

Soit :

- *NFON* le nombre de points du fond,
- *NMAFON* le nombre de mailles connectées au fond,
- *LISTPT* la liste des couples d'indices des points du fond associés aux mailles.

On initialise à 0 un vecteur d'entiers *TABPT* de longueur *NFON*, qui contiendra les indices des points ordonnés.

Le couple d'indices associé à une maille d'indice *IMA* est
(*LISTPT*($2 \times IMA$), *LISTPT*($2 \times IMA + 1$)).

le premier point de la liste des points ordonnés:
TABPT(1) = *PTEXTR*(1)

une fois le point extrémité utilisé, on le met à 0, pour éviter de le prendre en compte en cas de recherche d'une autre extrémité si on a un fond multiple:
PTEXTR(1) = 0

Boucle sur les points du fond : *IPT* = 1 à *NFON* - 1 :

◦ Boucle sur les mailles connectées au fond : *IMA* = 1 à *NMAFON* :

On cherche le couple d'indices non nuls où se situe le point du fond dernièrement rentré dans *TABPT*. Si c'est le cas, le deuxième indice du couple est forcément le point du fond suivant :

■ Si le deuxième indice du couple associé à *IMA* vaut 0, on ignore ce couple :

- passer à la maille suivante *IMA* + 1

■ Si le premier indice du couple correspond à celui recherché :

- Si *IPT* > 2 :

◦ Si le couple de *IMA* a déjà été trouvé (cas d'un couple en double) :

- mettre la deuxième valeur du couple à 0 pour ignorer ce couple à la prochaine itération
- aller à la maille suivante *IMA* + 1

• Si on a affaire à un nouveau couple, le deuxième indice est le point du fond suivant :

◦ *TABPT*(*IPT* + 1) = *LISTPT*($2 \times IMA + 1$)

◦ mettre la deuxième valeur du couple à 0 pour ignorer ce couple à la prochaine itération

■ Si c'est le deuxième indice du couple qui correspond à celui recherché :

- Si *IPT* > 2 :

◦ Si le couple de *IMA* a déjà été trouvé (cas d'un couple en double) :

- mettre la deuxième valeur du couple à 0 pour ignorer ce couple à la prochaine itération
- aller à la maille suivante *IMA* + 1

- Si on a affaire à un nouveau couple, le premier indice est le point du fond suivant :
 - $TABPT(IPT+1) = LISTPT(2 \times IMA)$
 - mettre la deuxième valeur du couple à 0 pour ignorer ce couple à la prochaine itération
- Si on n'a pas trouvé de points à associer à IPT , c'est donc un point extrémité (cas des fonds multiples)
 - on vérifie que IPT est bien un point de la liste $PTEXTR$ puis on le met à 0 pour l'ignorer à la prochaine itération
 - on recherche un nouveau point extrémité dans la liste $PTEXTR$ pour débiter le nouveau fond de fissure et on passe à la maille suivante $IMA+1$

3 Problème de fissuration avec X-FEM

3.1 Problème général

Dans cette partie, on rappelle les équations du problème général d'une structure fissurée. On considère une fissure Γ_c dans un domaine $\Omega \in \mathbb{R}^3$ délimité par $\partial\Omega$ de normale extérieure n_{ext} . Les lèvres de la fissure sont notées Γ^1 et Γ^2 de normales extérieures n^1 et n^2 . Les champs de contraintes et de déplacements sont respectivement notés σ et u . Un chargement quasi-statique est imposé sur la structure par l'intermédiaire d'une densité de forces volumiques f , d'une densité de forces surfaciques t sur Γ_t et d'une densité de forces surfaciques g sur les lèvres. Le solide est encastré sur Γ_u .

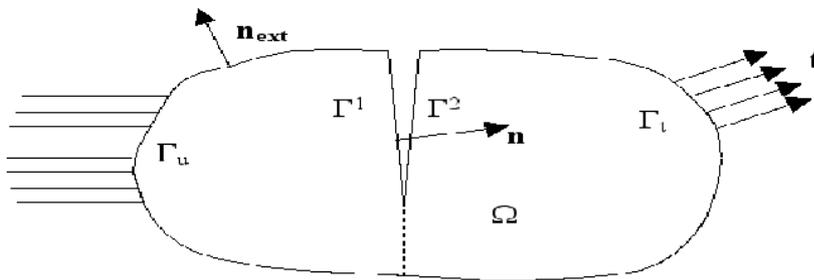


Figure 3.1-1 : Notations du problème général

La forme forte des équations d'équilibre et des conditions aux limites s'écrit :

$$\begin{aligned}
 \nabla \cdot \sigma &= f \quad \text{dans } \Omega \\
 \sigma \cdot n_{ext} &= t \quad \text{sur } \Gamma_t \\
 \sigma \cdot n^1 &= g \quad \text{sur } \Gamma^1 \\
 \sigma \cdot n^2 &= g \quad \text{sur } \Gamma^2 \\
 u &= 0 \quad \text{sur } \Gamma_u
 \end{aligned}
 \tag{3.1-1}$$

Nous nous plaçons dans le cadre de petites déformations et petits déplacements, pour lequel la relation déformations-déplacements s'écrit :

$$\varepsilon = \varepsilon(u) = \nabla_s u
 \tag{3.1-2}$$

où ∇_s est la partie symétrique du gradient et ε le tenseur des déformations.

On considère un matériau élastique linéaire¹. La loi de comportement du solide Ω s'écrit :

$$\sigma = C : \varepsilon \quad \text{dans } \Omega
 \tag{3.1-3}$$

où C est le tenseur de Hooke.

Le principe des travaux virtuels s'écrit :

1 Pour un matériau non-linéaire, rien ne change dans la formulation présentée ici. Seul le calcul des contraintes en fonction des déformations et des variables internes change, ce qui est transparent dans ce document.

$$\int_{\Omega} \sigma(u) : \varepsilon(v) d\Omega = \int_{\Omega} f \cdot v d\Omega + \int_{\Gamma_t} t \cdot v d\Gamma + \int_{\Gamma_c} g \cdot v d\Gamma \quad \forall v \in H_0^1(\Omega) \quad \text{éq 3.1-4}$$

où $H_0^1(\Omega)$ est l'espace de Sobolev des fonctions dont la dérivée est de carré intégrable, s'annulant sur $\partial\Omega$.

3.2 Enrichissement de l'approximation du déplacement

L'idée principale est d'enrichir la base des fonctions d'interpolation grâce à la partition de l'unité [bib68]. On rappelle l'approximation éléments finis classique :

$$u^h(x) = \sum_{i \in N_n(x)} a_i \phi_i(x)$$

où les a_i sont les degrés de liberté de déplacement au nœud i et ϕ_i les fonctions de forme associées au nœud i . $N_n(x)$ est l'ensemble des nœuds dont le support contient le point x . On assimile le support d'un nœud i au support des fonctions de forme associées à ce nœud, c'est-à-dire à l'ensemble de points x tels que $\phi_i(x) \neq 0$.

L'approximation enrichie s'écrit :

$$u^h(x) = \sum_{i \in N_n(x)} a_i \phi_i(x) + \sum_{j \in N_n(x) \cap K} b_j \phi_j(x) H(lsn(x)) + \sum_{k \in N_n(x) \cap L} \sum_{\alpha=1}^4 c_k^\alpha \phi_k(x) F^\alpha(lsn(x), lst(x))$$

Cette expression est composée de 3 termes. Le 1^{er} terme est le terme classique continu. Les 2^{ème} et 3^{ème} termes sont des termes enrichis. Étant au cœur de la méthode X-FEM, ces termes sont explicités dans les paragraphes suivants.

3.2.1 Enrichissent avec la fonction Heaviside (2^{ème} terme)

Afin de représenter le saut de déplacement à travers Γ_c , on introduit la fonction Heaviside généralisée $H(x)$ [bib68] définie par :

$$H(x) = \begin{cases} -1 & \text{si } x < 0 \\ +1 & \text{si } x \geq 0 \end{cases}$$

En se servant de la level set normale, la quantité $H(lsn(x))$ vaut -1 si le point x se trouve « au-dessous » la fissure et vaut $+1$ si le point x se trouve « au-dessus » de la fissure.

Les b_j sont les degrés de liberté enrichis. K est l'ensemble des nœuds dont le support est entièrement coupé par la fissure (nœuds représentés par un rond sur la Figure 3.2.1-1).

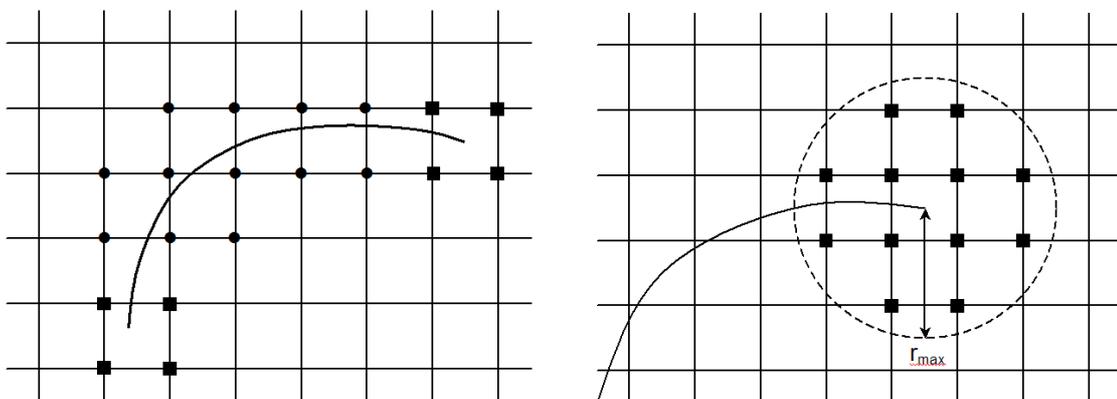


Figure 3.2.1-1 : À gauche, les nœuds « ronds » sont enrichis par la fonction Heaviside et les nœuds « carrés » par les fonctions singulières (enrichissement topologique). À droite, les nœuds « carrés » sont enrichis par les fonctions singulières (enrichissement géométrique).

3.2.2 Enrichissement de type jonction

Afin de représenter une fissure 2 qui se branche sur une fissure 1, on utilise un enrichissement de type jonction [bib70]. On considère un enrichissement de type Heaviside comme au paragraphe précédent pour définir la fissure 1. La fissure 2 se branchant sur la première, on considère un deuxième enrichissement pour celle-ci. Il s'agit d'une fonction Heaviside qui est « tronquée » au niveau du branchement. Les nœuds enrichis sont représentés sur la Figure 3.2.2-1.

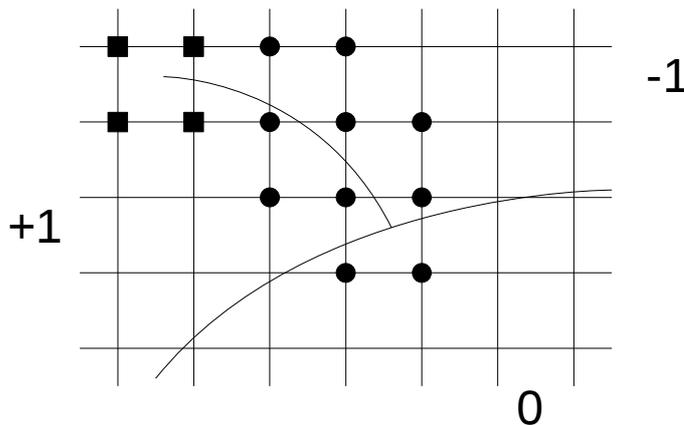


Figure 3.2.2-1 : Enrichissement pour la deuxième fissure. Les nœuds ronds sont enrichis par la fonction jonction, qui vaut +1, -1, 0.

La valeur de cette fonction dépend des levels set normales des 2 fissures. On considère le signe de la level set normale de la fissure 1, du côté où la fissure 2 est définie $sign(lsn_1(fiss_2))$ (dans la pratique, on regarde le signe d'un point appartenant au domaine de lsn_1 dans lequel se trouve la fissure 2, cf. opérande JONCTION de la doc [U4.82.08]), on a alors la fonction d'enrichissement jonction pour la fissure 2 qui s'écrit :

$$J_2(x) = \begin{cases} H(lsn_2(x)) & \text{si } sign(lsn_1(fiss_2)) H(lsn_1(x)) \geq 0 \\ 0 & \text{sinon} \end{cases}$$

Il est possible de brancher une troisième fissure sur la première (pour par exemple modéliser une intersection) :

$$J_3(x) = \begin{cases} H(lsn_3(x)) & \text{si } sign(lsn_1(fiss_3)) H(lsn_1(x)) \geq 0 \\ 0 & \text{sinon} \end{cases}$$

Si on souhaite brancher la troisième fissure sur la deuxième, il faut tenir compte du domaine de définition de la première, on aura donc :

$$J_3(x) = \begin{cases} H(lsn_3(x)) & \text{si } \forall i \in [1,2], sign(lsn_i(fiss_3)) H(lsn_i(x)) \geq 0 \\ 0 & \text{sinon} \end{cases}$$

Si on généralise l'approche à une fissure N qui se branche sur les fissures d'un ensemble K , on définit alors l'ensemble des fissures P , qui contient à la fois toutes les fissures de K et toutes les fissures sur lesquelles se branchent éventuellement les fissures de K . La fonction jonction s'écrit alors de manière générale :

$$J_N(x) = \begin{cases} H(\text{lsn}_N(x)) & \text{si } \forall i \in P, \text{sign}(\text{lsn}_i(\text{fiss}_n)) H(\text{lsn}_i(x)) \geq 0 \\ 0 & \text{sinon} \end{cases}$$

Un exemple de configuration est présenté Figure 3.2.2-2. Sur cet exemple on construit un arbre de connectivité des fissures. On déduit de cet arbre que pour $N=3$, on a $K=2$ et $P=[1,2]$. De même pour $N=5$, on déduit $K=[3,4]$ et $P=[1,2,3,4]$. Ainsi on a $J_5(x) = H(\text{lsn}_5(x))$ sur le domaine hachuré de la figure, et zéro ailleurs.

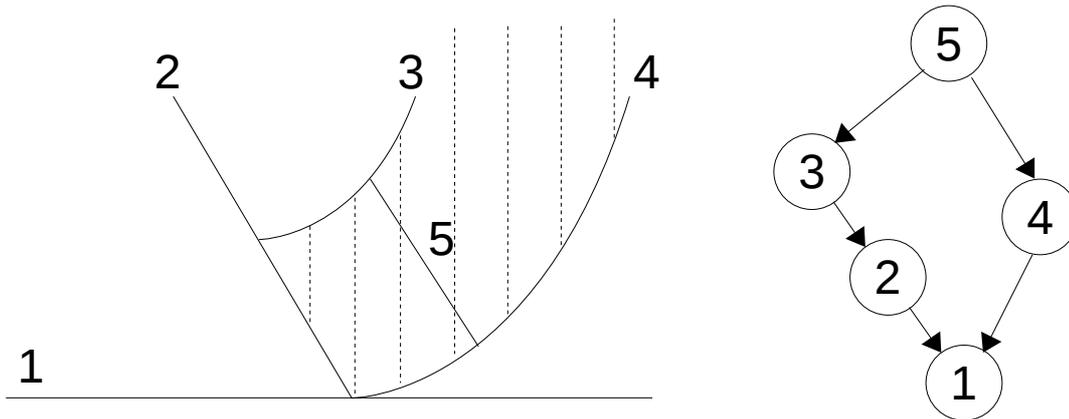


Figure 3.2.2-2 : Réseau de fissures à gauche, arbre de hiérarchie à droite, la zone hachurée correspond au domaine où la fonction d'enrichissement de la fissure 5 n'est pas nulle.

3.2.3 Enrichissement avec les fonctions singulières (3^{ème} terme)

Afin de représenter la singularité en fond de fissure, on enrichit l'approximation avec des fonctions basées sur les développements asymptotiques du champ de déplacement en mécanique de la rupture élastique linéaire [bib68]. Ces expressions ont été déterminées pour une fissure plane en milieu infini.

$$\begin{aligned} u_1 &= \frac{1}{2\mu} \sqrt{\frac{r}{2\pi}} \left(K_1 \cos \frac{\theta}{2} (\kappa - \cos \theta) + K_2 \sin \frac{\theta}{2} (\kappa + 2 + \cos \theta) \right) \\ u_2 &= \frac{1}{2\mu} \sqrt{\frac{r}{2\pi}} \left(K_1 \sin \frac{\theta}{2} (\kappa - \cos \theta) + K_2 \cos \frac{\theta}{2} (\kappa - 2 + \cos \theta) \right) \\ u_3 &= \frac{1}{2\mu} \sqrt{\frac{r}{2\pi}} K_3 \sin \frac{\theta}{2} \end{aligned} \quad \text{éq. 3.2.3-1}$$

$$\mu = \frac{E}{2(1+\nu)} \quad \text{et } \kappa = 3 - 4\nu \quad \text{en déformations planes.}$$

L'hypothèse de contraintes planes ne peut être retenue car aucune plaque fissurée n'est en situation de contraintes planes au voisinage de la singularité, lorsque l'on se place à une distance finie de la peau de la coque.

La base permettant de décrire ces champs comporte 4 fonctions :

$$\left\{ \sqrt{r} \cos \frac{\theta}{2}, \sqrt{r} \cos \frac{\theta}{2} \cos \theta, \sqrt{r} \sin \frac{\theta}{2}, \sqrt{r} \sin \frac{\theta}{2} \cos \theta \right\}.$$

Comme :

$$\begin{cases} \cos \frac{\theta}{2} \cos \theta = -\sin \theta \sin \frac{\theta}{2} + \cos \frac{\theta}{2} \\ \sin \frac{\theta}{2} \cos \theta = \sin \theta \cos \frac{\theta}{2} - \sin \frac{\theta}{2} \end{cases}$$

on choisit alors la base suivante² :

$$F = \left\{ \sqrt{r} \sin \frac{\theta}{2}, \sqrt{r} \cos \frac{\theta}{2}, \sqrt{r} \sin \frac{\theta}{2} \sin \theta, \sqrt{r} \cos \frac{\theta}{2} \sin \theta \right\} .$$

où (r, θ) sont les coordonnées polaires dans la base locale au fond de fissure (voir Figure 2.3-1 et Figure 3.2.3-1).

Ces coordonnées peuvent être exprimées aisément grâce aux level sets, puisque :

$$r = \sqrt{lsn^2 + lst^2}, \quad \theta = \arctan\left(\frac{lsn}{lst}\right), \quad \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$$

Dans la pratique, on utilise plutôt la fonction informatique $\text{atan2}(lsn, lst)$ qui renvoie la valeur principale de l'argument du nombre complexe (lst, lsn) exprimé en radians dans l'intervalle $[-\pi, \pi]$.

Pour des points se situant exactement sur la lèvres inférieure ($lsn=0$), la fonction $\text{atan2}(0, lst)$ donne un angle valant π . En théorie, $\text{atan2}(-0, lst)$ permet d'obtenir $-\pi$ comme attendu, mais ce n'est pas numériquement toujours le cas. Pour palier cet inconvénient, on utilise plutôt l'expression suivante pour l'angle θ :

$$\theta = H(lsn) |\text{atan2}(lsn, lst)|, \quad \theta \in [-\pi, \pi]$$

où $H(lsn)$ est la valeur de la fonction Heaviside. Ainsi, lorsqu'on se trouve sur la lèvres inférieure, la valeur $-\pi$ est bien atteinte.

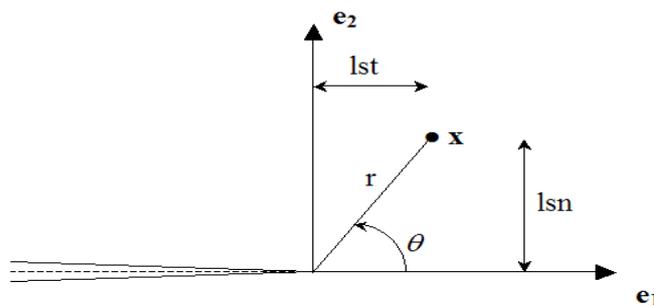


Figure 3.2.3-1 : Coordonnées polaires dans la base locale

On note que seule la 1^{ère} fonction de la base est discontinue au travers de la fissure. Les autres fonctions ne sont ajoutées que pour améliorer la précision. Ces fonctions sont les solutions de Westergaard, solutions asymptotiques analytiques d'un problème de rupture élastique en 2D. Cette base est bien adaptée aux cas 3D [bib67] [bib68], au moins pour les fissures dont le fond est assez régulier. Ces fonctions sont dites « singulières » car leurs dérivées sont singulières en $r=0$.

