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.

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]

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

Table des Matières

1Introduction....5
2Représentation de la fissure par des « level sets »....7
2.1Aspect théorique des level sets....7
2.2Calcul des level sets....9
2.2.1Calcul de la level set normale (lsn)....9
2.2.1.1Calcul de la level set normale en 3D....9
2.2.1.2Calcul de la level set normale en 2D....11
2.2.2Calcul de la level set tangente (lst)....13
2.2.3Approximation des level sets....15
2.2.4Réajustement des level sets....18
2.2.5Notion de vraie distance signée....20
2.2.6Multi-fissuration....21
2.3Base locale au fond de fissure....21
2.4Détermination du fond de fissure....22
2.4.1Recherche des points du fond de fissure....22
2.4.2Orientation du fond de fissure....24
2.4.2.1Explication de la méthode d'orientation du fond de fissure....24
2.4.2.2Cas d'un fond de fissure multiple....26
2.4.2.3Algorithme utilisé pour l'orientation du fond....28
3Problème de fissuration avec X-FEM....30
3.1Problème général....30
3.2Enrichissement de l'approximation du déplacement....31
3.2.1Enrichissement avec une fonction de sélection de domaine (2ème terme)....31
3.2.2Enrichissement avec une fonction de sélection de domaine pour un branchement de fissures....32
3.2.3Enrichissement avec les fonctions singulières (3ème terme)....38
3.2.4Enrichissement géométrique....41
3.2.5Enrichissement dans Code_Aster....42
3.2.5.1Enrichissement (statut) des nœuds....43
3.2.5.2Enrichissement (statut) des mailles....45
3.2.5.3Annulation des degrés de liberté enrichis « en trop »....45
3.2.6Conditionnement lié à l'enrichissement....46
3.2.6.1Généralités sur le conditionnement XFEM....46
3.2.6.2Les critères estimatifs pour améliorer le conditionnement....46
3.2.6.3Le pré-conditionneur XFEM pour les matrices....48
3.3Sous-découpage....51
3.3.1Phase préalable de découpe des hexaèdres pour se ramener à des tétraèdres....52

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

3.3.2Phase préalable de découpe des pentaèdres pour se ramener à des tétraèdres.....54
3.3.3Phase préalable de découpe des pyramides pour se ramener à des tétraèdres.....55
3.3.4Sous-découpage d'un tétraèdre en sous-tétraèdres.....56
3.3.4.1Trois points d'intersection dont deux points sommets.....57
3.3.4.2Trois points d'intersection dont un point sommet.....59
3.3.4.3Cinq points d'intersection dont deux points sommets.....59
3.3.4.4Trois points d'intersection dont aucun point sommet.....60
3.3.4.5Quatre points d'intersection dont un point sommet.....61
3.3.4.6Six points d'intersection dont deux points sommet.....62
3.3.4.7Quatre points d'intersection.....62
3.3.4.8Cinq points d'intersection dont un point sommet.....63
3.3.4.9Quatre points d'intersection dont un point sommet.....63
3.3.5Multi-découpage.....64
3.3.6Nombre maximum de sous-éléments.....65
3.3.6.1Cas du tétraèdre.....65
3.3.6.2Cas du pentaèdre.....65
3.3.6.3Cas de l'hexaèdre.....66
3.3.6.4Cas du multi-découpage.....66
3.3.7Sous-découpage 2D.....66
3.3.8Algorithmes de sous-découpage.....68
3.3.8.1Calcul des points d'intersection.....69
3.3.8.2Calcul des points milieu.....73
3.4Récupération des facettes de contact.....83
3.4.1Éléments XH.....83
3.4.2Éléments XHT et XT.....85
3.4.3Les éléments multi-Heaviside.....85
3.4.4Nombre maximum de facettes récupérées.....88
3.4.4.1Cas 2D.....88
3.4.4.2Cas 3D.....88
3.4.4.3Cas multi-Heaviside.....90
3.5Intégration de la rigidité.....90
3.5.1Intégrande du terme de rigidité mécanique.....90
3.5.2Intégrande du terme de rigidité géométrique.....90
3.5.3Calcul des dérivées des fonctions singulières :.....92
3.5.4Schémas d'intégration.....93
3.5.4.1Intégration des termes non singuliers.....93
3.5.4.2Intégration des termes singuliers.....93
3.6Intégration des seconds membres surfaciques.....94
3.6.1Intégrande du second membre des efforts surfaciques.....94

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

3.6.2Intégrande du second membre des efforts surfaciques sur les lèvres de la fissure.....94
4Thermique linéaire avec X-FEM.....97
4.1Introduction.....97
4.2Restrictions.....97
4.3Problème traité.....98
4.4Approximation X-FEM du champ de température.....99
4.5Intégration des matrices et des vecteurs élémentaires.....100
4.5.1Intégrales volumiques.....100
4.5.2Intégrales surfaciques.....101
5Calcul énergétique des Facteurs d'Intensité des Contraintes.....102
5.1Méthode G-thêta pour le calcul de G.....102
5.1.1Relations d'équilibre.....102
5.1.2Expression lagrangienne du taux de restitution d'énergie.....103
5.1.3Discrétisations.....105
5.2Méthode G-thêta pour le calcul de KI, KII et KIII avec les level sets.....106
5.2.1Forme bilinéaire de g.....106
5.2.1.1Expression de S1 et S1TH.....106
5.2.1.2Expression de S2 et S2TH.....107
5.2.1.3Terme surfacique.....108
5.2.1.4Terme thermique.....108
5.2.2Séparation des modes mixtes.....109
5.2.3Discrétisations.....111
5.3Méthode G-thêta avec X-FEM.....111
6Bibliographie.....112
7Description des versions.....116

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

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 [bib1], et dans certains cas en 3D [bib2], 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 [bib3]) 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 [bib4]). 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) [bib5].