² Pour des lois de comportement non-linéaires, ce choix est conservé, bien que cette base ne permette pas de retrouver la solution exacte du problème.

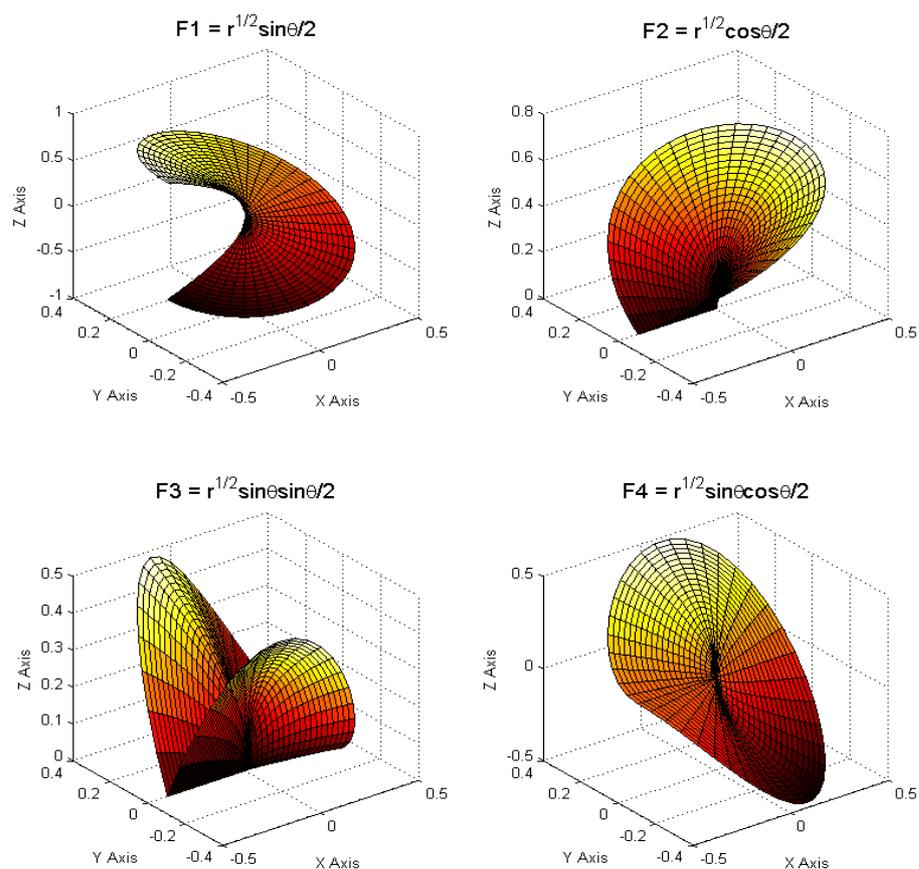


Figure 3.2.3-2 : Fonctions d'enrichissement en fond de fissure

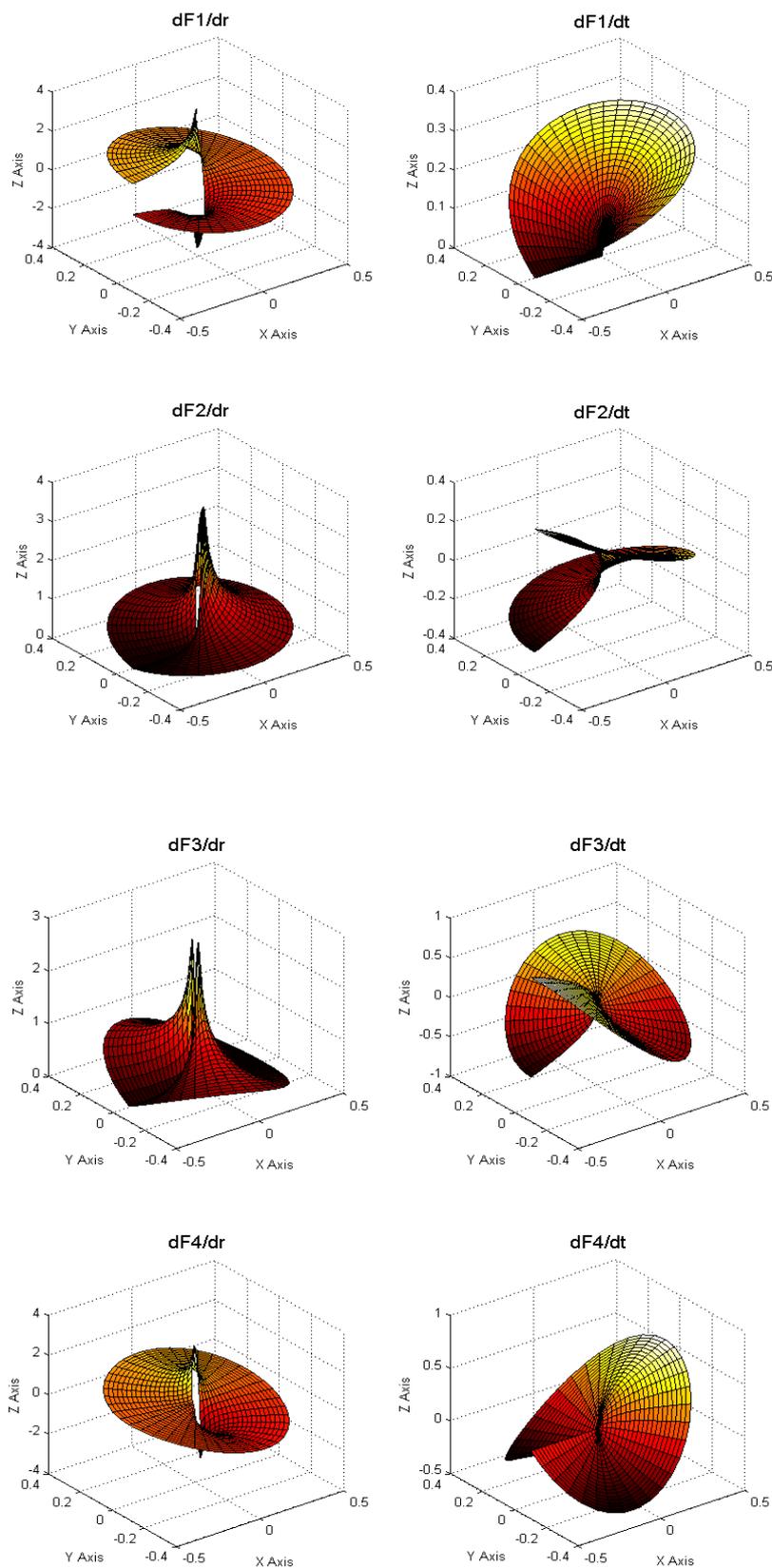


Figure 3.2.3-3 : Dérivées des fonctions d'enrichissement

Les c_k^α sont les degrés de liberté enrichis. L est l'ensemble des nœuds dont le support est partiellement coupé par le fond de fissure (nœuds représentés par un carré sur la Figure 3.2.1-1). Cela signifie qu'une seule couche d'éléments est enrichie autour du fond de fissure. Cet enrichissement est appelé « topologique ».

3.2.4 Enrichissement géométrique

Dès les premiers papiers sur X-FEM [bib68], il est notifié qu'un critère géométrique r_{\max} peut être défini pour déterminer les nœuds enrichis par les fonctions singulières (voir Figure 3.2.1-1) :

$$L = \{ \text{nœuds tels que } r < r_{\max} \}$$

Les premières études de convergence ont été effectuées en 2000 dans le cadre de la GFEM [bib68], avec prise en compte de plusieurs couches d'éléments enrichis en fond de fissure.

Quand on étudie la convergence, on s'intéresse à l'évolution de l'erreur par rapport au degré de raffinement du maillage. Généralement, on désigne par h une longueur caractéristique des éléments du maillage, et on cherche à déterminer le paramètre α appelé taux (ou vitesse ou ordre) de convergence, tel que l'erreur relative e_{rel} s'écrive de la forme :

$$e_{rel} = \|u - u_h\|_{H^1} \approx Ch^\alpha$$

où C est une constante indépendante de h .
Puisque

$$\log e_{rel} \approx \log C + \alpha \log h$$

le paramètre α apparaît comme la pente de la droite $\log e_{rel}$ en fonction de $\log h$ lorsque h tend vers 0.

Stazi *et al.* [bib68] étudie la convergence de l'erreur en énergie pour une plaque infinie avec une fissure droite, en mode I, pour des formulations linéaires et quadratiques. Il remarque que le quadratique améliore l'erreur, mais pas le taux de convergence. Béchet *et al.* [bib68] confirme cette observation et montre qu'une zone d'enrichissement fixe permet de retrouver un taux de convergence presque optimal.

Parallèlement, Laborde *et al.* [bib68] approfondit la question, et teste les taux de convergence pour les formulations polynomiales d'ordre supérieur. De plus, il apporte des améliorations afin de retrouver un taux optimal, voire une supra convergence. Le Tableau 3.2.4-1 rassemble les résultats obtenus par Laborde pour différentes variantes d'X-FEM, et ceci pour des approximations polynomiales de degré $k=1,2,3$.

	FEM	X-FEM	X-FEM (f. a.)	X-FEM (d. g.)	X-FEM (p. m.)
P1	0.5	0.5	0.9	0.5	1.1
P2	0.5	0.5	1.8	1.5	2.2
P3	0.5	0.5	2.6	2.6	3.3

Tableau 3.2.4-1 : Ordre de convergence des différentes variantes d'X-FEM

La première colonne correspond aux ordres de convergence de la méthode des éléments finis classique pour un problème de fissuration. Compte tenu de la singularité, la vitesse de convergence est en \sqrt{h} quel que soit le degré k . Les simulations 2D réalisées sur un problème test : fissure rectiligne sur un carré en mode I d'ouverture montre que X-FEM n'améliore pas le taux de convergence. Ceci peut s'expliquer par le fait que l'enrichissement topologique ne concerne qu'une seule couche d'éléments en fond de fissure. La zone d'influence de cet enrichissement est donc fortement lié à h . Ainsi, lorsque h tend vers 0, la taille de la zone d'influence de l'enrichissement tend aussi vers 0. L'idée qui semble naturelle est alors de ne plus limiter la zone d'enrichissement à une seule couche d'éléments, mais de l'élargir à une zone de taille fixe, indépendante du raffinement du maillage. La 3^{ème} colonne du Tableau 3.2.4-1 présente les résultats obtenus avec cette méthode dite X-FEM f. a. (pour Fixed enrichment Area). On retrouve presque les taux de convergence espérés ($\alpha=k$ pour une approximation P_k). Cependant, le conditionnement est dégradé par rapport à la méthode X-FEM habituelle. Afin de retrouver un conditionnement acceptable, Laborde propose de rassembler les degrés de libertés enrichis par les fonctions singulières. En clair, au lieu d'avoir des

ddls enrichis différents pour chaque nœud enrichi, ils sont globalisés afin d'en avoir un seul par fonction singulière et par bout de fissure (mais en 3D, on voit pas bien comment ça marche). Avec cette arrangement, le conditionnement est grandement amélioré, mais les taux de convergence sont plus faibles (X-FEM d. g.). Le problème provient des éléments de transition entre la zone enrichie et la zone non-enrichie. D'après Laborde, le phénomène s'explique par l'effet de la partition de l'unité qui ne peut pas être utilisée sur ces éléments partiellement enrichis. Pour pallier ce défaut, une ultime version est proposée : X-FEM p.m. (pour Pointwise Matching). Les déplacement des nœuds sur la frontière entre la zone enrichie et la zone non-enrichie sont imposés égaux. Grâce à ce recollement, on obtient des taux de convergence espérés (voire une légère super-convergence).

Remarque :

Pour les approximation polynomiales de degré $k=2,3$, Laborde et al. [bib68] comme beaucoup d'autres utilise des fonctions de forme de degré k pour les termes classiques et enrichis par Heaviside, de manière à ce que le saut soit de degré k . Par contre, pour les termes enrichis par les fonctions singulières, il suffit d'utiliser les fonctions de forme linéaires pour capter la singularité en fond de fissure et de pas trop détériorer le conditionnement des matrices.

Actuellement, dans Code_Aster, seule une approximation de degré 1 est possible, et le fait de renseigner ou non une valeur de rayon pour la zone d'enrichissement (mot-clé RAYON_ENRI) permet de se placer respectivement dans le cadre X-FEM (f. a.) ou X-FEM classique.

3.2.5 Enrichissement dans Code_Aster

3.2.5.1 Enrichissement (statut) des nœuds

Pour savoir si un nœud est enrichi par la fonction Heaviside (nœud de type « Heaviside) ou par les fonctions singulières (nœud de type « crack-tip »), on calcule le min et le max de la level set normale sur tous les nœuds appartenant au support du nœud (la notion de « support » est définie au paragraphe [§3.2]), et on calcule le min et max de la level set tangente sur tous les points appartenant au support du nœud considéré où la level set normale s'annule.

$$j \in K \Leftrightarrow \left(\min_{x \in N_n(j)} (l_{sn}(x)) \max_{x \in N_n(j)} (l_{sn}(x)) < 0 \right) \text{ et } \left(\max_{x \in N_n(j) \cap l_{sn}(x)=0} (l_{st}(x)) < 0 \right)$$
$$k \in L \Leftrightarrow \left(\min_{x \in N_n(j)} (l_{sn}(x)) \max_{x \in N_n(j)} (l_{sn}(x)) \leq 0 \right) \text{ et } \left(\min_{x \in N_n(j) \cap l_{sn}(x)=0} (l_{st}(x)) \max_{x \in N_n(j) \cap l_{sn}(x)=0} (l_{st}(x)) \leq 0 \right)$$

Des idées similaires apparaissent dans [bib68], mais il semblerait que certains cas de figures n'aient pas été pris en compte. Ces expressions sont l'aboutissement des premiers efforts [bib68] qui visaient à déterminer les types d'enrichissement uniquement à l'aide des level sets. Le lecteur y trouvera les figures correspondantes explicatives.

Dans le cas d'un enrichissement géométrique en fond de fissure, le critère de sélection des nœuds est le suivant :

$$k \in L \Leftrightarrow \sqrt{l_{sn}(x)^2 + l_{st}(x)^2} \leq r_{\max}$$

Concernant le choix de la valeur du rayon d'enrichissement, rien n'est clairement indiqué dans la littérature. Il semblerait cependant qu'un rayon valant entre $1/5$ et $1/10$ de la longueur de la fissure soit un choix pertinent.

Des récentes études ont montré que l'enrichissement géométrique dégradant fortement le conditionnement de la matrice de rigidité, il fallait le limiter dans une zone restreinte autour du fond de fissure, dans l'attente de traitement permettant d'améliorer le conditionnement. On propose une alternative, qui est à mi-chemin entre l'enrichissement topologique et géométrique : un enrichissement sur n couches [bib68]. Dans ce cas, on calcule un r_{\max} (unique pour chaque fissure) d'après la donnée utilisateur du nombre de couches, puis on applique la formule précédente.

Algorithme de choix de l'enrichissement des nœuds :

Soit $MAFIS$ l'ensemble des mailles sur lesquelles la level set normale s'annule

- boucle sur tous les nœuds P du maillage

initialisation des max et min des level sets

- boucle sur les mailles de $MAFIS$ contenant le nœud P

- boucle sur les arêtes de la maille

soient A et B les deux extrémités du segment

si $lsn(A)=0$ alors

actualisation si nécessaire de $maxlst$ et $minlst$ avec $lst(A)$

fin si

si $lsn(B)=0$ alors

actualisation si nécessaire de $maxlst$ et $minlst$ avec $lst(B)$

fin si

si $lsn(A)lsn(B)<0$ alors

$$C = A - \frac{lsn(A)}{lsn(B) - lsn(A)}(B - A) \quad \text{éq. 3.2.5.1-1}$$

actualisation si nécessaire de $maxlst$ et $minlst$ avec $lst(C)$

fin si

- fin boucle

- boucle sur les nœuds sommets de la maille

actualisation si nécessaire de $maxlsn$ et $minlsn$

- fin boucle

- fin boucle

si $(minlsn \cdot maxlsn < 0)$ et $(maxlst \leq 0)$ alors $P \in K$

cas d'enrichissement topologique

si $(minlsn \cdot maxlsn \leq 0)$ et $(minlst \cdot maxlst \leq 0)$ alors $P \in L$

cas d'enrichissement géométrique :

si $\sqrt{lsn(P)^2 + lst(P)^2} \leq r_{\max}$ alors $P \in L$

- fin boucle

Pour obtenir l'équation [éq. 3.2-2], ainsi que la valeur de la level set tangente au point C , on détermine au préalable l'abscisse curviligne s telle que

$$C = A + s(B - A)$$

grâce au fait que la level set normale s'annule en C , soit

$$lsn(C) = lsn(A) + s(lsn(B) - lsn(A)) = 0$$

On en déduit l'expression du point C ainsi que la valeur de la level set tangente en C :

$$\begin{aligned}lst(C) &= lst(A) + s(lst(B) - lst(A)) \\ &= lst(A) - \frac{lsn(A)}{lsn(B) - lsn(A)} (lst(B) - lst(A))\end{aligned}$$

Remarque :

| Un même nœud peut appartenir aux ensembles K et L .

3.2.5.2 Enrichissement (statut) des mailles

Dans *Code_Aster*, il faut définir des types d'éléments finis précis, et pour ne pas multiplier le nombre des possibilités, le choix a été fait de définir 3 types d'éléments finis X-FEM : les éléments « Heaviside », les éléments « crack-tip » et les éléments mixte « Heaviside et crack-tip ».

Si la maille possède au moins un nœud de type « Heaviside », alors c'est une maille « Heaviside ».
Si la maille possède au moins un nœud de type « crack-tip », alors c'est une maille « crack-tip ».
Si la maille possède au moins un nœud « Heaviside » et au moins un nœud « crack-tip », ou si la maille contient au moins un nœud « Heaviside et crack-tip », alors c'est une maille « Heaviside et crack-tip ».

Notons $GRMAEN1$ les mailles « Heaviside », $GRMAEN2$ les mailles « crack-tip » et $GRMAEN3$ les mailles « Heaviside et crack-tip ».

Soit la maille i , et Ni l'ensemble des nœuds j de la maille i .

si $(\exists j \in Ni, \text{ tel que } j \in K)$ alors $i \in GRMAEN1$

si $(\exists j \in Ni, \text{ tel que } j \in L)$ alors $i \in GRMAEN2$

si $(\exists (j, k) \in Ni^2, \text{ tels que } j \in K \text{ et } k \in L)$ ou $(\exists j \in Ni, \text{ tel que } j \in K \cap L)$ alors $i \in GRMAEN3$

On remarque que tous les nœuds d'un élément seront affectés des mêmes caractéristiques et du même enrichissement, or ce n'est pas forcément ce qui est voulu. Il faut donc annuler les degrés de liberté enrichis « en trop ».

3.2.5.3 Annulation des degrés de liberté enrichis « en trop »

On a vu au paragraphe précédent qu'une maille de type « Heaviside » ne peut comporter par exemple qu'un seul nœud de type « Heaviside », les autres nœuds de la maille étant des nœuds classiques ne nécessitant aucun enrichissement. Or ces nœuds vont être affectés des degrés de liberté de la maille « Heaviside », donc de degrés de liberté enrichis. Par conséquent, il faut procéder à une annulation de ces degrés de liberté enrichis à tort. L'annulation permet dans les faits de passer continûment d'une zone enrichie à une zone non enrichie et permet de faire cohabiter deux types éléments (au sens d'ils partagent une frontière commune), l'un enrichi, l'autre non enrichi. La variable non enrichie est la même sur la frontière commune et les degrés de liberté correspondant à l'enrichissement sont mis à zéro pour l'élément qui est enrichi sur cette même frontière (mais pas ailleurs dans ce même élément). Cette façon de faire évite d'avoir à résoudre la question des « blending elements » dont on peut trouver un traitement dans [bib71].

Plusieurs cas se présentent :

- degrés de liberté Heaviside à annuler aux nœuds classiques d'une maille Heaviside ou mixte
- degrés de liberté crack-tip à annuler aux nœuds classiques d'une maille crack-tip ou mixte
- degrés de liberté Heaviside à annuler aux nœuds crack-tip d'une maille mixte
- degrés de liberté crack-tip à annuler aux nœuds Heaviside d'une maille mixte.

La technique d'annulation de ces degrés de liberté est expliquée en détail dans [R5.03.54, §4.4].

3.2.6 Conditionnement lié à l'enrichissement

Le conditionnement, noté 10^δ correspond au rapport entre la plus grande et la plus petite valeur propre d'un système à inverser. Pour un calcul en double précision avec une erreur numérique en

10^{-15} , l'erreur relative obtenue sur le calcul est de l'ordre de $10^{-15+\delta}$. On doit donc vérifier la condition $\delta < 9$ pour garantir une précision numérique de l'ordre de 10^{-6} .

L'enrichissement géométrique dégrade fortement le conditionnement de la matrice de rigidité [bib68], [bib68]. Béchet et al. [bib68] proposent une technique d'orthogonalisation des degrés de liberté lors du calcul des matrices de rigidité élémentaires afin d'améliorer le conditionnement de la matrice assemblée. Laborde et al. [bib68] expliquent que le mauvais conditionnement est dû au fait que la base d'enrichissement choisie ne forme pas une famille libre localement. Ils proposent donc de mettre un seul degré de liberté pour ces fonctions sur toute la zone d'enrichissement et de raccorder les déplacements à la limite entre zones enrichies et non enrichies afin de retrouver des taux de convergence optimaux. Le problème de conditionnement est d'ailleurs tel qu'avec des éléments quadratiques il devient impossible d'obtenir des résultats, sans mettre en place une des techniques [bib68], [bib68]. En effet, pour ces éléments, le mauvais conditionnement est dû non seulement à la partie singulière de l'enrichissement, mais aussi à l'enrichissement Heaviside, lorsque la fissure passe très proche d'un nœud. On présente l'évolution du nombre de conditionnement à mesure que l'interface se rapproche des nœuds du maillage sur la figure 3.2.6-1, pour des éléments linéaires et quadratiques respectivement. Les valeurs de cette figure sont très approximatives. D'une part on ne prend pas en compte les éléments du voisinage. D'autre part ces valeurs sont obtenues de façon grossière. Dans le cas du cube coupé dans un coin par exemple (à droite sur la figure), on considère la rigidité du petit cube du coin de côté $10^{-\gamma}$ au lieu du petit tétraèdre du coin. Enfin le conditionnement d'un problème global est généralement plus grand que le conditionnement local et augmente lorsque l'on raffine le maillage. En prenant en compte toutes ces considérations, on obtient dans la pratique des conditionnements d'un ordre 2 à 4 fois supérieurs à ceux de la figure 3.2.6-1.

$\delta = \gamma$	$\delta = 2\gamma$	$\delta = \gamma$	$\delta = 4\gamma$	$\delta = 3\gamma$	$\delta = \gamma$	$\delta = 4\gamma$	$\delta = 7\gamma$
$\delta = 3\gamma$	$\delta = 4\gamma$	$\delta = 3\gamma$	$\delta = 6\gamma$	$\delta = 5\gamma$	$\delta = 3\gamma$	$\delta = 6\gamma$	$\delta = 9\gamma$

Figure 3.2.6-1 : la distance au nœud sommet le plus proche normalisée par la longueur du côté valant $10^{-\gamma}$ on montre la dépendance de δ , nombre de conditionnement, par rapport au paramètre γ pour les éléments linéaires en partie supérieure et quadratiques en partie inférieure.

La technique de réajustement de la level set évoquée au paragraphe [§2.2.4] permet de se prémunir de ce mauvais conditionnement. En notant $10^{-\gamma}$ la distance du point d'intersection de la level set au nœud sommet le plus proche normalisée par la longueur du côté, le réajustement à 1% de la longueur d'arête faite au paragraphe [§2.2.4] correspond à $\gamma=2$. Ce réajustement doit agir suffisamment vite pour que le conditionnement ne soit pas trop détérioré, mais pas pour des valeurs de γ trop faibles de façon à ne pas perturber le système en déplaçant de façon non réaliste la surface de fissuration. Pour des éléments hexaèdres quadratiques, s'il faut que $10^{-15+9\gamma+2}$ soit de l'ordre de 10^{-5} , on obtient $9\gamma+2=10$ soit un réajustement à 13% de la longueur d'arête. Il n'est donc pas possible dans ce cas là d'activer raisonnablement le réajustement des level sets au sommet, afin que le conditionnement ne soit pas détérioré.

Dans ces conditions, on met en place un critère permettant de détecter si un degré de liberté Heaviside pose problème. Si le critère est vérifié, le degré de liberté est éliminé par mise à zéro, comme au paragraphe [§ 3.2.5.3].

Le principe est le même que le critère volumique proposé par Daux [bib68] pour les jonctions. L'idée est de regarder pour chaque nœud dont le support est coupé par la level set, le rapport des tailles des zones de part et d'autres de la level set sur ce support (zones affectées d'une valeur de Heaviside valant ± 1). Si :

$$\frac{\min(V_{-1}, V_{+1})}{V_{support}} \leq 10^{-\alpha} \quad \text{éq. 3.2.6-1}$$

les degrés de liberté Heaviside du nœud concerné sont mis à zéro dans toutes les directions.

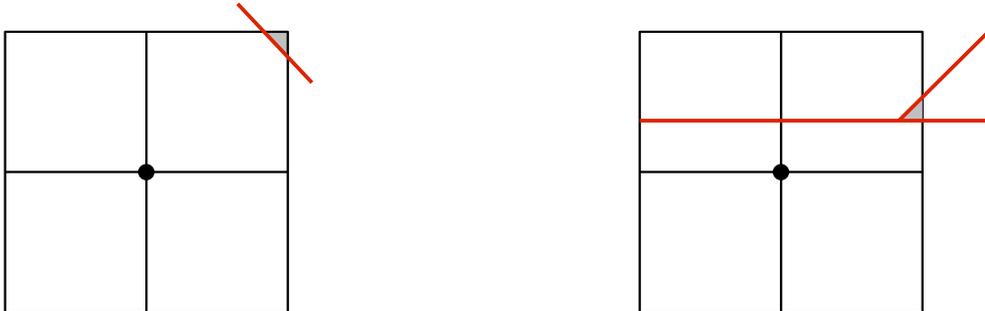


Figure 3.1-3.2.6-2 : illustration de la mise en place du critère volumique. Le support du nœud coupé par la level set est illustré. On compare les volumes gris et blanc par rapport au volume total du support du nœud (réunion des volumes gris et blanc) dans le cas d'une fissure (gauche) et d'une jonction (droite).

Si ce critère est pertinent pour les éléments linéaires (triangles, tétraèdres) avec une valeur de α de 4, il n'est pas satisfaisant pour les éléments multi-linéaires (quadrangles, pyramides, pentaèdres, hexaèdres) et quadratiques. En effet la valeur de α de 4 conduit à éliminer des degrés de liberté qui ne devraient pas l'être, ce qui perturbe la solution, alors que des valeurs plus élevées de α dégradent le conditionnement. Afin de prendre en compte ces éléments, on utilise un critère de rigidité, qui se base sur une comparaison des rigidités du support et non plus simplement des volumes. Pour chaque nœud dont le support est coupé par la level set, on regarde pour ce support le rapport des rigidités des zones de part et d'autres de la level set (zones affectées d'une valeur de Heaviside valant ± 1). Si en un nœud n où se sont les sous-éléments de son support, on a :

$$\frac{\min\left(\sum_{se-1} \int_{\Omega_{se}} \|\phi_{n,X}\|^2 d\Omega_{se}, \sum_{se+1} \int_{\Omega_{se}} \|\phi_{n,X}\|^2 d\Omega_{se}\right)}{\sum_{se} \int_{\Omega_{se}} \|\phi_{n,X}\|^2 d\Omega_{se}} \leq 10^{-\delta} \quad \text{éq 3.2.6-2}$$

où $\phi_{n,X}$ est la dérivée de la fonction de forme au nœud n dans la direction globale X , alors les degrés de liberté Heaviside du nœud n sont mis à zéro dans toutes les directions. On remarquera que le comportement n'est pas présent dans le critère de rigidité de éq3.2.6, le critère ayant été normalisé. Cela permet de discriminer à l'avance les degrés de liberté à éliminer, sans connaissance du problème à résoudre (non-linéarités, plasticité, contact, etc). Ce critère, très proche d'un critère de conditionnement, nous amène à choisir des valeurs de δ comprises entre 8 et 10. Dans la pratique, nous prenons une valeur de δ de 9.

Le critère décrit dans éq3.2.6-2, quantifie en ordre de grandeur, le compromis entre qualité de la solution et conditionnement. Comme ce critère n'est valable qu'en ordre de grandeur, il n'est pas forcément pertinent, de calculer exactement toutes les intégrales. Dans la programmation Aster, on approxime donc le calcul intégral sur les sous-éléments ([bib71]):

$$\int_{\Omega_{se}} \|\phi_{n,X}\|^2 d\Omega_{se} \approx \|\phi_{n,X}(G^{se})\|_{\infty}^2 V^{se} \quad \text{éq 3.2.6-3}$$

où G^{se} désigne le barycentre du sous-élément, V^{se} le volume du sous-élément.

Au de lieu de calculer l'intégrale sur le sous-élément, on évalue les dérivées de fonctions de forme en un point. Cette stratégie permet de capter en ordre de grandeur le critère d'élimination de éq3.2.6, avec un faible coût de calcul.

Cette approximation s'avère robuste et peu coûteuse avec des éléments linéaires.

En revanche, avec des éléments quadratiques, les dérivées des fonctions de forme admettent des racines. Le calcul en un point est risqué: le barycentre peut-être une racine des dérivées de fonctions de forme, où l'estimation sera nulle, mais pas l'intégrale sur le sous-élément. Cette mauvaise estimation peut engendrer aléatoirement des éliminations, ce qui perturbe la solution et n'est pas acceptable.

On renforce donc l'estimation pour les éléments quadratiques, en rajoutant d'autres points d'évaluation, en plus du barycentre:

$$\int_{\Omega_{se}} \|\phi_{n,x}\|^2 d\Omega_{se} \approx \max_{P_i \in \Omega_{se}} (\|\phi_{n,x}(P_i)\|_{\infty}^2) V^{se} \quad \text{éq 3.2.6-4}$$

Pour que l'évaluation soit pertinente, les points sont suffisamment bien distribués: on considère les points équidistants entre les nœuds sommets et le barycentre (par exemple, 3 points pour un tri6).

$$\{P_i\} = \{G^{se}\} \cup \left\{ \frac{\text{Node}_i + G^{se}}{2} \right\} \quad \text{éq 3.2.6-5}$$

Une analyse de ces différents critères est faite dans [bib 71] . Elle montre que les éliminations conduisent à une absence de convergence de l'erreur en énergie sur un exemple simple de compression homogène d'un cube traversé par une interface inclinée. La solution proposée consiste à remplacer l'élimination des degrés de liberté Heaviside par une orthogonalisation des matrices de rigidité locales, idée qui vient du pré-conditionneur X-FEM de [bib 68] . L'orthogonalisation n'aura lieu que si le critère éq 3.2.6 est satisfait. On propose ainsi avec le critère éq 3.2.6 de procéder de la manière suivante :

$$\begin{cases} \delta < 5 & \text{on ne fait rien} \\ 5 < \delta < 14 & \text{on applique l'orthogonalisation} \\ \delta > 14 & \text{le degré de liberté est éliminé} \end{cases} \quad \text{éq 3.2.6-6}$$

En faisant de la sorte, le conditionnement est ramené à 10^6 , ce qui permet d'utiliser un solveur direct sans pré-conditionneur global. Cependant, si le bon conditionnement du système est assuré, les erreurs en énergie ou en déplacement augmentent lorsque δ est proche de 14. En effet, on est alors obligé de réactiver l'élimination pour $\delta > 14$ car l'orthogonalisation ne permet d'obtenir des résultats corrects que si $10^{-15+\delta} \ll 1$ où 10^{-15} est l'erreur numérique en double précision. Afin de lever cette dernière difficulté on propose une estimation des matrices de rigidité locales en triple précision.

3.3 Sous-découpage

Une attention particulière doit être portée lors de l'intégration numérique des termes de rigidité et de second membre d'un élément traversé par la fissure. En effet, sur un élément traversé par une fissure, les gradients des déplacements peuvent être discontinus, et dans ce cas l'intégration numérique de Gauss-Legendre sur la totalité de l'élément n'est pas applicable. Afin de se replacer dans des conditions de régularité classiques, il convient de procéder à une intégration sur des domaines où l'intégrande est au moins continue. Pour un élément traversé par une fissure, il faut donc intégrer séparément de part et d'autre de la fissure (ceci apparaît pour la 1^{ère} fois dans [bib68] pour le 2D et dans [bib68] pour le 3D). Plusieurs procédures sont envisageables, et facilement mises en œuvre en 2D. Les difficultés apparaissent avec le 3D.

On cherche à sous-découper en sous-tétraèdres un élément volumique quelconque (tétraèdre, pentaèdre, hexaèdre) coupé par une surface. On rappelle que cette découpe ne sert qu'à des fins d'intégration, elle est purement virtuelle et aucun nœud n'est ajouté au maillage. Le maillage n'est en rien modifié.

Pour les pentaèdres et les hexaèdres, le nombre de possibilités étant trop important et les configurations trop compliquées, on préfère se ramener au découpage de tétraèdres. On réalise donc une phase préalable qui consiste à découper de manière systématique les pentaèdres et hexaèdres en tétraèdres (indépendamment de la position de la fissure). Chaque tétraèdre ainsi obtenu sera alors découpé en sous-tétraèdres d'après l'algorithme ci-dessous.

3.3.1 Phase préalable de découpe des hexaèdres pour se ramener à des tétraèdres :

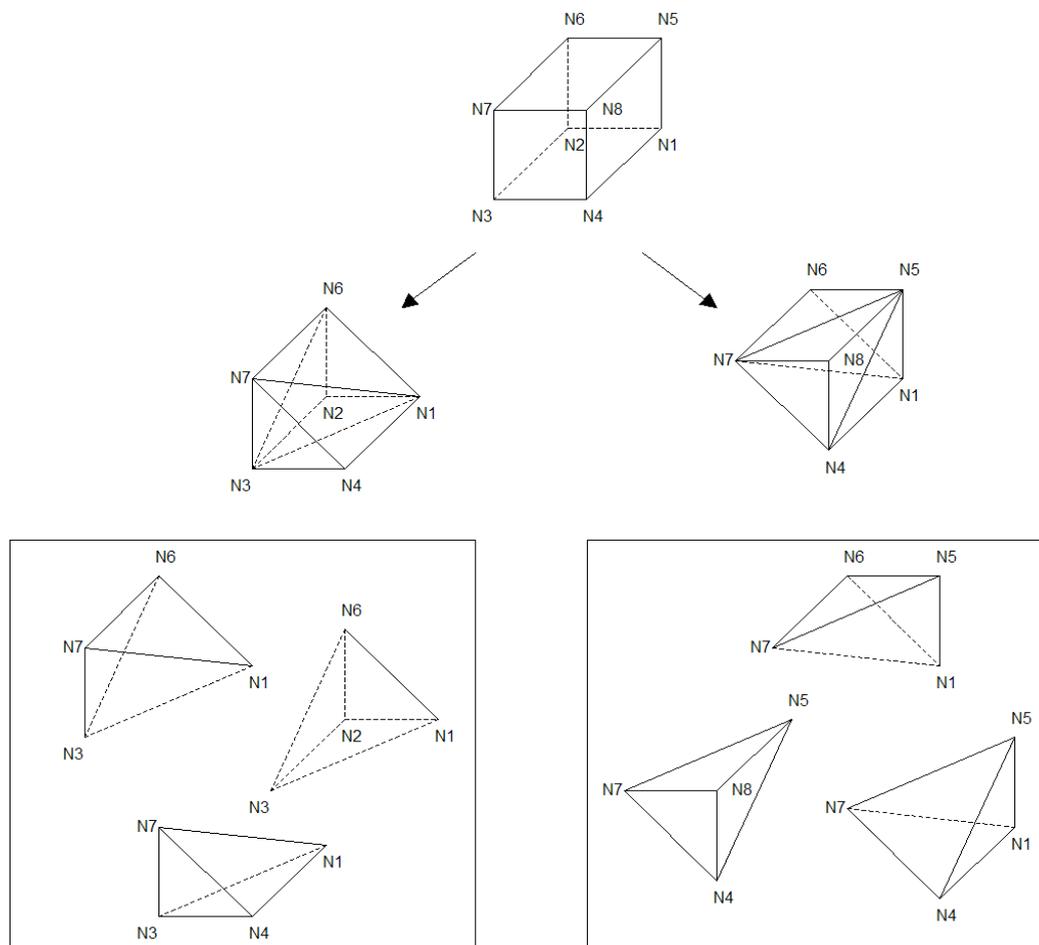


Figure 3.3.1-1 : Division d'un hexaèdre en tétraèdres

Un hexaèdre se divise alors en 6 tétraèdres, répertoriés dans le tableau suivant :

hexaèdre	tétraèdre
<i>N1 N2 N3 N4 N5 N6 N7 N8</i>	<i>N7 N4 N3 N1</i>
	<i>N1 N6 N2 N3</i>
	<i>N3 N6 N7 N1</i>
	<i>N6 N1 N5 N7</i>
	<i>N4 N7 N8 N5</i>
	<i>N4 N5 N1 N7</i>

Tableau 3.3.1-1 : Division d'un hexaèdre en tétraèdres

On note que ce découpage est arbitraire. On aurait pu choisir une autre manière de découper un hexaèdre en 2 pentaèdres ainsi qu'une autre façon de découper un pentaèdre en 3 tétraèdres.

3.3.2 Phase préalable de découpe des pentaèdres pour se ramener à des tétraèdres :

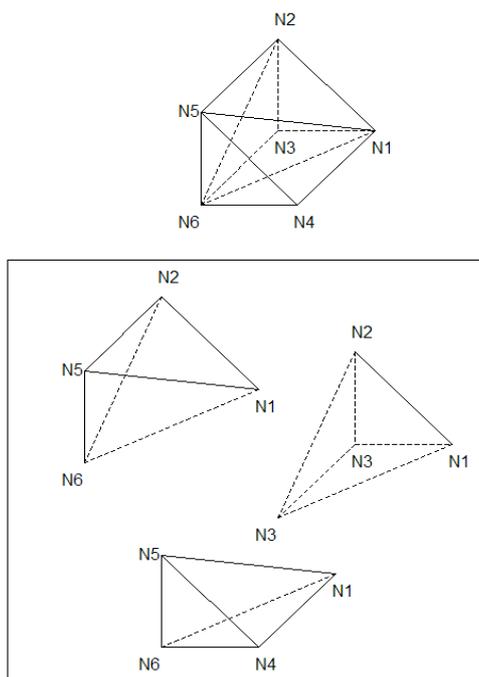


Figure 3.3.2-1 : Schéma de division d'un pentaèdre en tétraèdres

Un pentaèdre se divise alors en 3 tétraèdres, répertoriés dans le tableau suivant :

pentaèdre	tétraèdre
<i>N1 N2 N3 N4 N5 N6</i>	<i>N5 N4 N6 N1</i>
	<i>N1 N2 N3 N6</i>
	<i>N6 N2 N5 N1</i>

Tableau 3.3.2-1 : Division d'un pentaèdre en tétraèdres

3.3.3 Sous-découpage d'un tétraèdre en sous-tétraèdres :

Le tétraèdre de référence est défini sur la Figure 3.3.3-1. On détermine les points d'intersection P_i entre la surface $l_{sn}^h = 0$ et les arêtes du tétraèdre.

Soit n le nombre de points d'intersection P_i .

À chaque point d'intersection P_i , on associe deux entiers : A_i et NS_i

A_i est le numéro de l'arête sur laquelle se trouve P_i (par exemple si P_i se trouve sur l'arête d'extrémités $N2-N3$, alors $A_i=4$). Dans le cas où P_i coïncide avec un nœud sommet du tétraèdre, A_i vaut 0,

NS_i est le numéro du nœud sommet dans le cas où P_i coïncide avec un nœud sommet du tétraèdre (par exemple si P_i coïncide avec $N3$, alors $NS_i=3$). Dans le cas où P_i se trouve sur une arête, NS_i vaut 0.