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 » [bib6] ont été proposées pour s'affranchir des contraintes liées au maillage. Les meshless sont basées sur une discrétisation uniquement nodales, 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 [bib7]. Même dans les récentes versions des méthodes Meshless combinant des supports physiques et mathématiques [bib8], 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) [bib9], 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.* [bib10] 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.

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

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 [bib11], qui appliquent la partition de l'unité [bib23] 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 [bib24] 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 [bib15]. 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 ([bib15], [bib14] et [bib17]) et en 3D [bib18]. 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 [bib12], couplée avec l'utilisation de la Fast Marching Method [bib16].

Ce document s'articule autour de 4 sections, dont cette introduction qui tient lieu de section 5.
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 [§3.5].

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 [bib48]. 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 intégrale 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 [bib49].

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

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 $\Re^n$. 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 :

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

Cette méthode s'applique aisément aux problèmes de fissuration 2D, notamment dans le cadre des approches où la fissure n'est pas maillée. ([bib14] [bib15] 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 ([bib17] en 2D et [bib18] en 3D) :

- une level set normale (lsn) qui représente la distance à la surface de la fissure (surface étendue par prolongement à tout le domaine),
- une level set tangente (lst) qui représente la distance au fond de fissure.

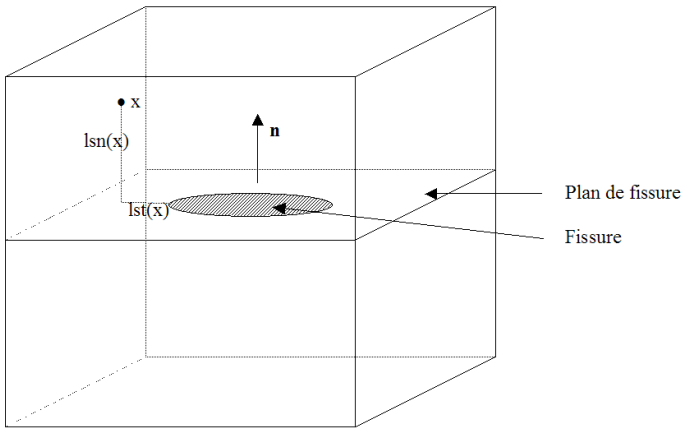


Figure 2.1-1 : Level sets et distance à la fissure

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 $(lsn=0)\cap(lst<0)$. 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 $lsn(x)$ est négatif sont dits « au-dessous » de la fissure, et ceux pour lesquels $lsn(x)$ est positif sont dits « au-dessus » de la fissure (voir Figure 2.1-2 et Figure 2.1-3).

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

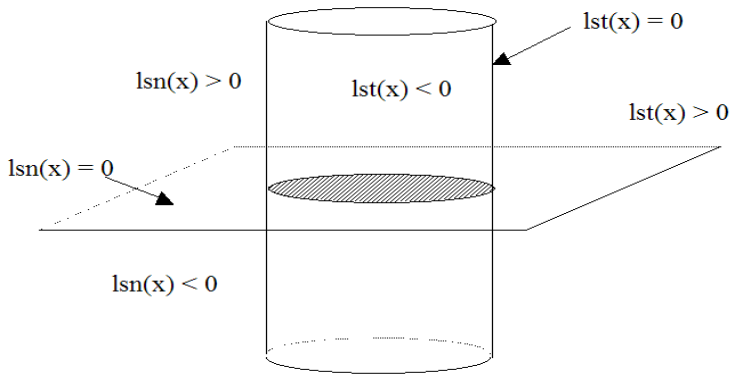


Figure 2.1-2 : Level sets pour la représentation d'une fissure 3D

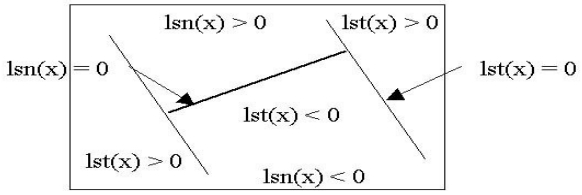


Figure 2.1-3 : Level sets pour la représentation d'une fissure 2D

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 [bib13] :

- •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 [bib12] :

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

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

La Fast Marching Method [bib16] 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].

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

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 lèvre (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 [bib19], 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 [bib19]) et sur un quadrangle (voir le paragraphe [§2.3.3] de [bib19]). 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:

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

- 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: $I(n_{el})=sign\left(I(j)\cdot\vec{n}_{n_{el}}\cdot\vec{n}_j\right)$
            - 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 : $\vec{N}=\overrightarrow{AB}\wedge\overrightarrow{AC}$
    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 $lsn(P)=I\cdot\overrightarrow{PM}\cdot\vec{N}$

  - 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 [bib19].

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

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

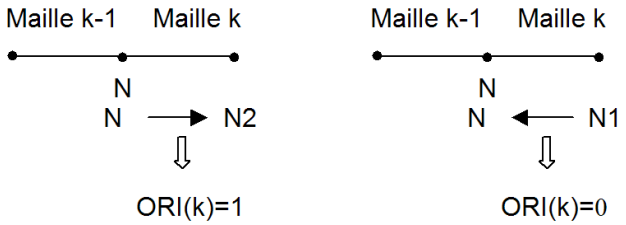


Figure 2.2.1.2-1 : Orientation de la maille suivante

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.

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

•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, car on est en fin de fissure)
•fin de boucle

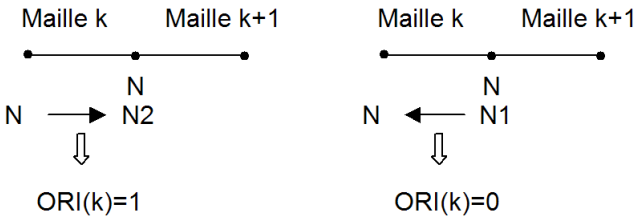


Figure 2.2.1.2-2 : Orientation de la maille précédente

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
$lsn=PM_P.sign(\overrightarrow{AB},\overrightarrow{AP})$
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

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

2.2.2 Calcul de la level set tangente (lst)

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 [bib19]). La valeur de lst 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 lst 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) : $\vec{N}=\overrightarrow{AB}\wedge\overrightarrow{AC}$
        - calcul de la normale au fond de fissure dans le plan de la fissure : $\vec{N'}=\overrightarrow{AB}\wedge\vec{N}$
        - 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 $lst(P)=\overrightarrow{PM}\cdot\vec{N'}$
      - 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 $\alpha=MPM'$ 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.

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

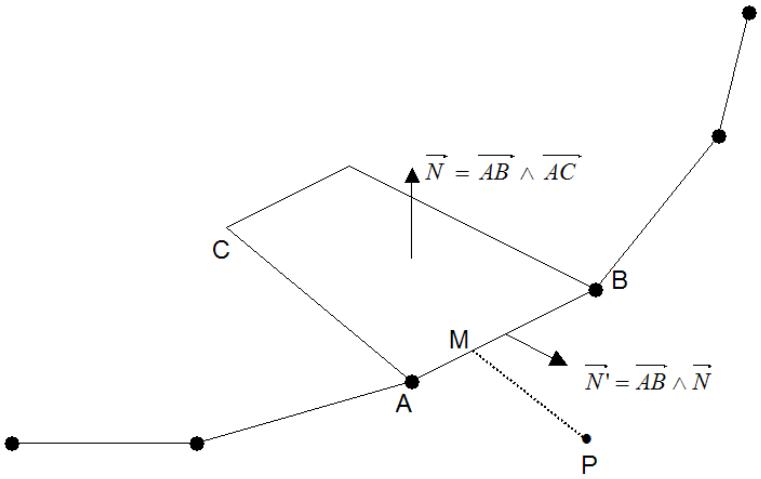


Figure 2.2.2-1 : Calcul de la level set tangente

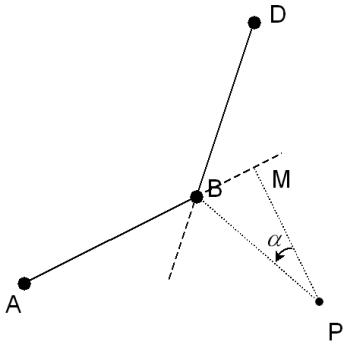


Figure 2.2.2-2 : Critère supplémentaire pour le choix du segment le plus proche

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 ε tel que $\overrightarrow{AM} = \varepsilon \times \overrightarrow{AB}$
        - si $AM < dmin$ alors $dmin = AM$ et $lst = -\varepsilon \times AB$
      - finsi
    - fin boucle
  - fin boucle
- fin boucle

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

On considère le cas pathologique suivant :