Remarque :

| Le produit de A_i par NS_i vaut toujours 0.

Les points d'intersection sont ensuite triés suivant l'ordre croissant des A_i . Les points d'intersection coïncidant avec des nœuds sommet vont donc se retrouver au début de la liste.

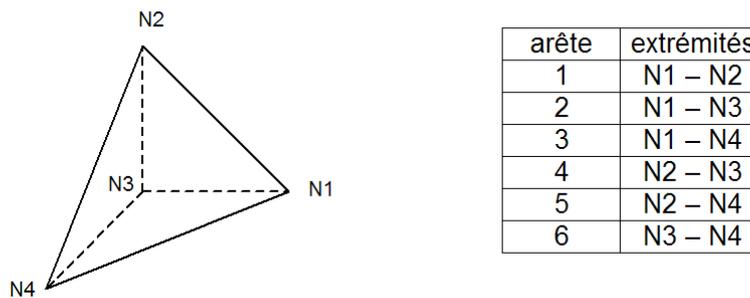


Figure 3.3.3-1 : Tétraèdre de référence

L'approximation de la level set utilisant les fonctions de forme du tétraèdre, la surface $l_{sn}^h = 0$ est alors un plan. Le problème se ramène donc à la découpe d'un tétraèdre par un plan. Examinons les différents cas possibles suivant la valeur de n (nombre de points d'intersection P_i). On peut déjà éliminer les cas triviaux où aucun sous-découpage n'est nécessaire :

lorsque $n < 3$ la trace du plan dans le tétraèdre est un sommet ou une arête,

lorsque $n = 3$ et que les 3 points d'intersection sont des point sommets, la trace du plan dans le tétraèdre est une face du tétraèdre.

Dans ces deux cas, on obtient un seul sous-tétra, qui correspond au tétraèdre.

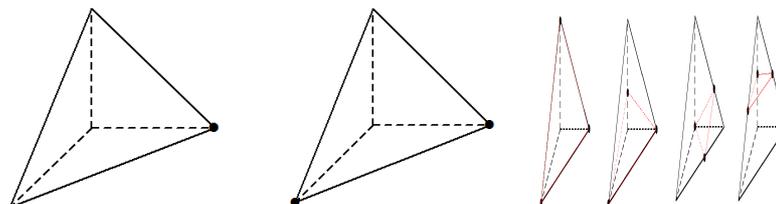


Figure 3.3.3-2 : Cas sans sous-découpage

3.3.3.1 3 points d'intersection dont 2 points sommet

$P1$ et $P2$ sont forcément les deux points sommet (donc $A1=A2=0$). D'après $A3$, numéro d'arête correspondant à $P3$, on peut déterminer les 2 extrémités de cette arête, soient $E1$ et $E2$.

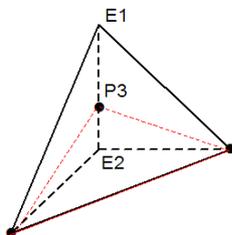


Figure 3.3.3.1-1 : Cas général où $n=3$ dont 2 points sommet

On obtient un sous-tétraèdre de chaque coté de l'interface, soit au final 2 sous-tétraèdres.

2 sous-tétraèdres
$P1 P2 P3 E1 + P1 P2 P3 E2$

Tableau 3.3.3.1-1 : Sous-tétraèdres

3.3.3.2 3 points d'intersection dont un point sommet

$P1$ est forcément le point sommet (donc $A1=0$ et $A2 \neq 0$). Les arêtes $A2$ et $A3$ ont un nœud en commun, $E1$, et 2 nœuds différents $E2$ et $E3$.

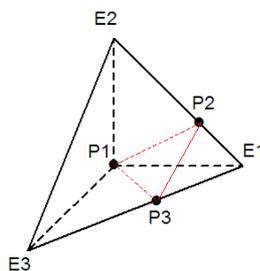


Figure 3.3.3.2-1 : Cas général où $n=3$ dont un point sommet

On obtient un sous-tétraèdre d'un coté, et de l'autre une pyramide que l'on divise en 2 sous-tétraèdres, soit au final 3 sous-tétraèdres.

3 sous-tétraèdres
$P1 P2 P3 E1 + P1 P2 P3 E3 + P1 P2 E2 E3$

Tableau 3.3.3.2-1 : Sous-tétraèdres

3.3.3.3 3 points d'intersection dont aucun point sommet

Dans chacune des 4 configurations répertoriées, on obtient un sous-tétraèdre et un sous-pentaèdre (Figure 3.3.3.3-1). Ces configurations sont déterminées suivant les numéros des arêtes intersectées. Par exemple, la configuration n°2 est celle où le point $P1$ est sur l'arête 1, le point $P2$ est sur l'arête 4 et le point $P3$ sur l'arête 5, soit $A1=1$, $A2=4$ et $A3=5$.

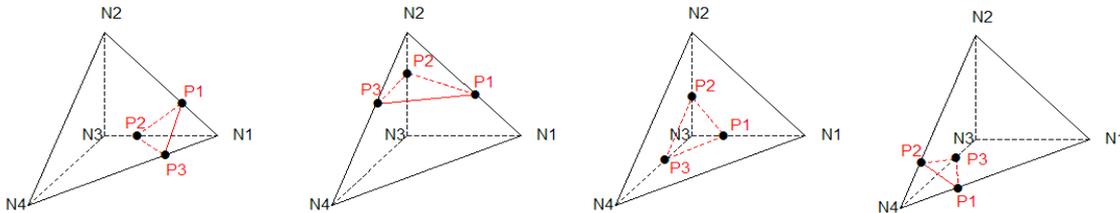


Figure 3.3.3.3-1 : Configurations d'un tétraèdre avec 3 arêtes intersectées

configuration	un sous-tétraèdre + un sous-pentaèdre
1	$P1 P2 P3 N1 + P3 P1 P2 N4 N2 N3$
2	$P1 P2 P3 N2 + N1 N3 N4 P1 P2 P3$
3	$P1 P2 P3 N3 + N4 N2 N1 P3 P2 P1$
4	$P1 P2 P3 N4 + N1 N2 N3 P1 P2 P3$

Tableau 3.3.3.3-1 : Sous-découpage suivant les configurations

Le sous-pentaèdre est divisé en 3 sous-tétraèdres d'après le Tableau 3.3.2-1, ce qui fait au final 4 sous-tétraèdres.

3.3.3.4 4 points d'intersection

Dans chacune des 3 configurations répertoriées, on obtient 2 sous-pentaèdres (Figure 3.3.3.4-1). Ces configurations sont déterminées suivant les numéros des arêtes intersectées. Par exemple, la configuration n°1 est celle où le point $P1$ est sur l'arête 1, le point $P2$ est sur l'arête 2, le point $P3$ sur l'arête 5, et le point $P4$ sur l'arête 6, soit $A1=1$, $A2=2$, $A3=5$ et $A4=6$.

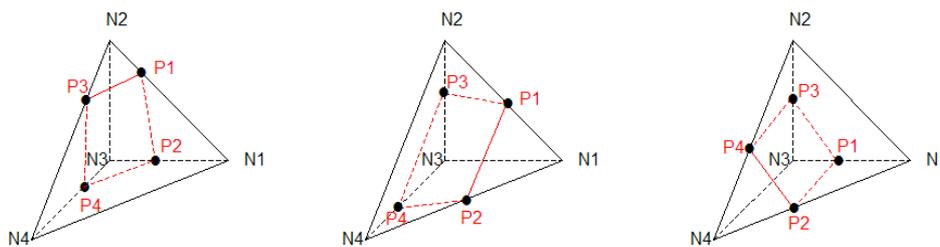


Figure 3.3.3.4-1 : Configurations d'un tétraèdre avec 4 arêtes intersectées

configuration	2 sous-pentaèdres
1	$P4 P2 N3 P3 P1 N2 + N1 P1 P2 N4 P3 P4$
2	$P1 N2 P3 P2 P4 N4 + P2 P1 N1 P4 P3 N3$
3	$P1 P3 N3 P2 P4 N4 + N2 P4 P3 N1 P2 P1$

Tableau 3.3.3.4-1 : Sous-découpage suivant les configurations

Les 2 sous-pentaèdres sont divisés en 3 sous-tétraèdres comme indiqué au Tableau 3.3.2-1, ce qui fait au final, 6 sous-tétraèdres.

3.3.4 Multi-découpage

Lorsque l'on veut modéliser des jonctions, des intersections ou simplement que deux fissures sont assez proches pour découper le même élément, il faut être capable de diviser l'élément en domaines qui respectent toutes les discontinuités introduites.

La stratégie retenue consiste à découper l'élément plusieurs fois séquentiellement. C'est une stratégie qui a pour mérite d'être assez rapide à implémenter, car la découpe d'un tétraèdre de référence par une fissure engendre des sous-tétraèdres qui peuvent à leur tour être considérés comme des tétraèdres de référence pour la découpe par la fissure suivante.

Le problème est que l'on n'optimise pas le nombre de sous-éléments total engendré, qui risque d'être très élevé si on redécoupe plus de 3 fois en 3D. Pour résoudre ce problème, il faudrait pouvoir découper directement n'importe quel type d'élément (hexaèdre, pentaèdre, pyramide, tétraèdre) en combinaison de tous ces type d'éléments. Cela nécessite des développements assez lourds et n'est donc pas envisageable. Une autre stratégie consisterait à regrouper les sous-éléments par zone et à trouver le nombre de points de gauss (et les poids) optimal par zone. On ne stockerait plus les sous-éléments mais directement les zones avec les points de gauss associés.

3.3.5 Nombre maximum de sous-éléments

Afin de dimensionner correctement les structures de données relatives au sous-découpage, il convient de déterminer le nombre maximum de sous-éléments engendrés par la phase de sous-découpage, suivant le type de la maille initiale.

On considère dans ce paragraphe que l'élément n'est découpé que par une seule fissure.

3.3.5.1 Cas du tétraèdre

Le cas engendrant le plus grand nombre de sous-éléments est celui décrit au paragraphe [§3.3.3.4], qui aboutit à 6 sous-éléments.

3.3.5.2 Cas du pentaèdre

Un pentaèdre étant au préalable divisé en trois tétraèdres, on pourrait penser que le plus grand nombre de sous-éléments engendrés est celui où chacun de ces trois tétraèdres est à son tour sous-découpé en 6 sous-éléments ; ce qui entraînerait un découpage final en 18 sous-éléments. Cependant, un tel cas de figure est impossible. Au maximum, sur les trois tétraèdres, deux seront découpés en 6 sous-éléments et un seul sera découpé en 4 sous-éléments, aboutissant à 16 sous-éléments.

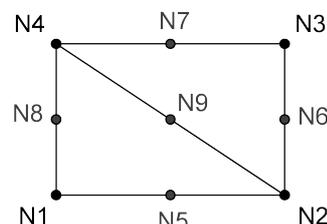
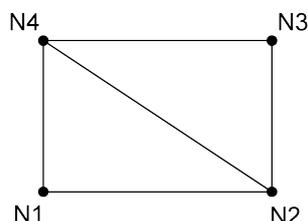
3.3.5.3 Cas de l'hexaèdre

Comme précédemment, le cas de figure où tous les six tétraèdres sont chacun sous-découpés en 6 sous-éléments est impossible. Le cas maximum est celui qui correspond au cas évoqué paragraphe précédent : les deux pentaèdres déduits de l'hexaèdre (voir Figure 3.3.1-1) sont sous-découpés chacun en 16 sous-éléments. Le nombre de sous-éléments est donc $(6+6+4)+(6+6+4)=32$.

3.3.6 Sous-découpage 2D

On utilise une méthode comparable au découpage 3D pour le découpage 2D. Les quadrangles seront subdivisés en triangles eux mêmes sous découpés en fonction du passage de la fissure.

On découpe les quadrangles en 2 triangles :



**Figure 3.3.6-1 : Sous-découpage d'un quadrangle en triangles :
linéaire (à gauche), quadratique (à droite)**

quadrangles	triangles
$N1\ N2\ N3\ N4$	$N1\ N2\ N4$
	$N2\ N3\ N4$
$N1\ N2\ N3\ N4\ N5\ N6\ N7\ N8$	$N1\ N2\ N4\ N5\ N9\ N8$
	$N2\ N3\ N4\ N6\ N7\ N9$

Puis on recoupe les triangles en fonction du passage de la fissure. On effectue le même tri qu'en 3D pour les points d'intersection P_i .

Remarque :

Le nœud $N9$ n'est pas un nœud du maillage. Il est rajouté pour faire la connectivité entre le quadrangle et les triangles quadratiques. Il est défini comme étant le milieu du segment $N2N4$ (et pas le nœud central du quadrangle).

Lorsque l'on a 1 ou 2 points d'intersection qui tombe uniquement sur les sommets, aucun découpage n'est nécessaire.

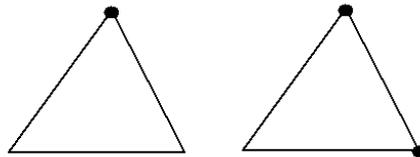


Figure 3.3.6-2 : Cas sans sous-découpage

Si 1 ou 2 des points d'intersection ne tombe pas sur un sommet, on recoupe les triangles.

Si on a 1 point sommet et 1 point non sommet, $P1$ est forcément sommet. On détermine les sommets de l'arête du 2^e point $N2$ et $N3$ pour découper le triangle en 2 sous triangles $P1N2P2$ et $P1N3P2$. Si le triangle est quadratique, on détermine les points milieux de l'arête du 2^e point $Q1$ et $Q2$ et celui de la fissure $Q3$.



**Figure 3.3.6-3 : Cas de découpage linéaire (à gauche) et quadratique (à droite)
avec 2 points d'intersection dont 1 sommet**

Avec 2 points non sommet, on doit découper en 3 triangles. On obtient les triangles $N1P1P2$, $P1P2N3$ et $P1N2N3$. $N1$ étant le point proche des 2 points d'intersections d'après le tri effectué. Si le triangle est quadratique, on détermine les points milieux de l'arête du 1^{er} point $Q1$ et $Q2$, ceux du 2^e point $Q3$ et $Q4$, celui de la fissure $Q5$ et celui de la nouvelle arête $Q6$.



Figure 3.3.6-4 : Cas de découpage linéaire (à gauche) et quadratique (à droite) avec 2 points d'intersection sans sommet

3.3.7 Sous-découpage et erreur de discrétisation

Pour déterminer les intersections entre les arêtes des tétraèdres et la surface de la fissure, on utilise l'interpolation de la level set normale de manière linéaire sur les arêtes (Figure 3.3.7-1). Cela n'est pas conforme avec l'approximation de la level set normale sur l'élément volumique (à part pour les tétraèdres). Il faudrait utiliser l'approximation sur l'élément volumique dans tous les cas. Ce problème peut être facilement résolu.

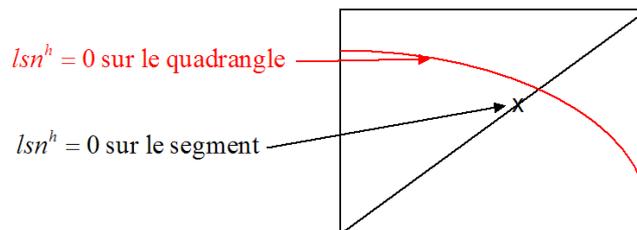


Figure 3.3.7-1 : Erreur d'interpolation de la level set normale

Mettons-nous dans le cas où on a effectivement utilisé l'approximation sur l'élément volumique pour déterminer les intersections entre les arêtes des tétraèdres et la surface de la fissure. Les faces des sous-éléments créés pour l'intégration étant planes, on ne respecte pas vraiment la frontière $l_{SN}^h = 0$. Les quantités intégrées sur la région hachurée (Figure 3.3.7-2) sont donc fausses. De même que précédemment, ce problème se rencontre seulement pour les pentaèdres et les hexaèdres.

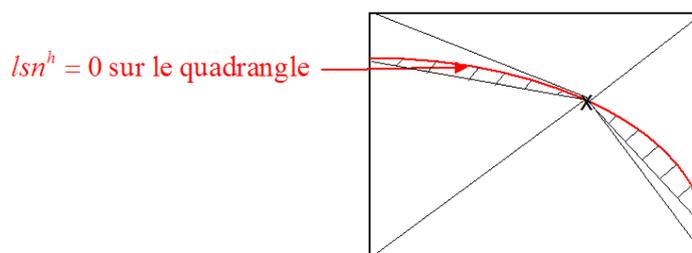


Figure 3.3.7-2 : Erreurs d'intégration dues à la discrétisation

Nicolas Moës propose de modifier l'approximation des level sets par une approximation linéaire par morceaux, de manière à ce que l'on ait de manière exacte $l_{SN}^h = 0$ sur les faces des sous-éléments délimitant la frontière. Cela nécessite un découpage conforme en sous-éléments. Or notre procédé de sous-découpage ne permet pas d'obtenir à coup sûr un maillage conforme. Par exemple, rien n'exclut

de se retrouver dans la configuration de la Figure 3.3.7-3 où les diagonales de deux pentaèdres accolés se croisent.

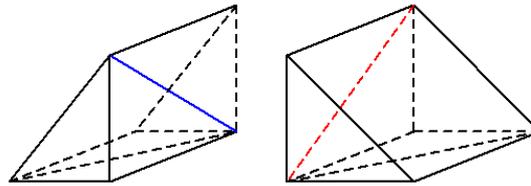


Figure 3.3.7-3 : Incompatibilité du sous-découpage

Pour cela, on pourrait introduire lors du sous-découpage un point au centre de l'élément volumique (le barycentre par exemple) qui servira de sommet aux sous-tétraèdres. Le seul avantage de cette méthode est d'intégrer proprement de part et d'autre de l'iso-zéro de la level set normale. Ceci n'a d'intérêt qu'en 3D et pour des fissures non-planes. En effet, le sous-découpage obtenu en 2D est toujours conforme, et pour les fissures planes, les faces des sous-éléments délimitent exactement la frontière $l_{SN}^h = 0$.

À ce propos, il faudrait pouvoir quantifier l'erreur commise par la méthode actuelle. Mais on considère que celle-ci ne dépasse par l'erreur d'approximation liée à la représentation de la vraie fissure par des surfaces discrétisées (telle qu'on la définit sur la Figure 2.2.3-1).

3.4 Intégration de la rigidité

3.4.1 Intégrande du terme de rigidité mécanique

L'approximation X-FEM du champ de déplacement est écrite sous forme vectorielle :

$$\{u^h\} = \left[\phi_i \quad \phi_j H \quad \phi_k F^1 \quad \phi_k F^2 \quad \phi_k F^3 \quad \phi_k F^4 \right] \cdot \begin{pmatrix} a_i \\ b_j \\ c_k^1 \\ c_k^2 \\ c_k^3 \\ c_k^4 \end{pmatrix}$$

Soit encore :

$$\{u^h\} = [N] \cdot \{u\}$$

où $[N]$ est le vecteur de la base enrichie des fonctions de forme et $\{u\}$ le vecteur des ddl's nodaux de déplacement.

Les déformations s'écrivent :

$$\{\varepsilon\} = [B] \cdot \{u\}$$

où $[B]$ est la matrice des dérivées des fonctions de formes

$$[B] = \left[\left(\phi_i \right)', \left(\phi_j \right)' H \left(\phi_k \right)' F^\alpha + \phi_k \left(F^\alpha \right)' \right], \quad \alpha = 1, 4$$

Sur un élément fini X-FEM occupant un domaine Ω_e , l'expression de la matrice de rigidité élémentaire est (d'après l'expression éq 3.1.4 du PTV) :

$$[K_e] = \int_{\Omega_e} [B]^t [D] [B] d\Omega_e$$

On subdivise l'intégrale en une somme d'intégrales sur les sous-éléments :

$$[K_e] = \sum_{\text{sous-éléments}} \int_{\Omega_{se}} [B]^t [D] [B] d\Omega_{se}$$

Notons au passage que les quantités à intégrer ne changent pas, et font toujours référence à l'élément fini X-FEM Ω_e (appelé « élément parent ») et non pas aux sous-éléments Ω_{se} .

Sur chaque sous-élément, l'intégrande est continue. En effet, les sous-éléments respectent la discontinuité de la fissure, donc la fonction $H(x)$ y est constante, et les fonctions ϕ_i' , F^α et $(F^\alpha)'$ y sont continues.