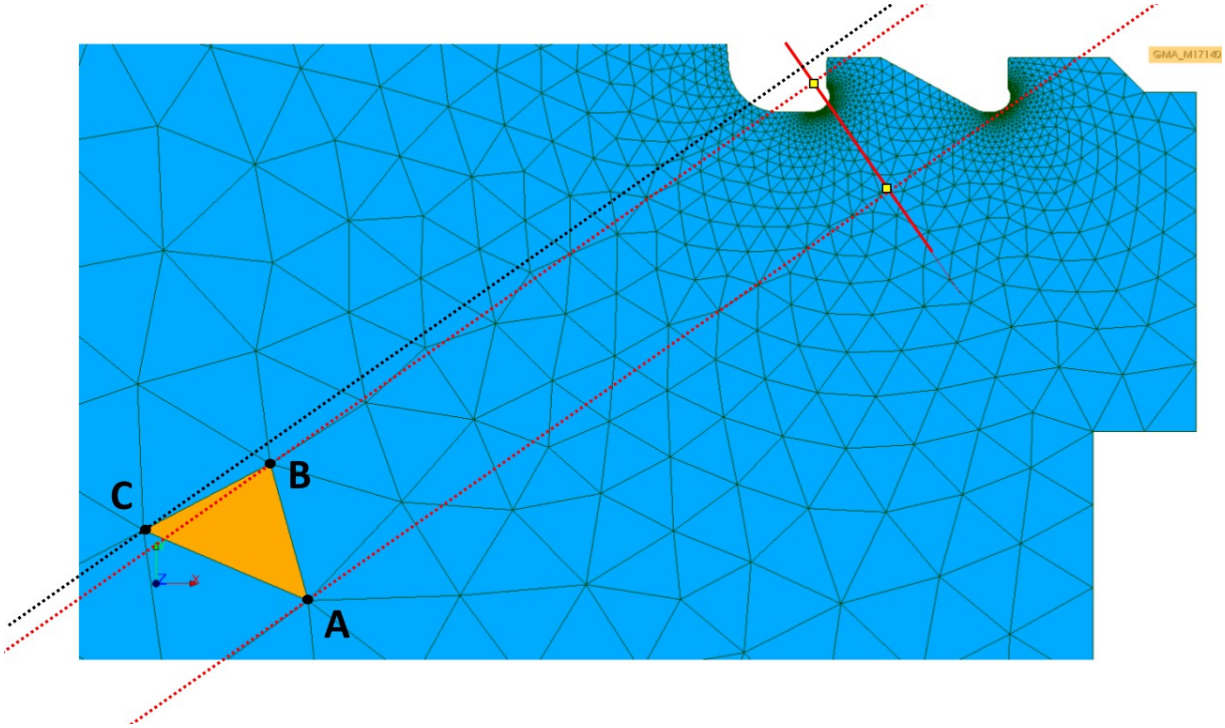


Figure 2.2.2-3: Problème d'annulation de la level-set tangente

La maille ABC voit successivement ses trois nœuds sommets se faire annuler la level-set tangente :

- à cause du critère fit-to-vertex pour les nœuds A et B à mesure que leur support est parcouru (§2.2.4)
- à cause des critères propres aux éléments quadratiques pour le nœud C.

Une fois la level-set tangente de tous les nœuds sommets annulée, on a:

$$lst_{max} = max\{lst_{ABC}\} = 0 \quad (1)$$

Donc, quand on calcule la quantité $d = lstm / lst_{max}$, on a un problème.

Cette situation peut être fréquente en 2D avec une fissure « segment » (FORM_FISS='SEGMENT') et un maillage caractérisé par un fort gradient de taille de maille (et ce d'autant plus si le maillage est triangulaire et libre).

Pour corriger, avant de calculer d , on regarde si lst_{max} est nulle.
Si oui, et si les conditions particulières sont respectées :

- tous les nœuds sommets doivent vérifier $|lst| < \varepsilon$;
- les level-sets doivent avoir été calculées depuis le catalogue de formes géométriques de DEFI_FISS_XFEM dans le cas 2D avec FORM_FISS='SEGMENT'
- pour chaque arête de la maille courante, on compare la longueur de son projeté orthogonal sur le segment (qui constitue la fissure) a la longueur de ce segment. Au moins un de ces projetés doit avoir une longueur comparable (critère relatif) à celle du segment. De cette manière on s'assure que la maille se trouve « loin » de la fissure (a moins que le maillage ne soit extrêmement grossier)

Alors, o n court-circuite la fin de l'itération courante de la boucle sur les arêtes (celle du bloc relatif aux ajustements spécifique au quadratique). Sinon, on s'arrête en erreur fatale.

2.2.3 Approximation des level sets

Quelle que 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 [bib17] :

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

$$lsn(x)=\sum_i \varphi_i(x)\,lsn_i$$
$$lst(x)=\sum_i \varphi_i(x)\,lst_i$$

où Φ_i sont les fonctions de forme linéaires classiques et lsn_i et lst_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 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 cordonnées (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 :

$$lsn^h(x,y)=-0.44347\,\Phi_1(x,y)-0.1\,\Phi_2(x,y)+0.22227\,\Phi_3(x,y)+0.040806\,\Phi_4(x,y)$$

où $\varphi_i\ i=1,4$ 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).

$$\varphi_1(x,y)=N_1(s(x,y),t(x,y))=\frac{1}{4}(1-s)(1-t)$$
$$\varphi_2(x,y)=N_2(s(x,y),t(x,y))=\frac{1}{4}(1+s)(1-t)$$
$$\varphi_3(x,y)=N_3(s(x,y),t(x,y))=\frac{1}{4}(1+s)(1+t)$$
$$\varphi_4(x,y)=N_4(s(x,y),t(x,y))=\frac{1}{4}(1-s)(1+t)$$

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 projection 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.

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

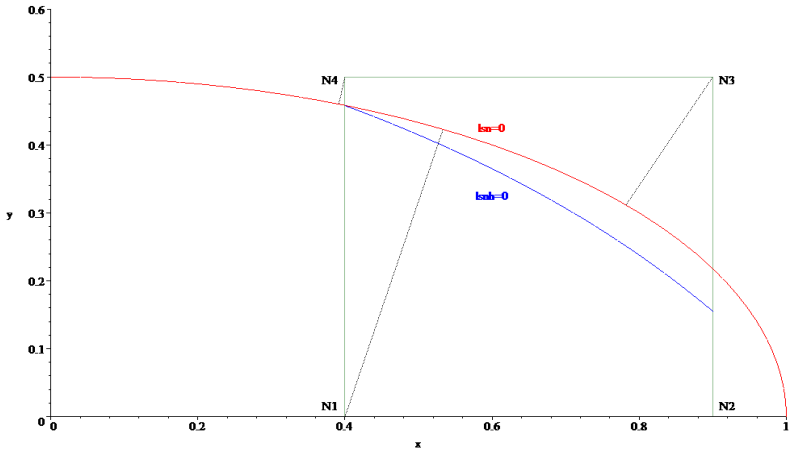


Figure 2.2.3-1 : Erreur de discrétisation de la level set

Avec une interpolation quadratique, sur un élément quadrangle QUAD8, l'approximation de la level set s'écrit :

$$lsn^h(x,y) = -0.4434712\,\varphi_1(x,y) - 0.1\,\varphi_2(x,y) + 0.2222696\,\varphi_3(x,y) + 0.0408060\,\varphi_4(x,y) - 0.3304038\,\varphi_5(x,y) + 0.0228051\,\varphi_6(x,y) + 0.1112398\,\varphi_7(x,y) - 0.2027748\,\varphi_8(x,y)$$

où $\varphi_i \; i=1,8$ sont les fonctions de forme associées aux nœuds du quadrangle. Ces fonctions s'expriment 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).