3.4.2 Intégrande du terme de rigidité géométrique

Lorsqu'on se place dans le cadre des grandes rotations ou des grandes déformations, il faut prendre en compte la rigidité géométrique. La rigidité géométrique est aussi utile dans le cas de l'analyse modale.

Dans le cas des grandes rotations par exemple, le travail virtuel des forces internes peut être écrit sur la configuration initiale sous la forme :

$$\delta \pi_{\text{int}} = \int_{\Omega} (\mathbf{S}(\mathbf{E}) : \delta \mathbf{E}^*) d\Omega$$

où E et S sont les vecteurs de déformation de Green-Lagrange et de contrainte Piola-Kirchhoff de deuxième espèce respectivement.

La variation itérative du travail virtuel s'écrit alors :

$$\Delta \delta \pi_{\text{int}} = \int_{\Omega} (\Delta \mathbf{S}(\mathbf{E}) : \delta \mathbf{E}^* + \mathbf{S}(\mathbf{E}) : \Delta \delta \mathbf{E}^*) d\Omega$$

et en utilisant le fait que :

$$[E] = \frac{1}{2} (\tilde{N}u + \tilde{N}u^T + \tilde{N}u^T \tilde{N}u)$$

$$[\Delta E] = \frac{1}{2} (\nabla \Delta u + \nabla \Delta u^T + \nabla \Delta u^T \nabla u^T + \nabla u^T \nabla \Delta u)$$

et :

$$[\delta \Delta E] = \frac{1}{2} (\nabla \Delta u^T \nabla \delta u + \nabla \delta u^T \nabla \Delta u)$$

la variation du travail virtuel devient:

$$\Delta \delta \pi_{\text{int}} = \int_{\Omega} (\Delta \mathbf{S}(\mathbf{E}) : \delta \mathbf{E}^* + \text{tr}(\nabla^T u^* \cdot \mathbf{S} \cdot \nabla \delta u)) d\Omega$$

L'intégrale sur un élément X-FEM du deuxième terme de cette équation qui s'écrit (en 2D):

$$\int_{\Omega_e} \text{tr}(\nabla^T u^* \cdot \mathbf{S} \cdot \nabla \delta u) d\Omega_{se} = \sum_{se} \int_{\Omega_{se}} \sum_{m,n} (a_X^* \quad a_Y^* \quad b_X^* \quad b_Y^* \quad c_X^{\alpha*} \quad c_Y^{\alpha*})_m \cdot [K_{se}^{geom}]_{m,n} \cdot \begin{pmatrix} \delta a_X \\ \delta a_Y \\ \delta b_X \\ \delta b_Y \\ \delta c_X^\beta \\ \delta c_Y^\beta \end{pmatrix}_n d\Omega_{se}$$

fait apparaître la matrice de rigidité géométrique du sous-élément :

$$[K_{se}^{geom}]_{m,n} = \begin{pmatrix} \phi_i^m S_{ij} \phi_j^n & 0 & H \phi_i^m S_{ij} \phi_j^n & 0 & \phi_i^m S_{ij} (\phi_j^n F^\beta + \phi_j^n F_j^\beta) & 0 \\ 0 & \phi_i^m S_{ij} \phi_j^n & 0 & H \phi_i^m S_{ij} \phi_j^n & 0 & \phi_i^m S_{ij} (\phi_j^n F^\beta + \phi_j^n F_j^\beta) \\ H \phi_i^m S_{ij} \phi_j^n & 0 & H^2 \phi_i^m S_{ij} \phi_j^n & 0 & H \phi_i^m S_{ij} (\phi_j^n F^\beta + \phi_j^n F_j^\beta) & 0 \\ 0 & H \phi_i^m S_{ij} \phi_j^n & 0 & H^2 \phi_i^m S_{ij} \phi_j^n & 0 & H \phi_i^m S_{ij} (\phi_j^n F^\beta + \phi_j^n F_j^\beta) \\ (\phi_i^m F^\alpha + \phi_i^m F_j^\alpha) S_{ij} \phi_j^n & 0 & (\phi_i^m F^\alpha + \phi_i^m F_j^\alpha) S_{ij} H \phi_j^n & 0 & (\phi_i^m F^\alpha + \phi_i^m F_j^\alpha) S_{ij} (\phi_j^n F^\beta + \phi_j^n F_j^\beta) & 0 \\ 0 & (\phi_i^m F^\alpha + \phi_i^m F_j^\alpha) S_{ij} \phi_j^n & 0 & (\phi_i^m F^\alpha + \phi_i^m F_j^\alpha) S_{ij} H \phi_j^n & 0 & (\phi_i^m F^\alpha + \phi_i^m F_j^\alpha) S_{ij} (\phi_j^n F^\beta + \phi_j^n F_j^\beta) \end{pmatrix}$$

où n et m correspondent aux numéros des nœuds dans l'élément parent.

3.4.3 Calcul des dérivées des fonctions singulières :

On cherche les dérivées des fonctions singulières par rapport aux coordonnées globales (x, y, z) . Les fonctions singulières sont données en fonction des coordonnées polaires (r, θ) .

Dérivées des fonctions singulières dans la base polaire (r, θ) :

$$\begin{pmatrix} \frac{\partial F^1}{\partial r} & \frac{\partial F^1}{\partial \theta} \\ \frac{\partial F^2}{\partial r} & \frac{\partial F^2}{\partial \theta} \\ \frac{\partial F^3}{\partial r} & \frac{\partial F^3}{\partial \theta} \\ \frac{\partial F^4}{\partial r} & \frac{\partial F^4}{\partial \theta} \end{pmatrix} = \begin{pmatrix} \frac{1}{2\sqrt{r}} \sin \frac{\theta}{2} & \frac{\sqrt{r}}{2} \cos \frac{\theta}{2} \\ \frac{1}{2\sqrt{r}} \cos \frac{\theta}{2} & -\frac{\sqrt{r}}{2} \sin \frac{\theta}{2} \\ \frac{1}{2\sqrt{r}} \sin \frac{\theta}{2} \sin \theta & \frac{\sqrt{r}}{2} \cos \frac{\theta}{2} \sin \theta + \sqrt{r} \sin \frac{\theta}{2} \cos \theta \\ \frac{1}{2\sqrt{r}} \cos \frac{\theta}{2} \sin \theta & -\frac{\sqrt{r}}{2} \sin \frac{\theta}{2} \sin \theta + \sqrt{r} \cos \frac{\theta}{2} \cos \theta \end{pmatrix}$$

Dérivées des fonctions singulières dans la base locale (e_1, e_2, e_3) :

$$\begin{pmatrix} \frac{\partial F^1}{\partial x_1} & \frac{\partial F^1}{\partial x_2} & \frac{\partial F^1}{\partial x_3} \\ \frac{\partial F^2}{\partial x_1} & \frac{\partial F^2}{\partial x_2} & \frac{\partial F^2}{\partial x_3} \\ \frac{\partial F^3}{\partial x_1} & \frac{\partial F^3}{\partial x_2} & \frac{\partial F^3}{\partial x_3} \\ \frac{\partial F^4}{\partial x_1} & \frac{\partial F^4}{\partial x_2} & \frac{\partial F^4}{\partial x_3} \end{pmatrix} = \begin{pmatrix} \frac{\partial F^1}{\partial r} & \frac{\partial F^1}{\partial \theta} \\ \frac{\partial F^2}{\partial r} & \frac{\partial F^2}{\partial \theta} \\ \frac{\partial F^3}{\partial r} & \frac{\partial F^3}{\partial \theta} \\ \frac{\partial F^4}{\partial r} & \frac{\partial F^4}{\partial \theta} \end{pmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ \frac{1}{r} & \frac{1}{r} & 0 \end{bmatrix}$$

Dérivées des fonctions singulières dans la base globale (E_1, E_2, E_3) :

$$\begin{pmatrix} \frac{\partial F^1}{\partial x} & \frac{\partial F^1}{\partial y} & \frac{\partial F^1}{\partial z} \\ \frac{\partial F^2}{\partial x} & \frac{\partial F^2}{\partial y} & \frac{\partial F^2}{\partial z} \\ \frac{\partial F^3}{\partial x} & \frac{\partial F^3}{\partial y} & \frac{\partial F^3}{\partial z} \\ \frac{\partial F^4}{\partial x} & \frac{\partial F^4}{\partial y} & \frac{\partial F^4}{\partial z} \end{pmatrix} = \begin{pmatrix} \frac{\partial F^1}{\partial x_1} & \frac{\partial F^1}{\partial x_2} & \frac{\partial F^1}{\partial x_3} \\ \frac{\partial F^2}{\partial x_1} & \frac{\partial F^2}{\partial x_2} & \frac{\partial F^2}{\partial x_3} \\ \frac{\partial F^3}{\partial x_1} & \frac{\partial F^3}{\partial x_2} & \frac{\partial F^3}{\partial x_3} \\ \frac{\partial F^4}{\partial x_1} & \frac{\partial F^4}{\partial x_2} & \frac{\partial F^4}{\partial x_3} \end{pmatrix} [P]_{Ee}^{-1}$$

où $[P]_{Ee}$ est la matrice de passage de la base globale à la base locale, telle que

$$[P]_{Ee} = \begin{bmatrix} e_1 & e_2 & e_3 \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \begin{matrix} E_1 \\ E_2 \\ E_3 \end{matrix}$$

Pour exprimer une quantité dans le base globale, on effectue le produit de cette matrice par la quantité exprimée dans la base locale, soit

$$X_E = [P]_{Ee} X_e$$

Cette matrice étant orthonormée, son inverse s'obtient facilement :

$$[P]_{Ee}^{-1} = [P]_{Ee}^t$$

3.4.4 Schémas d'intégration

3.4.4.1 Intégration des termes non singuliers

Pour les termes non singuliers (termes classiques et enrichissement Heaviside), les fonctions à intégrer sont des polynômes. L'intégration numérique par un schéma de Gauss-Legendre sur chaque sous-tétraèdre est alors bien adaptée. Soit npg le nombre de points de Gauss. Ces points ont pour coordonnées dans le sous-tétraèdre de référence (s_g, t_g, u_g) et le poids associé est noté w_g . Soit J la Jacobienne de la transformation sous-tétraèdre \rightarrow sous-tétraèdre de référence. Son déterminant est constant par sous-élément. La procédure classique s'écrit :

$$\int_{\Omega_{se}} f d\Omega_{se} = \int_{\Omega_{se}^{ref}} f |\det(J)| d\Omega_{se} \approx \sum_{g=1}^{npg} f(s_g, t_g, u_g) w_g |\det(J)|$$

La fonction f étant seulement connue dans le repère lié à l'élément parent de référence, il est nécessaire d'exprimer les coordonnées du point de Gauss (s_g, t_g, u_g) dans ce repère [bib68]. Tout d'abord, on les transporte dans le repère réel (utilisation simple des fonctions de forme du sous-tétraèdre), puis on les transporte dans le repère lié à l'élément parent de référence (résolution d'équations non-linéaires par l'algorithme de Newton-Raphson) :

$$(s_g, t_g, u_g) \xrightarrow{t_1} (x_g, y_g, z_g) \xrightarrow{t_2} (\xi_g, \eta_g, \zeta_g)$$

Remarque :

En effet, il faudrait plutôt écrire $\int_{\Omega_{se}^{ref}} f \circ t_2 \circ t_1(s_g, t_g, u_g) |\det(J)| d\Omega_{se}$. La transformation t_1 est linéaire. Par contre, la transformation t_2 est non-linéaire. C'est l'inverse d'une transformation linéaire. C'est une combinaison de fractions rationnelles et de racines carrées à plusieurs variables. De ce fait, on ne connaît pas l'effet de $f \circ t_2 \circ t_1$ sur l'ordre maximal des monômes dans la cas général. Par contre, pour un élément réel à bords parallèles (losange en 2D), t_2 étant linéaire, on considère que $f \circ t_2 \circ t_1(s_g, t_g, u_g)$ sont des monômes en (s_g, t_g, u_g) du même ordre que ceux définis sur l'élément parent de référence.

Les fonctions de forme des éléments iso-paramétriques sont des monômes de type $\xi^i \eta^j \zeta^k$ avec $i+j+k \leq p$ (voir le Tableau 3.4.4.1-1). Les dérivées des fonctions de forme sont alors des monômes de même type avec $i+j+k \leq p-1$. Les quantités à intégrer sont donc des monômes d'ordre $m = (p-1)^2$. On se réfère à [bib68] pour déterminer le schéma adapté permettant une intégration exacte³ de tous les termes. Pour les éléments à bords non parallèles, l'intégration sera alors approchée.

Élément parent	p	$m = (p-1)^2$	npg
hexaèdre	3	4	15
pentaèdre	2	1	1
tétraèdre	1	0	1

Tableau 3.4.4.1-1 : Schémas d'intégration selon le type de l'élément parent

3.4.4.2 Intégration des termes singuliers

Les schémas d'intégration numérique de Gauss-Legendre ont été conçus pour l'intégration de polynômes. Or la présence de monômes en $1/\sqrt{r}$ provenant de l'enrichissement avec les champs asymptotiques conduit à une convergence très lente de la précision de ces schémas : un nombre considérable de points de Gauss est alors nécessaire pour obtenir une erreur acceptable. Lorsque la singularité se trouve sur la frontière du domaine d'intégration, des changements de variables successifs bien choisis permettent de se ramener à l'intégration sur un domaine unitaire d'une fonction non-singulière, polynomiale (ou quasi-polynomiale), sur laquelle une intégration classique de Gauss-Legendre s'avère efficace et rapidement convergente [bib68]. Une dizaine de points de Gauss dans chaque direction suffit à atteindre les limites de la précision numérique d'un ordinateur. Lorsque la singularité ne se trouve pas sur la frontière, il convient de découper le domaine en sous-domaines, de telle sorte que la singularité se trouve sur les frontières des sous-domaines. Malgré des résultats très satisfaisants en terme de convergence de l'erreur relative d'intégration sur chaque terme [bib68], l'effort d'implémentation à fournir pour les structures 3D nous paraît considérable [bib69]. D'autant plus que la différence au niveau des erreurs globales (après résolution) entre une intégration classique avec un nombre de points de Gauss assez élevé mais raisonnable et une intégration singulière modifiée n'est pas vraiment significative.

Pour le moment, on a mis en place le même schéma classique de Gauss-Legendre que celui qui est utilisé pour l'intégration des termes non singuliers.

3.5 Intégration des seconds membres surfaciques

3.5.1 Intégrande du second membre des efforts surfaciques

Les seconds membres dus aux efforts surfaciques s'écrivent de manière discrétisée :

$$\int_{\Gamma^t} \mathbf{t} \cdot \mathbf{u}^* d\Gamma^t$$

la surface Γ^t discrétisée correspond aux faces des éléments 3D. un sous-découpage possible pour ses faces 2D est celui correspondant à la trace du sous-découpage des éléments 3D. Dans Aster, on choisit de prendre le sous-découpage générique 2D, correspondant à celui décrit au paragraphe [§3.3.6].

De même que pour l'intégration de la rigidité, il faut déterminer les coordonnées des points de Gauss du sous-triangle de référence dans l'élément parent de référence

Soit le point de Gauss de coordonnées (s_g, t_g) dans le sous-triangle de référence. Le problème est que l'environnement de calcul est 3D, On crée donc un repère réel 2D local à chaque sous-triangle

Soit le sous-triangle ABC , de normale $n = \frac{\vec{AB} \wedge \vec{AC}}{\|\vec{AB} \wedge \vec{AC}\|}$. La base locale s'écrit :

$$x_l = \frac{\vec{AB}}{\|\vec{AB}\|}, y_l = n \wedge x_l.$$

Le point de Gauss a alors pour coordonnées locales :

$$(x_g, y_g) = \left(\sum_{i=1}^3 \psi_i(s_g, t_g) x^i, \sum_{i=1}^3 \psi_i(s_g, t_g) y^i \right)$$

où ψ_i sont les fonctions de formes du sous-triangles et (x^i, y^i) les coordonnées des sommets du sous-triangle dans le repère local 2D (x_l, y_l) .

Comme pour l'intégration des termes de rigidité, il faut maintenant utiliser une méthode de Newton-Raphson pour déterminer les coordonnées du point de Gauss dans le repère de référence de l'élément

parent (élément de face). Pour cela, toutes les coordonnées réelles sont exprimées dans le repère local réel 2D.

3.5.2 Intégrande du second membre des efforts surfaciques sur les lèvres de la fissure

Les seconds membres dus aux efforts surfaciques sur les lèvres de la fissure s'écrivent de manière discrétisée :

$$\int_{\Gamma_c} \mathbf{g} \cdot \mathbf{u}^* d\Gamma$$

L'intégration se fait sur les deux lèvres de la fissure ($\Gamma_c = \Gamma^1 + \Gamma^2$), avec pour convention :

la lèvre Γ^1 est telle que la fonction Heaviside est négative et l'angle polaire vaut $-\pi$,

la lèvre Γ^2 est telle que la fonction Heaviside est positive et l'angle polaire vaut π ,

Remarque :

Les normales extérieures aux lèvres étant opposées, les efforts surfaciques équivalents à la pression imposée ont des signes opposés.

L'intégration du second membre dû aux efforts appliqués sur les lèvres se fait sur les facettes des lèvres, grâce à la donnée des objets correspondant à la discrétisation des lèvres en facettes triangulaires (voir [§4.3 dans R5.03.54]). Ces objets ont été créés à l'origine pour la prise en compte du contact.

Seules les options CHAR_MECA_PRES_R et CHAR_MECA_PRES_F sont actuellement programmées, pour les éléments 2D (contraintes planes / déformations planes) et 3D, ce qui signifie qu'il n'est pas possible d'appliquer sur les lèvres des efforts de cisaillement.

En multi-fissuration, on peut avoir des efforts surfaciques différents sur différentes fissures.

4 Thermique linéaire avec X-FEM

4.1 Introduction

La méthode X-FEM a été introduite dans l'opérateur de thermique linéaire afin de pouvoir réaliser des calculs chaînés thermo-mécaniques en prenant en compte la fissure dans la résolution thermique. : le champ de température est alors discontinu à travers la fissure si aucun chargement thermique n'est appliqué sur ses lèvres (avec une condition naturelle de flux nul). Il est également possible de définir une condition d'échange thermique entre les lèvres de la fissure par la donnée d'un coefficient d'échange.

Remarque :

La résolution des problèmes de thermique linéaire avec X-FEM s'appuie sur une algorithmie quasi-identique à celle mise en œuvre pour la résolution de problèmes mécanique de structures fissurées avec X-FEM, détaillée dans la partie 24 « ». En particulier les procédures permettant :

- l'enrichissement de l'approximation du champ des inconnues nodales (statut des nœuds, statut des mailles, annulation des degrés de liberté enrichis à tort) ;
 - le sous-découpage des éléments traversés par une fissure ;
 - les procédures d'intégration des matrices et des vecteurs élémentaires sur ce sous-découpage ;
- sont identiques à celles qui ont été précédemment détaillées, elle ne le seront donc pas dans cette partie.*

4.2 Restrictions

L'ensemble des modélisations et des types de chargement disponibles pour l'algorithme de thermique linéaire transitoire dans le cas de modèles classiques (non X-FEM) n'a pas été intégralement répercuté pour les éléments finis thermiques X-FEM. Toutes ces fonctionnalités restent bien entendu disponibles sur la partie non-enrichie du modèle, et sont documentées dans [bib71]. Des évolutions à venir permettront de venir compléter progressivement la liste des fonctionnalités disponibles à ce jour.

Type de problèmes traités

Résolution de transitoires thermiques linéaires (indépendance à la température de paramètres matériaux), pour un matériau isotrope (conductivité scalaire). Le branchement de fissure (multi-Heaviside) est exclu.

Éléments finis disponibles

Pour le moment, seuls les éléments X-FEM des modélisations 3D, axisymétrique, et plane ont été introduits pour les mailles support linéaires suivantes :

modélisation	mailles support éléments principaux	mailles support éléments de bord
3D	TETRA4, HEXA8, PENTA6, PYRA5	TRIA3, QUAD4
plane / axisymétrique	TRIA3, QUAD4	SEG2

Il n'existe donc pas d'éléments thermiques X-FEM lumpés, et le maillage à partir duquel on définit le modèle enrichi est obligatoirement linéaire.