$$\varphi_1(x,y) = N_1(s(x,y),t(x,y)) = \frac{1}{4}(1-s)(1-t)(-1-s-t)$$
$$\varphi_2(x,y) = N_2(s(x,y),t(x,y)) = \frac{1}{4}(1+s)(1-t)(-1+s-t)$$
$$\varphi_3(x,y) = N_3(s(x,y),t(x,y)) = \frac{1}{4}(1+s)(1+t)(-1+s+t)$$
$$\varphi_4(x,y) = N_4(s(x,y),t(x,y)) = \frac{1}{4}(1-s)(1+t)(-1-s+t)$$
$$\varphi_5(x,y) = N_4(s(x,y),t(x,y)) = \frac{1}{2}(1-s^2)(1-t)$$
$$\varphi_6(x,y) = N_4(s(x,y),t(x,y)) = \frac{1}{2}(1+s)(1-t^2)$$
$$\varphi_7(x,y) = N_4(s(x,y),t(x,y)) = \frac{1}{2}(1-s^2)(1+t)$$
$$\varphi_8(x,y) = N_4(s(x,y),t(x,y)) = \frac{1}{2}(1-s)(1-t^2)$$

avec les changements de variables suivants :

$$s(x,y) = 4x - 2.6$$
$$t(x,y) = 4y - 1$$

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

On compare alors l'erreur de discrétisation entre l'interpolation linéaire et l'interpolation quadratique Figure 2.2.3-2: **à raffinement égal, les éléments quadratiques sont plus adaptés pour décrire une iso-zéro de forme courbe**.

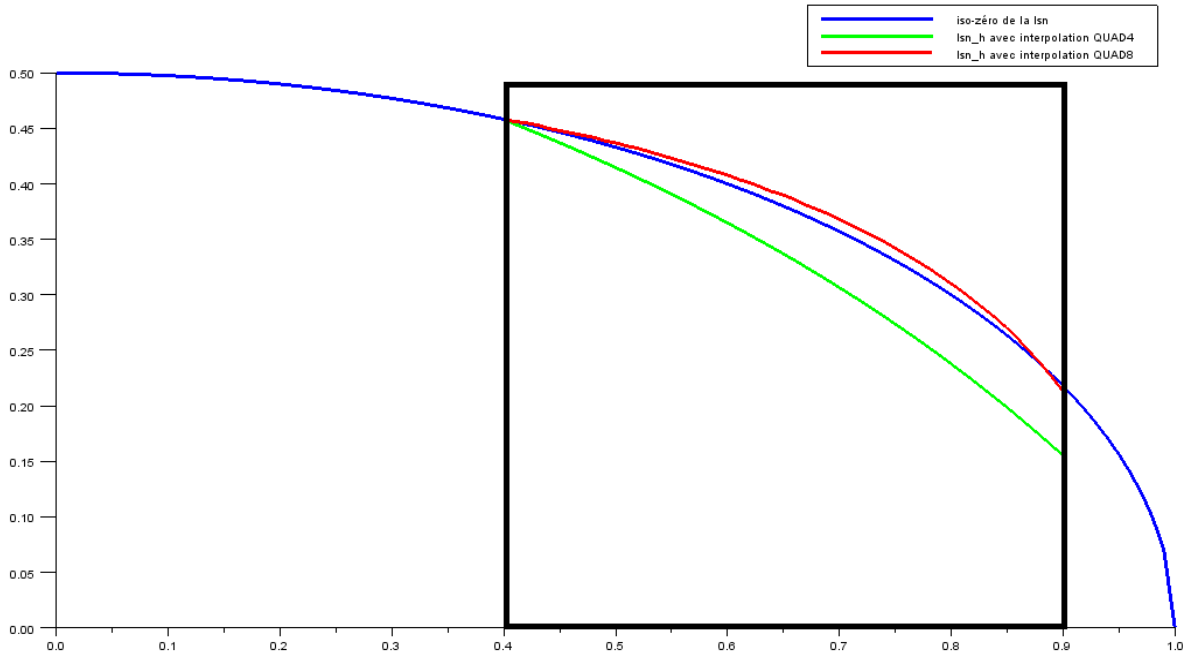


Figure 2.2.3-2: Erreur d'interpolation l'iso-zero: quadratique vs linéaire

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, lsn s'annule trop « près » d'un nœud extrémité, la valeur de lsn 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 [bib20]).
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 $lsn(A) \neq lsn(B)$