Chargements thermiques disponibles sur la partie enrichie du modèle

Pour le moment, le seul chargement disponible sur la partie enrichie du modèle est la condition d'échange thermique entre les lèvres de la fissure. Aucune autre condition aux limites ne peut être imposée sur des éléments thermiques X-FEM de bord, ou sur des nœuds appartenant à des éléments thermiques X-FEM. De même, aucun terme source ne peut être imposé sur des éléments thermiques X-FEM principaux. L'utilisateur doit donc s'assurer d'appliquer ses chargements suffisamment « loin » de la fissure, c'est à dire sur des éléments non enrichis (ou des nœuds dont le support n'est formé que d'éléments non enrichis).

4.3 Problème traité

Afin de ne pas alourdir l'expression de la formulation variationnelle, on considère le problème simplifié d'un solide fissuré, constitué d'un matériau linéaire et isotrope, dont une partie de la frontière (suffisamment éloignée de la fissure) est soumise à une température imposée. Le reste de la frontière (fissure exclue) est soumis à une condition de flux nul. Enfin, on impose une condition d'échange thermique entre les lèvres de la fissure.

On considère donc une fissure Γ^c dans un domaine $\Omega \subset \mathbb{R}^{n_{dim}}$, avec $n_{dim} = 2$ ou 3 . On note \mathbf{n} la normale sortante à sa frontière $\partial\Omega$, Γ_1^c et Γ_2^c les lèvres de la fissure de normales sortantes respectives \mathbf{n}_1 et \mathbf{n}_2 , et Γ^d la partie de $\partial\Omega$ soumise à une température imposée (voir Figure 4.3-1). On note \mathbf{x} la variable d'espace, t le temps, et T la température. L'intervalle $]t_0, t_f]$ correspond à l'intervalle de temps considéré pour la résolution du problème.

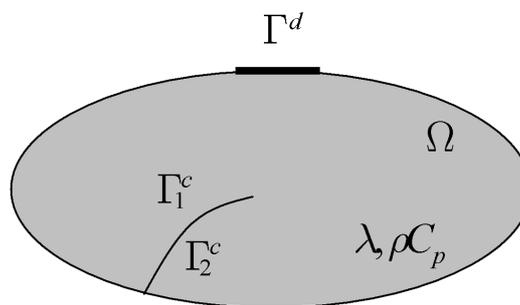


Figure 4.3-1: Notations

Pour le problème considéré, en notant respectivement λ et ρC_p le coefficient de conductivité thermique et la chaleur volumique à pression constante, l'équation de la chaleur s'écrit :

$$-\nabla \cdot (\lambda \nabla T(\mathbf{x}, t)) = \rho C_p \frac{\partial T}{\partial t}(\mathbf{x}, t), \quad \forall (\mathbf{x}, t) \in \Omega \times]t_0, t_f] \quad \text{éq 4.3-1}$$

en notant $T_1 = T|_{\Gamma_1^c}$, $T_2 = T|_{\Gamma_2^c}$, et h le coefficient d'échange thermique entre les lèvres de la fissure, les conditions aux limites s'écrivent :

$$\begin{cases} T = \bar{T} \text{ sur } \Gamma^d \\ \lambda \frac{\partial T_1}{\partial n_1} = h(T_2 - T_1) \text{ sur } \Gamma_1^c \\ \lambda \frac{\partial T_2}{\partial n_2} = h(T_1 - T_2) \text{ sur } \Gamma_2^c \end{cases} \quad \text{éq 4.3-2}$$

avec la la condition initiale :

$$T(\mathbf{x}, 0) = T_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega$$

On introduit ensuite les espaces fonctionnels suivants :

$$V = \{v \in H^1(\Omega), v|_{\Gamma^d} = \bar{T}\} \quad \text{et} \quad V_0 = \{v \in H^1(\Omega), v|_{\Gamma^d} = 0\}$$

La formulation variationnelle associée à éq 4.3-1 et éq 4.3-2 s'écrit alors : trouver $T \in V$ telle que

$$\int_{\Omega} \rho C_p \frac{\partial T}{\partial t} v d\Omega + \int_{\Omega} \lambda \nabla T \cdot \nabla v d\Omega + \int_{\Gamma_1^c} h(T_1 - T_2) v d\Gamma + \int_{\Gamma_2^c} h(T_2 - T_1) v d\Gamma = 0, \forall v \in V_0 \quad \text{éq 4.3-3}$$

La discrétisation en temps est réalisée par un θ -schéma (voir [bib71]). On se place entre deux piquets de temps successifs t_n et t_{n+1} , on note $\Delta t = t_{n+1} - t_n$ le pas de temps, et $\theta \in [0,1]$ le paramètre du θ -schéma. On introduit alors les notations suivantes :

$$T^+ = T(\mathbf{x}, t_{n+1}), \text{ et } T^- = T(\mathbf{x}, t_n)$$

$$V^+ = \{v \in H^1(\Omega), v|_{\Gamma^d} = \bar{T}^+\} \text{ et } V^- = \{v \in H^1(\Omega), v|_{\Gamma^d} = \bar{T}^-\}$$

L'application du θ -schéma à éq 4.3-3 conduit à la formulation variationnelle suivante : étant donnée $T^- \in V^-$ trouver $T^+ \in V^+$ telle que

$$\int_{\Omega} \rho C_p \frac{T^+}{\Delta t} v d\Omega + \int_{\Omega} (\theta \lambda \nabla T^+ \cdot \nabla v d\Omega) + \int_{\Gamma_1^c} \theta h^+(T_1^+ - T_2^+) v d\Gamma + \int_{\Gamma_2^c} \theta h^+(T_2^+ - T_1^+) v d\Gamma$$

$$= \int_{\Omega} \rho C_p \frac{T^-}{\Delta t} v d\Omega - \int_{\Omega} ((1-\theta) \lambda \nabla T^- \cdot \nabla v d\Omega) + \int_{\Gamma_1^c} (1-\theta) h^-(T_2^- - T_1^-) v d\Gamma + \int_{\Gamma_2^c} (1-\theta) h^-(T_1^- - T_2^-) v d\Gamma, \quad \forall v \in V_0 \quad \text{éq 4.3-4}$$

4.4 Approximation X-FEM du champ de température

On utilise les notations précédemment adoptées pour la mécanique au paragraphe 25 « ». On rappelle l'approximation éléments finis classique :

$$T^h(\mathbf{x}) = \sum_{i \in N_n(\mathbf{x})} T_i \varphi_i(\mathbf{x})$$

L'approximation enrichie s'écrit quant à elle :

$$T^h(\mathbf{x}) = \sum_{i \in N_n(\mathbf{x})} T_i^C \varphi_i(\mathbf{x}) + \sum_{j \in N_n(\mathbf{x}) \cap K} T_j^H \varphi_j(\mathbf{x}) H(\text{lsn}(\mathbf{x})) + \sum_{k \in N_n(\mathbf{x}) \cap L} T_k^{CT} \varphi_k(\mathbf{x}) F^1(\text{lsn}(\mathbf{x}), \text{lst}(\mathbf{x}))$$

Cette expression est composée de 3 termes. Le 1^{er} terme est le terme classique où T_i^C désignent les degrés de liberté classiques, le deuxième terme est le terme enrichi par la fonction Heaviside H avec T_j^H les degrés de liberté enrichis correspondants, et le troisième terme est le terme enrichi par la fonction singulière F^1 (enrichissement « Crack-Tip ») avec T_k^{CT} les degrés de liberté enrichis correspondants. On rappelle ci-dessous l'expression de F^1 :

$$F^1(\text{lsn}(\mathbf{x}), \text{lst}(\mathbf{x})) = \sqrt{r} \sin \frac{\theta}{2}$$

où (r, θ) sont les coordonnées polaires dans la base locale au fond de fissure déterminées à partir des level-sets $(\text{lsn}(\mathbf{x}), \text{lst}(\mathbf{x}))$.

Remarque :

En mécanique, quatre fonctions déterminées à partir du développement asymptotique du champ de déplacement en fond de fissure sont introduites dans le terme d'enrichissement singulier. En thermique, on introduit uniquement la fonction F^1 qui est la seule fonction (parmi les quatre) discontinue au travers de la fissure. Ce type d'enrichissement, retenu dans [bib 71] dans le cas d'une condition de flux nul imposée sur les lèvres de la fissure (fissure adiabatique), est ici introduit pour le cas d'une fissure adiabatique comme pour le cas d'une condition d'échange entre les lèvres de la fissure.

4.5 Intégration des matrices et des vecteurs élémentaires

On détaille dans ce paragraphe l'expression des quantités élémentaires associées à chaque terme de la formulation variationnelle éq 4.3-4 du problème de thermique linéaire discrétisé en temps.

4.5.1 Intégrales volumiques

Soit E un élément thermique X-FEM. On note $\{\mathbf{T}\}_E^T$, $[\mathbf{N}]_E$ et $[\mathbf{B}]_E$ respectivement le vecteur élémentaire des inconnues nodales (à l'instant t), la matrice élémentaire des fonctions de formes et la matrice élémentaire des dérivées des fonctions de formes. A titre d'exemple, on peut considérer un élément de dimension $n_{dim}=2$ ou 3 , à n nœuds, et de type « Heaviside / Crack-Tip ». $\{\mathbf{T}\}_E^T$, $[\mathbf{N}]_E$ et $[\mathbf{B}]_E$ s'écrivent dans ce cas :

$$\{\mathbf{T}\}_E^T = [T_1^C \quad T_1^H \quad T_1^{CT} \quad \dots \quad T_n^C \quad T_n^H \quad T_n^{CT}] \text{ de taille } 3n$$

$$[\mathbf{N}]_E = [N_1 \quad HN_1 \quad F^1 N_1 \quad \dots \quad N_n \quad HN_n \quad F^1 N_n] \text{ de taille } (1,3n)$$

$$[\mathbf{B}]_E = [\nabla N_1 \quad \nabla(HN_1) \quad \nabla(F^1 N_1) \quad \dots \quad \nabla N_n \quad \nabla(HN_n) \quad \nabla(F^1 N_n)] \text{ de taille } (n_{dim}, 3n)$$

Toutes les quantités volumiques sont intégrées selon la procédure s'appuyant sur le découpage en sous-éléments, et détaillée dans le cas de la mécanique à la page 46. On donne ci-dessous l'expression de ces quantités élémentaires.

matrice de masse (option MASS_THER) :

$$\int_E \frac{\rho C_p}{\Delta t} [\mathbf{N}]_E^T [\mathbf{N}]_E d\Omega$$

matrice de rigidité (option RIGI_THER) :

$$\int_E \theta \lambda [\mathbf{B}]_E^T [\mathbf{B}]_E d\Omega$$

vecteur relatif à la discrétisation temporelle (option CHAR_THER_EVOL) :

$$\int_E \frac{\rho C_p}{\Delta t} [\mathbf{N}]_E^T [\mathbf{N}]_E \{\mathbf{T}^-\}_E d\Omega - \int_E (1-\theta) \lambda [\mathbf{B}]_E^T [\mathbf{B}]_E \{\mathbf{T}^-\}_E d\Omega$$

4.5.2 Intégrales surfaciques

Ces quantités correspondent à la condition d'échange thermique entre les lèvres de la fissure, et sont intégrées le long des lèvres de la fissure selon la procédure s'appuyant sur la discrétisation des lèvres en « facettes de contact ». Cette procédure est détaillée dans le cas de la mécanique à la page 51.

On introduit tout d'abord les matrices élémentaires $[\tilde{\mathbf{N}}_1]_E$ et $[\tilde{\mathbf{N}}_2]_E$, définies comme les matrices des traces des fonctions de formes respectivement sur Γ_1^c et Γ_2^c :

$$[\tilde{\mathbf{N}}_1]_E = [\mathbf{N}]_{\Gamma_1^c}, \quad [\tilde{\mathbf{N}}_2]_E = [\mathbf{N}]_{\Gamma_2^c}$$

Remarque :

D'une matrice à l'autre, les coefficients correspondant à l'enrichissement classique sont égaux, et les coefficients correspondant à l'enrichissement Heaviside et « Crack-Tip » sont égaux au signe près. En effet $H_{|\Gamma_1^c} = -H_{|\Gamma_2^c}$, et $F_{|\Gamma_1^c}^1 = -F_{|\Gamma_2^c}^1$ car l'angle polaire vaut $\pm\pi$.

On introduit ensuite les matrices de couplage suivantes :

$$[\tilde{\mathbf{N}}_{12}]_E = [\tilde{\mathbf{N}}_1]_E - [\tilde{\mathbf{N}}_2]_E, \quad [\tilde{\mathbf{N}}_{21}]_E = [\tilde{\mathbf{N}}_2]_E - [\tilde{\mathbf{N}}_1]_E$$

Les quantités élémentaires sont alors données par les expressions suivantes, avec Γ_E^c l'ensemble des facettes constituant la discrétisation de la fissure au sein de l'élément E :

matrice relative à la condition d'échange (option RIGI_THER_PARO_R) :

$$\int_{\Gamma_E^c} \theta h^+ [\tilde{\mathbf{N}}_1]_E^T [\tilde{\mathbf{N}}_{12}]_E d\Gamma + \int_{\Gamma_E^c} \theta h^+ [\tilde{\mathbf{N}}_2]_E^T [\tilde{\mathbf{N}}_{21}]_E d\Gamma$$

vecteur relatif à la condition d'échange (option CHAR_THER_PARO_R) :

$$\int_{\Gamma_E^c} (1-\theta) h^- [\tilde{\mathbf{N}}_1]_E^T [\tilde{\mathbf{N}}_{21}]_E \{\mathbf{T}^-\}_E d\Gamma + \int_{\Gamma_E^c} (1-\theta) h^- [\tilde{\mathbf{N}}_2]_E^T [\tilde{\mathbf{N}}_{12}]_E \{\mathbf{T}^-\}_E d\Gamma$$

5 Calcul énergétique des Facteurs d'Intensité des Contraintes

Cette partie rappelle la méthode G -thêta utilisée dans *Code_Aster* pour le calcul du taux de restitution d'énergie en mécanique de la rupture linéaire, suite à un calcul éléments finis classiques (en 2D ou 3D). Cette méthode est aussi utilisée pour le calcul des facteurs d'intensité des contraintes, en utilisant la forme bilinéaire de g mais seulement en 2D. Car pour pouvoir l'utiliser en 3D, il faut connaître la base locale au fond de fissure afin d'exprimer les champs asymptotiques en fonction des coordonnées polaires (r, θ) .

Ensuite, on présente l'apport des level sets en éléments finis classiques, qui a été de pouvoir construire une base locale au fond de fissure en 3D. Comme on l'a vu au paragraphe [§2.3], les gradients des level sets donnent une base locale au fond de fissure. De plus, les coordonnées polaires dans cette base locale s'expriment facilement en fonction des level sets (voir Figure 3.2.3-1). Ainsi grâce aux level sets, la méthode G -thêta a été applicable au calcul des facteurs d'intensité des contraintes en 3D avec des éléments finis classiques. Dans ce contexte, la fissure est maillée, un calcul classique est réalisé. En post-traitement, on détermine les deux champs level sets à partir de la fissure maillée et on en déduit une discrétisation du fond de fissure. Le champ thêta est construit à partir de ce fond de fissure discrétisé, et la méthode G -thêta est appliquée.

Le dernier point est le calcul du taux de restitution d'énergie et des facteurs d'intensité des contraintes, suite à un calcul X-FEM. Dans ce cas, la fissure n'est pas maillée, mais la procédure suivie est quasiment la même : la discrétisation du fond de fissure est déterminée à partir des level sets, un champ thêta est construit et la G -thêta est appliquée. La seule différence se situe au niveau des intégrations numériques : avec X-FEM, les intégrations numériques nécessitent un sous-découpage des éléments à intégrer. En fait, on utilise le sous-découpage réalisée pour le calcul des matrices tangentes et des seconds membres.

5.1 Méthode G-thêta pour le calcul de G

5.1.1 Relations d'équilibre

En élasticité linéaire, on définit la densité d'énergie libre Ψ , en l'absence de déformations et de contraintes initiales par une forme quadratique définie positive des composantes du tenseur des déformations :

$$\Psi(\varepsilon, T) = \frac{1}{2} (\varepsilon - \varepsilon^{\text{th}}) : C : (\varepsilon - \varepsilon^{\text{th}})$$

C est le tenseur de Hooke (tenseur du 4^{ème} ordre), et les deux points désignent le produit tensoriel contracté sur deux indices.

Le tenseur des contraintes σ dérive du potentiel Ψ pour donner la loi d'état (ou loi de comportement du matériau) :

$$\sigma = \frac{\partial \Psi}{\partial \varepsilon}(\varepsilon, T) = C : (\varepsilon - \varepsilon^{\text{th}})$$

Les relations d'équilibre en formulation faible sont obtenues en minimisant l'énergie potentielle globale du système :

$$W(v) = \int_{\Omega} \Psi(\varepsilon(v)) d\Omega - \int_{\Omega} f_i v_i d\Omega - \int_S g_i v_i d\Gamma$$

Cette fonctionnelle étant minimale pour le champ de déplacement u :

$$\begin{aligned} \delta W &= \int_{\Omega} \frac{\partial \Psi}{\partial \varepsilon_{ij}} \delta \varepsilon_{ij} d\Omega - \int_{\Omega} f_i \delta v_i d\Omega - \int_S g_i \delta v_i d\Gamma \\ &= \int_{\Omega} \sigma_{ij} \frac{1}{2} (\delta v_{i,j} + \delta v_{j,i}) d\Omega - \int_{\Omega} f_i \delta v_i d\Omega - \int_S g_i \delta v_i d\Gamma \\ &= \int_{\Omega} \sigma_{ij} \delta v_{i,j} d\Omega - \int_{\Omega} f_i \delta v_i d\Omega - \int_S g_i \delta v_i d\Gamma = 0 \end{aligned}$$

Les relations d'équilibre en formulation faible s'écrivent donc de la manière suivante :
Trouver $u \in V$ tel que

$$\int_{\Omega} \sigma_{ij} v_{i,j} d\Omega = \int_{\Omega} f_i v_i d\Omega + \int_S g_i v_i d\Gamma, \quad \forall v \in V_0$$

5.1.2 Expression lagrangienne du taux de restitution d'énergie

Par définition, le taux de restitution d'énergie locale G est défini par l'opposé de la dérivée de l'énergie potentielle par rapport au domaine :

$$G = - \frac{\partial W}{\partial \Omega}$$

La méthode des extensions virtuelles utilisée ici est une méthode lagrangienne de dérivation de l'énergie potentielle. Elle consiste à introduire des transformations :

$$F^\eta : P \in \Omega \rightarrow M = P + \eta \theta(P) \in \Omega^\eta$$

qui à chaque point matériel P du domaine de référence Ω , associent un point spatial M du domaine transformé Ω^η . Ces transformations représentant des propagations de la fissure, ne doivent modifier que la position du fond de fissure Γ_0 . Les champs θ doivent être tangents à la surface de la fissure.

Soit n la normale à la surface de la fissure, les champs θ doivent vérifier :

$$\theta_j n_j = 0 \quad \text{sur } \Gamma_{cr}$$

Soit m la normale unitaire au fond de fissure, située dans le plan tangent de la fissure.

D'après [bib69], le taux de restitution d'énergie local G est solution de l'équation variationnelle suivante :

$$\int_{\Gamma_0} G \theta \cdot m = G(\theta), \quad \forall \theta \in \Theta \quad \text{éq. 5.1.2-1}$$

où $G(\theta)$ est défini par l'opposé de la dérivée de l'énergie potentielle $W(u(\eta))$ à l'équilibre par rapport à l'évolution initiale du fond de fissure :

$$G(\theta) = - \left. \frac{dW(u(\eta))}{d\eta} \right|_{\eta=0} \quad (\text{dérivée particulière})$$

Pour dériver l'énergie potentielle par rapport à son support (au voisinage du support de référence), on utilise le théorème de transport de Reynolds :

Pour une transformation F^η de classe C^1 et une fonction φ (tenseur d'ordre 0, 1 ou 2) de classe C^1 , en notant $\dot{\varphi}$ la dérivée lagrangienne liée à la variation de domaine, on a la relation suivante :

$$\left. \frac{\partial}{\partial \eta} \left(\int_{\Omega} \varphi d\Omega \right) \right|_{\eta=0} = \int_{\Omega} \dot{\varphi} + \varphi \operatorname{div} \left(\frac{\partial F^\eta}{\partial \eta} \right)_{\eta=0} d\Omega$$

En fait, dans la pratique, la transformation F^η n'est pas toujours de classe C^1 car le champ θ est seulement C^1 par morceaux.