    $$d = \frac{lsn(A)}{lsn(A) - lsn(B)}$$

    si $|d| \leq 0.01$ alors $lsn(A) = 0$ fin si

    si $|d-1| \leq 0.01$ alors $lsn(B) = 0$ fin si

    fin si

    si élément quadratique

    si $lsn(A) = 0$ et $lsn(B) = 0$ alors

    $d = lsn(M)$

    si $|d| \leq 0.0001$ alors $lsn(M) = 0$ fin si

    fin si

    fin si
  - fin boucle
- fin boucle

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

Dans le cas d'une interpolation quadratique, un réajustement complémentaire est nécessaire afin de limiter et contrôler les situations d'annulation multiple de la level set le long d'une arête. En effet, pour ne pas multiplier les configurations de découpe des éléments enrichis en sous éléments d'intégration, on autorise seulement deux situations d'annulation multiple le long de l'arête d'un élément enrichi. La première correspond au cas où la level set est nulle tout au long de l'arête (donc nulle en ses deux nœuds extrémités et en son nœud milieu). La deuxième est la situation où la level set est nulle sur l'un des nœuds extrémité et le nœud milieu (voir Figure 2.2.4-1) . Dans toutes les autres situations, la level set ne doit s'annuler qu'une fois le long d'une arête. Il sera alors aisé de détecter les annulations de la level set le long d'une arête pour former les sous éléments d'intégration. Pour se ramener dans ces situations, un réajustement est effectué lorsque l'on se trouve dans l'une des situations représentées sur la Figure 2.2.4-2



Figure 2.2.4-1: Annulation multiple de la level set le long d'une arête

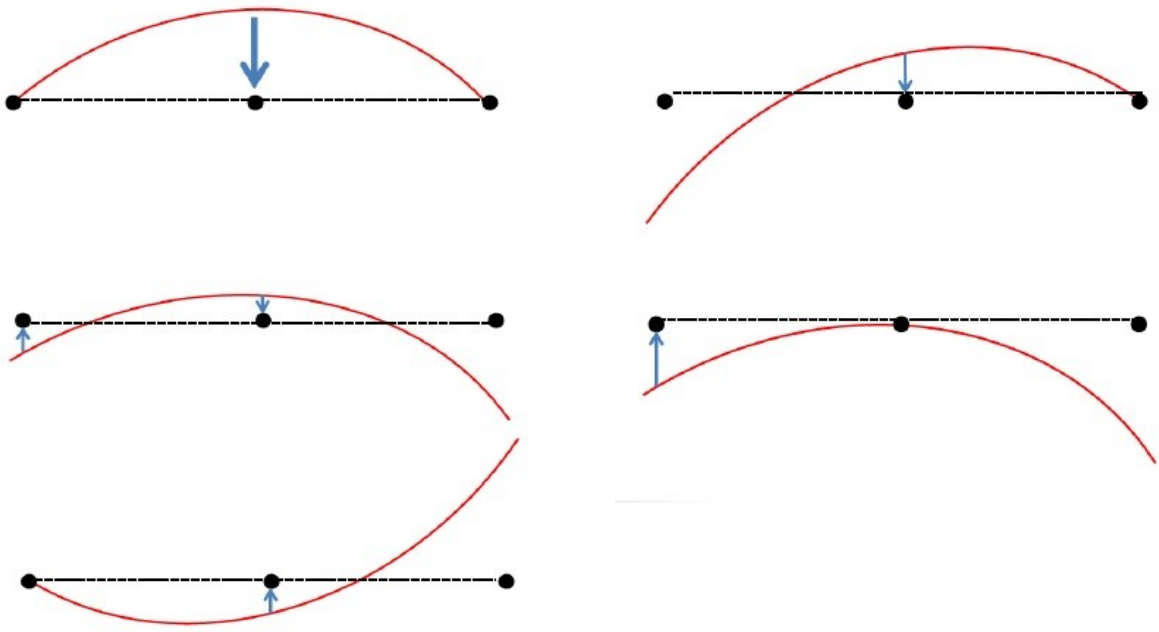


Figure 2.2.4-2: Réajustement complémentaire des level set

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

Dans les deux configurations de la Figure 2.2.4-2 où l'on ajuste la level set de l'un des nœuds extrémités à zéro, c'est l'extrémité de l'arête qui a la plus petite level set en valeur absolue qui est choisie pour être ajustée. Aussi, lorsque l'ajustement à effectuer est jugé trop important, ce qui révèle une courbure importante de la level set par rapport à la taille de l'élément, un message d'alarme préconisant d'utiliser un maillage plus fin est émis. Le critère choisi est le suivant : le message d'alarme est émis dès lors que :

$$|lsn\,à\,ajuster| > 0.01 * \max_{noeuds\,de\,l'élément} |lsn|$$

Cette procédure est appliquée aussi bien pour la level set tangente que pour la level set normale.

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 [bib21].

Remarque :
Si la forme de la level-set représentée, n'est pas assez régulière (par exemple, possède un angle vif ou une bifurcation), la définition de la distance par projection orthogonale sur l'iso-zéro, peut-être ambiguë Figure 2.2.5-1.

Au voisinage de la singularité, la projection sur la surface de la level-set n'est pas possible: le gradient l'iso-zéro de la level-set est discontinu, donc la normale à l'iso-zéro de la level-set n'existe pas. On désigne alors par « zone d'ombre », la région où le calcul de distance par projection orthogonale sur l'iso-zéro, est à proscrire.

Dans la « zone d'ombre », on calcule la distance en revenant à sa définition fondamentale : la distance d'un point M à l'iso-zéro est la longueur minimale reliant le point M à un point de l'iso-zéro.
Soit C le point correspondant au sommet de l'angle droit sur l'iso-zéro.
Pour tout point M dans la « zone d'ombre » Figure 2.2.5-1, C vérifie la longueur minimale , c'est-à-dire : $d(M, \text{iso_zero}) = min(\|\overrightarrow{MN}\|, N \in \{\text{iso_zero}\}) = \|\overrightarrow{MC}\|$.

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

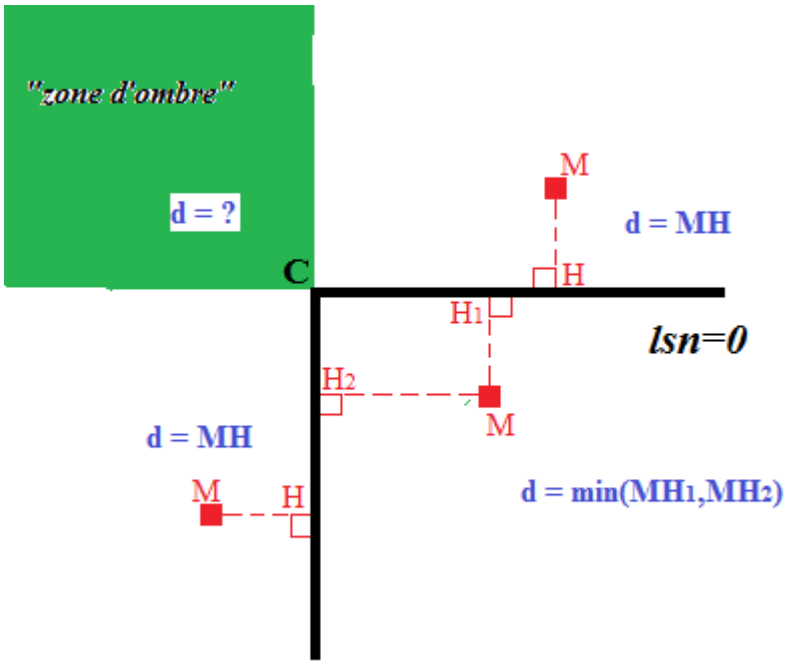


Figure 2.2.5-1 : difficulté de la définition d'une distance par projection orthogonale en cas de singularité de l'iso-zéro

Par ailleurs, notons que l'interpolation de level-set régularise la singularité. Comme la level-set non régulière est interpolée par des fonctions polynomiales §2.2.3, la level-set discrétisée devient une fonction régulière au sein de l'élément.

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 créé 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).

- •Ceux lies au sous-découpage (voir § 3.3),

Les champs par éléments « concaténés » créés sont :
- •Ceux lies 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 [bib22].

2.3 Base locale au fond de fissure

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

Les gradients des level sets peuvent être utilisés pour définir la base locale au fond de fissure [bib17] [bib16].

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 $\left(e_1, e_2, e_3\right)$ se calcule alors en tout point grâce aux champs de gradients nodaux :

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

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

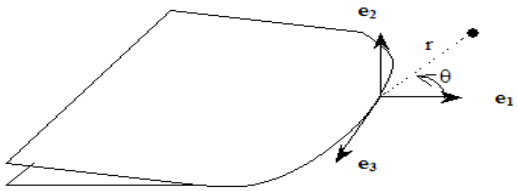


Figure 2.3-1 : Base locale au fond de fissure

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]).

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

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

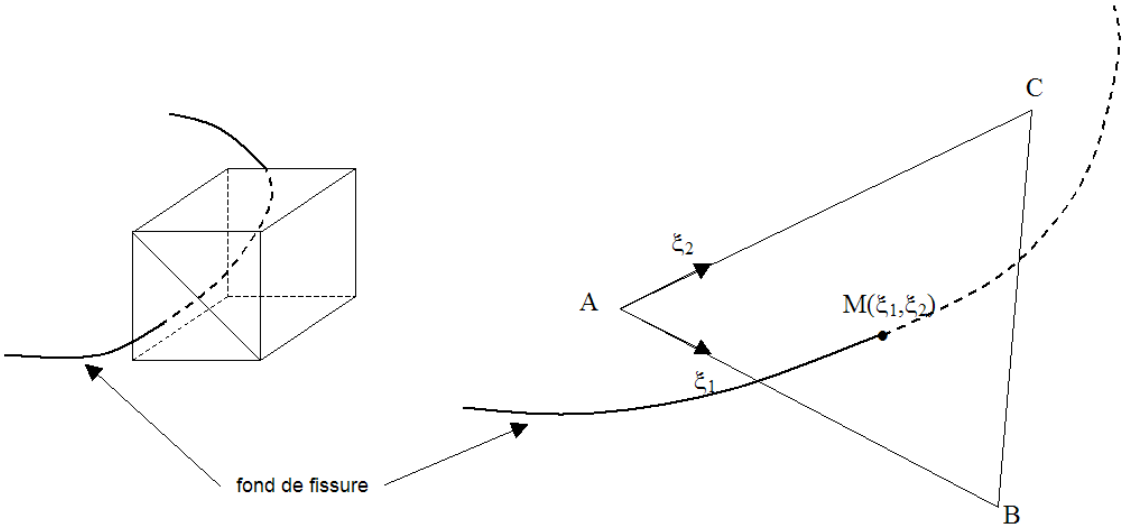


Figure 2.4.1-1 : Face d'un élément intersectée par le fond de fissure

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 $\left(A, \overrightarrow{AB}, \overrightarrow{AC}\right)$