Ainsi,

$$\begin{aligned} -G(\theta) &= \left. \frac{\partial}{\partial \eta} \left(\int_{\Omega} \Psi - f_i u_i d\Omega - \int_S g_i u_i d\Gamma \right) \right|_{\eta=0} \\ &= \int_{\Omega} \overbrace{\Psi - f_i u_i}^{\dot{\Psi} - f_i \dot{u}_i} + (\Psi - f_i u_i) \operatorname{div}(\theta) d\Omega - \int_S \overbrace{g_i u_i}^{\dot{g}_i u_i + g_i \dot{u}_i} \left(\operatorname{div}(\theta) - \frac{\partial \theta}{\partial n_k} n_k \right) d\Gamma \end{aligned}$$

où:

$$\Psi(\varepsilon, T) = \frac{1}{2} \lambda (\varepsilon_{ii})^2 + \mu \varepsilon_{ij} \varepsilon_{ij} - 3 K \alpha (T - T_{réf})$$

$$\dot{\Psi}(\varepsilon) = \frac{\partial \Psi}{\partial \varepsilon_{ij}} \dot{\varepsilon}_{ij} + \frac{\partial \Psi}{\partial T} \dot{T} = \sigma_{ij} \dot{\varepsilon}_{ij} + \frac{\partial \Psi}{\partial T} \dot{T}$$

donc :

$$-G(\theta) = \int_{\Omega} \sigma_{ij} \dot{\varepsilon}_{ij} + \frac{\partial \Psi}{\partial T} \dot{T} - f_i u_i - f_i \dot{u}_i + (\Psi - f_i u_i) \theta_{k,k} d\Omega$$

$$- \int_S \dot{g}_i u_i - g_i \dot{u}_i + g_i u_i \left(\theta_{k,k} - \frac{\partial \theta}{\partial n_k} n_k \right) d\Gamma$$

Premièrement, utilisons la relation donnant la dérivée lagrangienne d'un champ φ en fonction de sa dérivée eulérienne $\frac{\partial \varphi}{\partial \eta}$:

$$\dot{\varphi} = \frac{\partial \varphi}{\partial \eta} + \nabla \varphi \cdot \theta$$

D'après cette relation, la force volumique f étant supposée indépendante de η , c'est-à-dire étant la restriction à Ω de champs définis sur \mathbb{R}^3 , on peut écrire que $\frac{\partial f_i}{\partial \eta} = 0$. Comme il en est de même pour g et T , on a les relations suivantes :

$$\dot{T} = T_{,k} \theta_k$$

$$\dot{f}_i = f_{i,k} \theta_k$$

$$\dot{g}_i = g_{i,k} \theta_k$$

Deuxièmement, utilisons la relation donnant la dérivée lagrangienne du gradient d'un champ en fonction du gradient du champ et du gradient de la dérivée du champ :

$$\dot{\nabla} \varphi = \nabla \dot{\varphi} - \nabla \varphi \cdot \nabla \theta$$

soit en notations indicielles,

$$\dot{\varphi}_{i,j} = \dot{\varphi}_{i,j} - \varphi_{i,p} \theta_{p,j}$$

donc

$$\dot{\varepsilon}_{i,j} = \frac{1}{2} (\dot{u}_{i,j} + \dot{u}_{j,i}) - \frac{1}{2} (u_{i,p} \theta_{p,j} + u_{j,p} \theta_{p,i})$$

On remplace les 2 expressions dans le 1^{er} et on obtient :

$$-G(\theta) = \int_{\Omega} \sigma_{ij} \dot{u}_{i,j} + \frac{\partial \Psi}{\partial T} T_{,k} \theta_k - \sigma_{ij} u_{i,p} \theta_{p,j} - f_{i,k} \theta_k u_i - f_i \dot{u}_i + (\Psi - f_i u_i) \theta_{k,k} d\Omega$$

$$- \int_S g_{i,k} \sigma_k u_i - g_i \dot{u}_i + g_i u_i \left(\theta_{k,k} - \frac{\partial \sigma}{\partial n_k} n_k \right) d\Gamma$$

On peut éliminer les termes en \dot{u} en remarquant que le champ \dot{u} est cinématiquement admissible et satisfait l'équation d'équilibre :

$$\int_{\Omega} \sigma_{ij} \dot{u}_{i,j} d\Omega = \int_{\Omega} f_i \dot{u}_i d\Omega + \int_S g_i \dot{u}_i d\Gamma$$

Ce qui donne l'expression finale de $J(\theta)$:

$$\begin{aligned}
 G(\theta) = & \int_{\Omega} \sigma_{ij} u_{i,p} \theta_{p,j} - \Psi \theta_{k,k} + \frac{\partial \Psi}{\partial T} T_{,k} \theta_k \, d\Omega \\
 & + \int_{\Omega} f_{i,k} \theta_k u_i + f_i u_i \theta_{k,k} \, d\Omega \\
 & + \int_S g_{i,k} \theta_k u_i + g_i u_i \left(\theta_{k,k} - \frac{\partial \theta}{\partial n_k} n_k \right) \, d\Gamma
 \end{aligned}
 \tag{5.1.2-2}$$

Remarque :

Cette forme de l'intégrale J n'utilise pas des intégrales de contour mais des intégrales de domaines.

5.1.3 Discrétisations

On note s l'abscisse curviligne en fond de fissure. Le calcul de la valeur locale de G nécessite de résoudre l'équation variationnelle suivante pour plusieurs champs θ^i :

$$\int_{\Gamma_0} G(s) \theta^i(s) \cdot m(s) \, ds = G(\theta^i) \quad \forall i \in [1, P]$$

Le champ scalaire $G(s)$ est discrétisé sur une base notée $(p_j(s))_{1 \leq j \leq N}$:

$$G(s) = \sum_{j=1}^N G_j p_j(s)$$

De même, les champs θ^i sont discrétisés sur une base notée $(q_k(s))_{1 \leq k \leq M}$. Soit $\bar{\theta}^i$ la trace du champ θ^i sur le fond de fissure Γ_0 : $\bar{\theta}^i(s) = \theta|_{\Gamma_0}(s)$ et soit θ_k^i les composantes de $\bar{\theta}^i(s)$ dans cette base :

$$\bar{\theta}^i(s) = \sum_{k=1}^M \theta_k^i q_k(s)
 \tag{5.1.3-1}$$

En injectant ces expressions dans l'équation variationnelle, il vient :

$$\int_{\Gamma_0} \sum_{j=1}^N G_j p_j(s) \sum_{k=1}^M (\theta_k^i q_k(s)) \cdot m(s) \, ds = G(\theta^i), \quad \forall i \in [1, P]$$

Les G_j peuvent donc être déterminés en résolvant le système linéaire à P équations et N inconnues :

$$\begin{cases}
 \sum_{j=1}^N a_{ij} G_j = b_i, \quad i = 1, P \\
 \text{avec } a_{ij} = \sum_{k=1}^M \theta_k^i \int_{\Gamma_0} p_j(s) q_k(s) \cdot m(s) \, ds \\
 b_i = G(\theta^i)
 \end{cases}$$

Ce système a une solution si on choisit P champs θ^i indépendants tels que : $P \geq N$ et si $M \geq N$. Il peut comporter plus d'équations que d'inconnues, auquel cas il est résolu au sens des moindres carrés.

5.2 Méthode G-thêta pour le calcul de K_I , K_{II} et K_{III} avec les level sets

Ce paragraphe présente l'apport des level sets pour le calcul des facteurs d'intensité des contraintes en 3D. Il est basé sur la séparation des modes mixtes grâce à la forme bilinéaire de g . Cette forme bilinéaire est utilisée dans *Code_Aster* pour le calcul par la méthode G -thêta de K_I , K_{II} en 2D [bib69].

5.2.1 Forme bilinéaire de g

Le calcul des facteurs d'intensité des contraintes peut se faire également par la méthode thêta, en définissant une forme bilinéaire symétrique g avec :

$$g(u, v) = \frac{1}{4} (G(u+v) - G(u-v)) \quad \text{éq. 5.2.1-1}$$

Ainsi, soient deux champs de déplacements u et v , le calcul de $g(u, v)$ s'effectue grâce à l'expression de G donnée par l'équation [éq. 5.1.2-1], dans laquelle les déformations liées à u et v dérivent des champs u et v , et les contraintes liées à u et v sont déduites par la loi de comportement :

$$\begin{aligned} \varepsilon(u) &= \nabla_s u, & \varepsilon(v) &= \nabla_s v \\ \sigma(u) &= C : \varepsilon(u), & \sigma(v) &= C : \varepsilon(v) \end{aligned}$$

En reprenant les notations de [bib69], le terme classique de $J(\theta)$ [éq. 5.1.2-1] est :

$$\begin{aligned} TCLA &= \sigma(u) : (\nabla u \nabla \theta) - \Psi(\varepsilon(u)) \operatorname{div} \theta \\ &= (S2 - S2TH) - \frac{1}{2} (S1 - S1TH) \operatorname{div} \theta \end{aligned}$$

On rappelle qu'en élasticité, l'énergie libre s'exprime en fonctions des coefficients de Lamé λ et μ :

$$\begin{aligned} \Psi &= \frac{1}{2} \lambda (\varepsilon_{ii})^2 + \mu \varepsilon_{ij} \varepsilon_{ij} - \Psi^{th} \\ &= \frac{1}{2} \left[(\lambda + 2\mu) (\varepsilon_{11}^2 + \varepsilon_{22}^2 + \varepsilon_{33}^2) + 2\lambda (\varepsilon_{11} \varepsilon_{22} + \varepsilon_{11} \varepsilon_{33} + \varepsilon_{22} \varepsilon_{33}) + 2\mu (2\varepsilon_{12}^2 + 2\varepsilon_{13}^2 + 2\varepsilon_{23}^2) \right] - \Psi^{th} \end{aligned}$$

avec $\Psi^{th} = 3K \alpha (T - T^{ref}) \operatorname{tr} \varepsilon$

Pour chaque terme $S1$ et $S2$, on va d'abord écrire leur expression habituelle, puis ensuite, on passera à la forme bilinéaire.

5.2.1.1 Expression de $s1$ et $s1TH$

On écrit $S1$ avec l'expression suivante :

$$S1 = C1 \cdot S11 + C2 \cdot S12 + C3 \cdot S13$$

en posant :

$$\begin{cases} C1 = \lambda + 2\mu \\ C2 = \lambda \\ C3 = \mu \end{cases} \quad \text{et} \quad \begin{cases} S11 = u_{1,1}^2 + u_{2,2}^2 + u_{3,3}^2 \\ S12 = 2(u_{1,1} u_{2,2} + u_{1,1} u_{3,3} + u_{2,2} u_{3,3}) \\ S13 = (u_{1,2} + u_{2,1})^2 + (u_{1,3} + u_{3,1})^2 + (u_{2,3} + u_{3,2})^2 \end{cases}$$

Écrivons maintenant la forme bilinéaire associée. L'énergie libre dépend alors de deux champs u et v . Les coefficients matériaux $C1$, $C2$ et $C3$ sont évidemment inchangés, et les nouveaux coefficients $S11$, $S12$ et $S13$ ont pour expressions :

$$\begin{aligned} S11 &= u_{1,1} v_{1,1} + u_{2,2} v_{2,2} + u_{3,3} v_{3,3} \\ S12 &= (u_{1,1} v_{2,2} + u_{2,2} v_{1,1}) + (u_{1,1} v_{3,3} + u_{3,3} v_{1,1}) + (u_{2,2} v_{3,3} + u_{3,3} v_{2,2}) \\ S13 &= (u_{1,2} + u_{2,1})(v_{1,2} + v_{2,1}) + (u_{1,3} + u_{3,1})(v_{1,3} + v_{3,1}) + (u_{2,3} + u_{3,2})(v_{2,3} + v_{3,2}) \end{aligned}$$

L'expression de la partie thermique est :

$$SITH = 3 K \alpha \left((T_u - T_{réf}) \operatorname{tr} \varepsilon(v) + (T_v - T_{réf}) \operatorname{tr} \varepsilon(u) \right)$$

Pour le calcul de K_i , v est le champ asymptotique en mode i , obtenu pour $T_v = T_{réf}$, donc les termes en $(T_v - T_{réf})$ se simplifient :

$$SITH = 3 K \alpha \left((T_u - T_{réf}) \operatorname{tr} \varepsilon(v) \right)$$

5.2.1.2 Expression de s2 et s2TH

Comme pour $S1$, on écrit $S2$ avec l'expression suivante :

$$S2 = C1 . S21 + C2 . S22 + C3 . S23$$

Les coefficients matériaux $C1$, $C2$ et $C3$ sont ceux définis au paragraphe précédent.

Les coefficients $S21$, $S22$ et $S23$ ont pour expressions :

$$\begin{aligned} S21 &= \sum_{k=1}^3 \sum_{p=1}^3 u_{k,k} u_{k,p} \theta_{p,k} \\ S22 &= \sum_{k=1}^3 \sum_{l \neq k}^3 \sum_{p=1}^3 u_{l,l} u_{k,p} \theta_{p,k} \\ S23 &= \sum_{k=1}^3 \sum_{l \neq k}^3 \sum_{\substack{m \neq k \\ m \neq l}}^3 \sum_{p=1}^3 u_{l,m} u_{l,p} \theta_{p,m} + u_{l,m} u_{m,p} \theta_{p,l} \end{aligned}$$

Cette écriture permet facilement de passer à la forme bilinéaire associée. Il suffit de remplacer les termes $u_{i,j} u_{k,l}$ par $\frac{1}{2} (u_{i,j} v_{k,l} + u_{k,l} v_{i,j})$. Si on introduit la notation suivante :

$$B(u_{i,j}, v_{k,l}) = \frac{1}{2} (u_{i,j} v_{k,l} + u_{k,l} v_{i,j})$$

alors les nouveaux coefficients $S21$, $S22$ et $S23$ ont pour expressions :

$$\begin{aligned} S21 &= \sum_{k=1}^3 \sum_{p=1}^3 B(u_{k,k}, v_{k,p}) \theta_{p,k} \\ S22 &= \sum_{k=1}^3 \sum_{l \neq k}^3 \sum_{p=1}^3 B(u_{l,l}, v_{k,p}) \theta_{p,k} \\ S23 &= \sum_{k=1}^3 \sum_{l \neq k}^3 \sum_{\substack{m \neq k \\ m \neq l}}^3 \sum_{p=1}^3 B(u_{l,m}, v_{l,p}) \theta_{p,m} + B(u_{l,m}, v_{m,p}) \theta_{p,l} \end{aligned}$$

L'expression de la partie thermique est :

$$S2TH = \frac{TH1}{2} 3 K \alpha (T_u - T_{réf}) \left(\frac{\partial v_i}{\partial x_j} \frac{\partial \theta_j}{\partial x_i} \right) + \frac{TH1}{2} 3 K \alpha (T_v - T_{réf}) \left(\frac{\partial u_i}{\partial x_j} \frac{\partial \theta_j}{\partial x_i} \right)$$

$$\text{où } TH1 = 1 \text{ en D_PLAN, AXIS et en 3D et } TH1 = \frac{1-2\nu}{1-\nu} \text{ en C_PLAN.}$$

Pour le calcul de K_i , v est le champ asymptotique en mode i , obtenu pour $T_v = T_{réf}$, donc les termes en $(T_v - T_{réf})$ se simplifient :

$$S2TH = \frac{TH1}{2} 3K \alpha (T_u - T_{réf}) \left(\frac{\partial v_i}{\partial x_j} \frac{\partial \theta_j}{\partial x_i} \right)$$

5.2.1.3 Terme surfacique

Le terme supplémentaire dans l'expression de G , dû à l'imposition d'une force surfacique g sur Γ_c de normale extérieure n est le suivant :

$$\begin{aligned} TSUR &= (\nabla g \cdot \theta) u + g \cdot u \left(\operatorname{div} \theta - n \frac{\partial \theta}{\partial n} \right) \\ &= g_{i,k} \theta_k u_i + g_i \cdot u_i \left(\theta_{k,k} - n_k \frac{\partial \theta}{\partial n_k} \right) \end{aligned}$$

Le terme $n \frac{\partial \theta}{\partial n}$ est nul car le gradient du champ θ est orthogonal à n .

Il reste donc :

$$\begin{aligned} TSUR &= (\nabla g \cdot \theta) u + g \cdot u \operatorname{div} \theta \\ &= g_{i,k} \theta_k u_i + g_i \cdot u_i \theta_{k,k} \end{aligned}$$

La forme bilinéaire $g(u, v)$ associée pour le calcul de G et de K_I , K_{II} et K_{III} est :

$$TSUR(u, v) = \frac{1}{2} \left[((\nabla g_u \cdot \theta) v + g_u \cdot v \operatorname{div} \theta) + ((\nabla g_v \cdot \theta) u + g_v \cdot u \operatorname{div} \theta) \right]$$

Pour le calcul de K_i , v est le champ asymptotique en mode i , obtenu pour $g_v = 0$, donc les termes en g_v et ∇g_v se simplifient :

$$TSUR(u, v) = \frac{1}{2} \left[(\nabla g_u \cdot \theta) v + g_u \cdot v \operatorname{div} \theta \right]$$

5.2.1.4 Terme thermique

Le terme supplémentaire dans l'expression de G , dû au champ de température T est le suivant :

$$\begin{aligned} TTHE &= \frac{\partial \Psi}{\partial T} T_{,k} \theta_k = \frac{1}{2} 3 K \alpha \operatorname{tr} \varepsilon(v) \left(\frac{\partial T_u}{\partial x} \theta_x + \frac{\partial T_u}{\partial y} \theta_y + \frac{\partial T_u}{\partial z} \theta_z \right) \\ &\quad + \frac{1}{2} 3 K \alpha \operatorname{tr} \varepsilon(u) \left(\frac{\partial T_v}{\partial x} \theta_x + \frac{\partial T_v}{\partial y} \theta_y + \frac{\partial T_v}{\partial z} \theta_z \right) \end{aligned}$$

Pour le calcul de K_i , v est le champ asymptotique en mode i , obtenu pour $T_v = T_{réf} = \text{cste}$, donc les termes en $TTHE = \frac{-\partial \Psi}{\partial T} (\nabla T \cdot \theta) = \frac{1}{2} 3 K \alpha \operatorname{tr} \varepsilon \left(\frac{\partial T}{\partial x} \theta_x + \frac{\partial T}{\partial y} \theta_y \right)$ se simplifient :

$$TTHE = \frac{1}{2} 3 K \alpha \operatorname{tr} \varepsilon(v) \left(\frac{\partial T_u}{\partial x} \theta_x + \frac{\partial T_u}{\partial y} \theta_y + \frac{\partial T_u}{\partial z} \theta_z \right)$$

5.2.2 Séparation des modes mixtes

La forme bilinéaire symétrique $g(u, v)$ a comme propriété intéressante celle de séparer les trois modes d'ouverture de la fissure. En effet, dans la forme $g(u, v)$, si le champ à gauche est le champ de déplacement solution et si le terme à droite est un champ de déplacement singuliers en fond de fissure u_I^S , u_{II}^S ou u_{III}^S , les facteurs d'intensité des contraintes s'expriment de la manière suivante :

$$K_I = \frac{E}{(1-\nu^2)} g(u, u_I^S) \quad \text{éq. 5.2.2-1}$$

$$K_{II} = \frac{E}{(1-\nu^2)} g(u, u_{II}^S) \quad \text{éq. 5.2.2-2}$$

$$K_{III} = 2\mu g(u, u_{III}^S) \quad \text{éq. 5.2.2-3}$$

Certains auteurs utilisent le concept d'intégrales d'interactions, à la place de la forme bilinéaire de g . Ces deux vocabulaires désignent en fait les même quantités. En effet, en notant I^{u, u_i^S} l'intégrale d'interaction entre le champ solution et le champ singulier en mode I , l'expression de K_I en terme d'intégrale d'interaction [bib69] est :

$$K_I = \frac{E}{2(1-\nu^2)} I^{u, u_i^S}$$

Cette expression pour K_I est identique à celle donnée par l'équation [éq. 5.2.2-1] à un terme multiplicatif près. Si on applique la méthode des extensions virtuelles à la forme bilinéaire de g , l'expression équivalente de l'équation variationnelle [éq. 5.1.2-1] pour les K_i locaux est :

$$\begin{aligned} \frac{(1-\nu^2)}{E} \int_{\Gamma_0} K_I \theta \cdot m \, ds &= g(u, u_I^S)_\theta, \quad \forall \theta \in \Theta \\ \frac{(1-\nu^2)}{E} \int_{\Gamma_0} K_{II} \theta \cdot m \, ds &= g(u, u_{II}^S)_\theta, \quad \forall \theta \in \Theta \\ \frac{1}{2\mu} \int_{\Gamma_0} K_{III} \theta \cdot m \, ds &= g(u, u_{III}^S)_\theta, \quad \forall \theta \in \Theta \end{aligned} \quad \text{éq. 5.2.2-4}$$

On retrouve l'analogie de ces expressions écrites avec des intégrales d'interaction dans [bib70] et [bib69].

On rappelle que le second membre des équations [éq. 5.2.2-4] est la forme bilinéaire de g (voir [éq. 5.2.1-1]) dans laquelle le terme à gauche u est le champ déplacement solution (provenant de la résolution du problème par la méthode des éléments finis) et le terme à droite $u_{i=I, II, III}^S$ est le champ asymptotique (en mode $i=I, II, III$) dont l'expression analytique est donnée par l'équation [éq. 3.2.3-1]. Ces expressions analytiques sont données dans la base locale au fond de fissure (e_1, e_2, e_3) issue des gradients des level sets, or les dérivées intervenants dans $g(u, u_{i=I, II, III}^S)_\theta$ sont faites par rapport au repère global (E_1, E_2, E_3) (voir les expressions des termes $S1$ et $S2$ au paragraphe précédent). Il faut donc procéder à un changement de base.

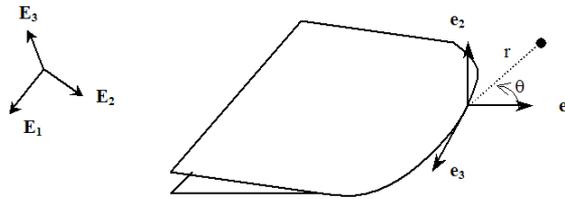


Figure 5.2.2-1 : Coordonnées polaires, base locale et base globale

Soit $DPODL$ la matrice des dérivées des coordonnées polaires dans la base locale :

$$DPODL = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial z} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial z} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\frac{\sin \theta}{r} & \frac{\cos \theta}{r} & 0 \end{bmatrix}$$

Soit u un champ auxiliaire exprimé en fonction des coordonnées polaires (r, θ) . Les dérivées de u par rapport aux variables (r, θ) sont déterminées analytiquement. On les écrit sous forme matricielle :

$$DUDPO = \begin{bmatrix} \frac{\partial u_1}{\partial r} & \frac{\partial u_1}{\partial \theta} \\ \frac{\partial u_2}{\partial r} & \frac{\partial u_2}{\partial \theta} \\ \frac{\partial u_3}{\partial r} & \frac{\partial u_3}{\partial \theta} \end{bmatrix}$$

Ensuite, on écrit les dérivées dans la base locale (e_1, e_2, e_3) :

$$DUDL = \begin{bmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_1}{\partial y} & \frac{\partial u_1}{\partial z} \\ \frac{\partial u_2}{\partial x} & \frac{\partial u_2}{\partial y} & \frac{\partial u_2}{\partial z} \\ \frac{\partial u_3}{\partial x} & \frac{\partial u_3}{\partial y} & \frac{\partial u_3}{\partial z} \end{bmatrix} = DUDPO \cdot DPODL = \begin{bmatrix} \frac{\partial u_1}{\partial r} & \frac{\partial u_1}{\partial \theta} \\ \frac{\partial u_2}{\partial r} & \frac{\partial u_2}{\partial \theta} \\ \frac{\partial u_3}{\partial r} & \frac{\partial u_3}{\partial \theta} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial z} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial z} \end{bmatrix}$$

Il reste maintenant à écrire les dérivées de u dans la base globale (E_1, E_2, E_3) . Pour cela, on note R la matrice des vecteurs de la base globale écrits dans la base locale (matrice de passage) :

$$R_{\alpha m} = (e_\alpha \cdot E_m)$$

Soient i et j des indices globaux, la j ème dérivée de la i ème composante de u s'écrit alors :

$$u_{i,j} = \left(\sum_{\alpha=1}^3 u_\alpha R_{\alpha i} \right)_{,j} = \sum_{\alpha=1}^3 (u_{\alpha,j} R_{\alpha i} + u_\alpha R_{\alpha i,j}) = \sum_{\alpha=1}^3 \left(\sum_{\beta=1}^3 (u_{\alpha,\beta} R_{\beta j}) R_{\alpha i} + u_\alpha R_{\alpha i,j} \right)$$

Dans cette expression, les termes $u_{\alpha,\beta}$ sont les composantes de la matrice $DUDL$. Le terme $R_{\alpha i,j}$ apparaît comme la courbure de la base locale. Il peut être négligé si la courbure du fond de fissure est faible. Dans la pratique, sa prise en compte ne change quasiment pas les résultats.

5.2.3 Discrétisations

De même que précédemment, on note s l'abscisse curviligne en fond de fissure. Le calcul de la valeur locale de K_I , K_{II} et K_{III} nécessite de résoudre l'équation variationnelle suivante pour plusieurs champs θ^i :

$$\begin{aligned} \frac{(1-\nu^2)}{E} \int_{\Gamma_0} K_I(s) \theta^i(s) \cdot m(s) ds &= g(u, u_I^S)_{\theta^i} \quad \forall i \in [1, P] \\ \frac{(1-\nu^2)}{E} \int_{\Gamma_0} K_{II}(s) \theta^i(s) \cdot m(s) ds &= g(u, u_{II}^S)_{\theta^i} \quad \forall i \in [1, P] \quad \text{éq. 5.2.3-1} \\ \frac{1}{2\mu} \int_{\Gamma_0} K_{III}(s) \theta^i(s) \cdot m(s) ds &= g(u, u_{III}^S)_{\theta^i} \quad \forall i \in [1, P] \end{aligned}$$

Les champs scalaires $K_I(s)$, $K_{II}(s)$ et $K_{III}(s)$ sont discrétisés sur la base notée $(p_j(s))_{1 \leq j \leq N}$:

$$\begin{aligned} K_I(s) &= \sum_{j=1}^N K_{Ij} p_j(s) \\ K_{II}(s) &= \sum_{j=1}^N K_{IIj} p_j(s) \\ K_{III}(s) &= \sum_{j=1}^N K_{IIIj} p_j(s) \end{aligned}$$

Pour les champs θ , l'approximation [éq. 5.1.3-1] est conservée.

De même que pour G , en injectant les expressions des discrétisations dans l'équation variationnelle [éq. 5.2.3-1], il vient le système linéaire :

$$\begin{aligned} \int_{\Gamma_0} \sum_{j=1}^N K_{Ij} p_j(s) \sum_{k=1}^M (\theta_k^i q_k(s)) \cdot m(s) ds &= g(u, u_I^S)_{\theta^i}, \quad \forall i \in [1, P] \\ \int_{\Gamma_0} \sum_{j=1}^N K_{IIj} p_j(s) \sum_{k=1}^M (\theta_k^i q_k(s)) \cdot m(s) ds &= g(u, u_{II}^S)_{\theta^i}, \quad \forall i \in [1, P] \\ \int_{\Gamma_0} \sum_{j=1}^N K_{IIIj} p_j(s) \sum_{k=1}^M (\theta_k^i q_k(s)) \cdot m(s) ds &= g(u, u_{III}^S)_{\theta^i}, \quad \forall i \in [1, P] \end{aligned}$$

5.3 Méthode G-thêta avec X-FEM

Jusqu'à présent, la méthode G-thêta a été présentée dans le cadre de la méthode des éléments finis (MEF). Pour adapter la méthode G-thêta au cadre X-FEM, il y a très peu de modifications à apporter. La seule différence réside au niveau des intégrations numériques, pour calculer les intégrales de domaines [éq. 5.1.2-2]. De même que pour l'intégration des matrices tangentes de des seconds membres, on réalise l'intégration sur les sous-éléments.

6 Bibliographie

- 1 COLOMBO D., GIGLIO M., "A methodology for automatic crack propagation modelling in planar and shell FE models", Engineering Fracture Mechanics, vol. 73, pp. 490-504 , 2006
- 2 DHONDT G., "Automatic 3-D mode I crack propagation calculations with finite elements", International Journal for Numerical Methods in Engineering , vol. 41, pp. 739-757, 1998
- 3 H ENSHELL R.D., S HAW K.G., "Crack tip finite elements are unnecessary", International Journal for Numerical Methods in Engineering , vol. 9, pp. 495-507, 1975
- 4 ANDRIER B., GARBAY E., HASNAOUI F., MASSIN P., "Helix-shaped and transverse cracking of rotor shafts based on disk shrunk technology", Actes de la 7^{ème} Conférence Internationale sur la Rupture et Fatigue biaxial/multiaxial, Berlin, Allemagne, 2004
- 5 RÉTHORÉ J., "Méthode éléments finis étendus en espace et en temps : application à la propagation dynamique des fissures", Thèse de Doctorat, Institut National des Sciences Appliquées de Lyon, 2005
- 6 BELYTSCHKO T., KRONGAUZ Y., ORGAN D., FLEMING M., "Meshless methods: an overview and recent developments", Computer Methods in Applied Mechanics and Engineering, vol. 139, pp. 3-47, 1996
- 7 FRIES T-P., MATTHIES H-G ., "Classification and overview of Meshfree methods", informatikbericht Nr.: 2003-3, July, 2004
- 8 LI S.C., CHENG Y.M ., "Enriched meshless manifold method for two-dimensional crack modelling", Theoretical and Applied Fracture Mechanics, vol. 44, pp. 234–248 , 2005
- 9 DUARTE C.A., BABUŠKA I., ODEN J.T., "Generalized finite element method for three-dimensional structural mechanics problems", Computers & Structures, vol. 77, pp. 215-232 , 2000
- 10 DUARTE C.A., HAMZEH O.N., LISZKA T.J., TWORZYDLO W.W., "A generalized finite element method for the simulation of three dimensional dynamic crack propagation", Computer Methods in Applied Mechanics and Engineering , vol. 190, pp. 2227-2262 , 2001
- 11 BELYTSCHKO T., BLACK T., "Elastic crack growth in finite elements with minimal remeshing". International Journal for Numerical Methods in Engineering , Vol.45, Pages 601-620, 1999
- 12 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
- 13 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
- 14 BELYTSCHKO T., MOËS N., USUI S., PARIMI C., "Arbitrary discontinuities in finite elements", International Journal for Numerical Methods in Engineering, vol. 50, pp. 993-1013, 2001
- 15 SUKUMAR N., CHOPP D.L., MOËS N., BELYTSCHKO T., "Modeling holes and inclusions by level sets in the extended finite-element method", Computer methods in applied mechanics and engineering, vol. 190, pp. 6183-6200, 2001
- 16 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

- 17 STOLARSKA M., CHOPP D.L., MOËS N., BELYTSCHKO T., "Modelling crack growth by level sets in the extended finite element method", International Journal for Numerical Methods in Engineering, vol. 51, pp. 943-960, 2001
- 18 MOËS N., GRAVOUIL A., BELYTSCHKO T., "Non-planar 3D crack growth by the extended finite element and level sets - Part I: Mechanical model", International Journal for Numerical Methods in Engineering, vol. 53, pp. 2549-2568, 2002
- 19 "Contact unilatéral par des conditions cinématiques", manuel de référence de *Code_Aster*, document n° [R5.03.50]
- 20 CLOIREC M., "Application d'X-FEM aux calculs parallèles et problèmes multi-échelles", Thèse de Doctorat, École Centrale de Nantes, 2005
- 21 GOMES J., FAUGERAS O., "Reconciling distance functions and level sets", rapport de recherche n°3666 de l'INRIA, 1999
- 22 DAUX C., MOËS N., DOLBOW J., SUKUMAR N., BELYTSCHKO T., "Arbitrary branched and intersectioning cracks with the extended finite element method", International Journal for Numerical Methods in Engineering, vol. 48, pp. 1171-1760, 2000
- 23 MELENK J.M., BABUŠKA I., "The partition of unity finite element method: Basic theory and applications", Computer Methods in Applied Mechanics and Engineering, vol. 139, pp. 289-314, 1996
- 24 MOËS N., DOLBOW J., BELYTSCHKO T., "A finite element method for crack growth without remeshing", International Journal for Numerical Methods in Engineering, vol. 46, pp. 131-150, 1999
- 25 IRWIN G. R., "Analysis of stresses and strains near the end of crack traversing a plate", Journal of Applied Mechanics, Sept. 1957
- 26 SROUBOULIS T., BABUŠKA I., COPPS K., "The design and analysis of the Generalized Finite Element Method", Computer Methods in Applied Mechanics and Engineering, vol. 181, pp. 43-69, 2000
- 27 STAZI F., BUDYN É., CHESSA J., BELYTSCHKO T., "An extended finite element method with high-order elements for curved cracks", Computational Mechanics, vol. 31, pp. 38-48, 2003
- 28 BÉCHET É., MINNEBO H., MOËS N., BURGARDT B., "Improved implementation and robustness study of the X-FEM for stress analysis around cracks", [International Journal for Numerical Methods in Engineering](#), vol. 64, pp. 1033-1056, 2005
- 29 LABORDE P., POMMIER J., RENARD Y., SALAÜN M., "High-order extended finite element method for cracked domains", International Journal for Numerical Methods in Engineering, vl. 64, pp. 354-381, 2005
- 30 MASSIN P., MOËS N., "Étude d'impact de l'implantation de la méthode X-FEM dans le *Code_Aster*", Compte rendu AMA, CR-AMA-03.151, EDF R&D
- 31 GENIAUT S., Convergences en mécanique de la rupture : validation des éléments finis classiques et X-FEM dans *Code_Aster*, note H-T64-2008-00047-FR, 2008
- 32 SUKUMAR N., MOËS N., MORAN B., BELYTSCHKO T., "Extended finite element method for three-dimensional crack modelling", International Journal for Numerical Methods in Engineering, vol. 48, pp. 1549-1570, 2000
- 33 SUKUMAR N., PRÉVOST J.H., "Modeling quasi-static crack growth with the extended finite element method - Part I: Computer implementation", International Journal of Solids and Structures, vol. 40, pp. 7513-7537, 2003
- 34 DHATT G., TOUZOT G., Une présentation de la méthode des éléments finis. 2ème édition, Maloine Ed., PARIS 1983

- 35 MINNEBO H., « Une nouvelle approche numérique pour le calcul de la durée de vie de disques de réacteurs », Thèse de Doctorat, École Centrale de Nantes, 2006
- 36 “Éléments de contacts dérivés d’une formulation hybride continue”, Documentation de Référence de *Code_Aster* n° [R5.03.52]
- 37 BEN DHIA H., VAUTIER I., “Une formulation pour traiter le contact frottement en 3d dans le *Code_Aster* », rapport de recherche HI-75/99/007/A, Juin 1999, EDF
- 38 ALART P., CURNIER A ., “ A mixed formulation for frictional contact problems prone to Newton like solution methods”, *Computer Methods in Applied Mechanics and Engineering*, vol. 92, pp. 353-375, 1991
- 39 JI H., DOLBOW J.E., “On strategies for enforcing interfacial constraints and evaluating jumps conditions with the extended finite element method”, *International Journal for Numerical Methods in Engineering*, vol. 61, pp. 2508-2535, 2004
- 40 PELLET J., “Dualisation des conditions aux limites”, Documentation de Référence de *Code_Aster* n° [R3.03.01]
- 41 ERN A., GUERMOND J.L., *Theory and practice of finite elements*, Springer, 2004
- 42 BRENNER S.C., SCOTT L .R., *The mathematical theory of finite element methods*, 2nd ed., Springer, 2002
- 43 LAURSEN T.A., SIMO J.C. “A continuum element-based formulation for the implicit solution of multi-body, large deformation frictional contact problem”, *International Journal for Numerical Methods in Engineering* , vol. 36, pp. 3451-3485, 1993
- 44 WRIGGERS P., “ Finite element algorithms for contact problem”, *Arch. Of Comp. Meth. In Eng.*, vol. 2, pp. 1-49, 1995
- 45 CURNIER A, HE, Q.C., KLARBRING A., “Continuum mechanics modelling of large deformation contact with friction”, *Contact mechanics*, ed. Plenum Press, 1995
- 46 PIETRZAK G., “continuum mechanics modelling and augmented Lagrangian formulation of large deformation frictional contact problems”, Thèse de doctorat, École Polytechnique Fédérale de Lausanne, 1997
- 47 ALART P., BARBOTEU M., « Éléments contact, méthode de Newton généralisée et décomposition de domaine » *Problèmes non linéaires Appliqués*, École CEA – EDF – INRIA 1999
- 48 « Taux de restitution de l’énergie en thermo-élasticité linéaire », Documentation de Référence de *Code_Aster* n° [R7.02.01]
- 49 « Calcul des coefficients d’intensité de contraintes en thermo-élasticité linéaire plane », Documentation de Référence de *Code_Aster* n° [R7.02.05]
- 50 MIALON P., Calcul de la dérivée d’une grandeur par rapport à un fond de fissure par la méthode thêta, EDF – Bulletin de la Direction des Études et Recherches, Série C, n°3, pp. 1-28 , 1988
- 51 CHAPELLE D., BATHE K. J. , “The Inf-sup test”, *Computers & Structures*, vol. 47, pp. 537-545, 1993
- 52 MOËS N., BÉCHET E., TOURBIER M. , “Imposing essential boundary conditions in the X-FEM”, *International Journal for Numerical Methods in Engineering* , 2006
- 53 GOSZ M., MORAN B., “An interaction energy integral domain method for computation of mixed-mode stress intensity factors along non-planar cracks fronts in three dimensions”, *Engineering Fracture Mechanics*, vol. 69, pp. 299-319, 2002

- 54 GOSZ M., DOLBOW J., MORAN B., "Domain integral formulation for stress intensity factor computation along curved three-dimensional interface cracks", International Journal of Solids and Structures, vol. 35, n°15, pp. 1763-1783, 1998
- 55 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
- 56 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
- 57 CRANDALL M.G., LIONS P.L., " Two approximations of solutions of Hamilton-Jacobi equations", Mathematics of Computation, vol. 43, pp. 1-19, 1984
- 58 DECONINCK H., STRUIJS R., ROE P.L., " Compact Advection Schemes on Unstructured Grids", Technical Report, VKI, VKI LS 1993-04, Computational Fluid Dynamics, 1993
- 59 ROE P.L., " Linear Advection Schemes on Triangular Meshes", Technical Report CoA 8720, Cranfield Institute of Technology, 1987
- 60 ROE P.L., "Optimum Upwind Advection on Triangular Mesh", ICASE 90-75, 1990
- 61 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
- 62 ADALSTEINSSON D., SETHIAN J.A., "A fast level set method for propagating interfaces", Journal of Computational Physics, vol. 118, pp. 269-277, 1995
- 63 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
- 64 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
- 65 MESCHKE G., DUMSTORFF P., « Energy-based modelling of cohesive and cohesionless cracks via X-FEM », Computational Methods in Applied Mechanics Engineering, Vol. 196, pp. 2338-2357, 2007
- 66 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
- 67 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
- 68 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
- 69 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
- 70 DAUX C., MOES N., DOLBOW J., SUKUMAR N., BELYTSCHKO T., « Arbitrary branched and intersecting cracks with the extended finite element method », International Journal for Numerical Methods in Engineering, 48 (2000), 1741-1760
- 71 DANIELE COLOMBO, « 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

- 72 SIAVELIS M., « Modélisation numérique X-FEM de grands glissements avec frottement le long d'un réseau de discontinuités. » Thèse de doctorat, Ecole Centrale de Nantes, 2011
- 73 FRIES T.P., « A corrected XFEM approximation without problems in blending elements », Int. J. Numer. Meth. Engng., 75:503-532, 2008.
- 74 « Algorithme de thermique linéaire transitoire », Documentation de Référence de *Code_Aster* n° [R5.02.01]
- 75 DUFLOT M., « The extended finite element method in thermoelastic fracture mechanics », International Journal for numerical methods in engineering, vol. 74, pp. 827-847, 2008

7 Description des versions

Indice document	Version Aster	Auteur(s) Organisme(s)	Description des modifications
A	9.4	S.GENIAUT, P.MASSIN EDF/R&D AMA	Texte initial
B	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
C	11	S.GENIAUT, EDF/R&D AMA	Séparation des parties sur le contact et la propagation