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

Le système s'écrit alors

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

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

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 $N^M = N^P$,

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

- si M est situé sur une arête caractérisée par les numéros de sommets N_1^M et N_2^M et que

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

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

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

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 $Dist(1) < 10^{-4} . Dist(2)$ 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 à 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 [§2.4.1], 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 :

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

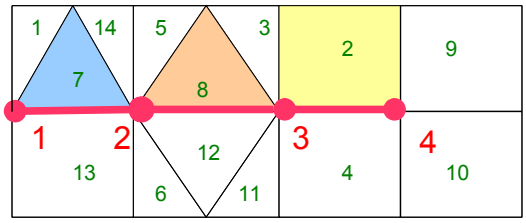


Figure 2.4.2.1-1 : Vue de dessus d'un maillage d'une structure parallélépipédique

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 hexaèdre* `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	$(1,0)$	$(4,3)$	$(3,0)$	$(3,4)$	$(2,0)$	$(2,0)$	$(2,1)$	$(3,2)$	...

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

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 a 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 : Ordonnancement des points du fond

Cette étape correspond à l'orientation du fond à proprement parler.
Pour commencer l'ordonnancement 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.

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

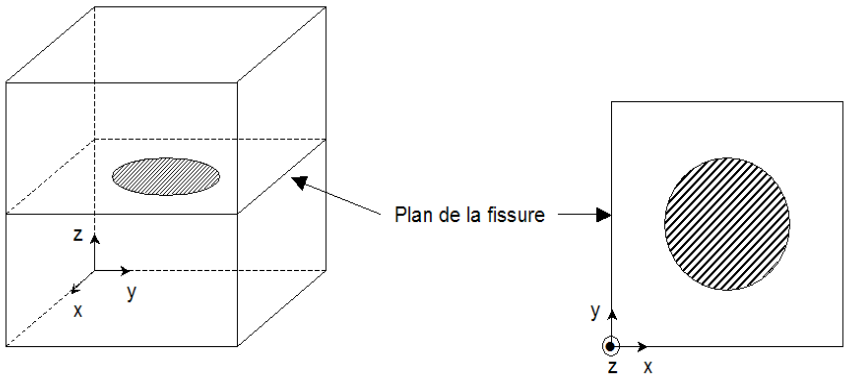


Figure 2.4.2.2-1 : Cas d'un fond de fissure continu

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).

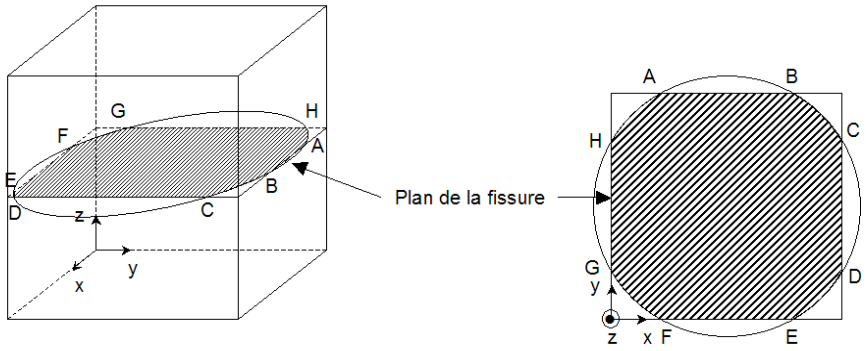


Figure 2.4.2.2-2 : Cas d'un fond de fissure multiple

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.

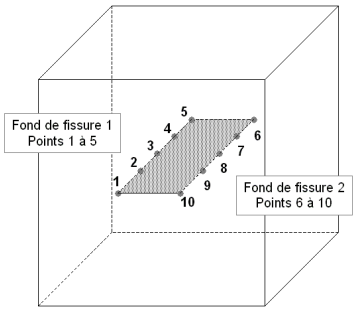


Figure 2.4.2.2-3 : Fond de fissure multiple

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.

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

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) :

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

- 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 débuter le nouveau fond de fissure et on passe à la maille suivante $IMA+1$

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

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 $\Omega \in \Re^3$ 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.

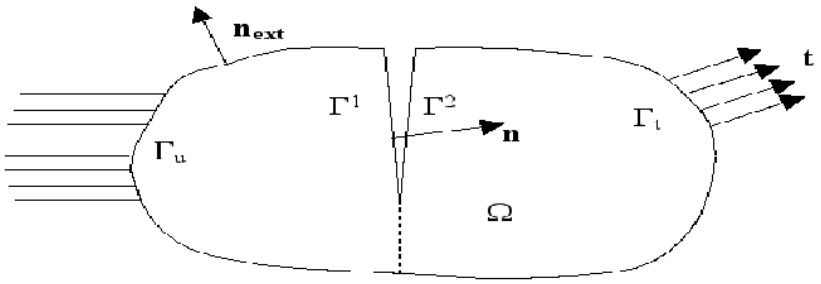


Figure 3.1-1 : Notations du problème général

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

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

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

$$\varepsilon = \varepsilon(u) = \nabla_s u \qquad \textbf{éq 3.1-2}$$

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

On considère un matériau élastique linéaire[1]. La loi de comportement du solide Ω s'écrit :

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

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.

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

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

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 [bib23]. 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 :

$$u^h(x)=\sum_{i\in N_n(x)} a_i\varphi_i(x)+\sum_{j\in N_n(x)\cap K} b_j\varphi_j(x)H_j(lsn(x))+\sum_{k\in N_n(x)\cap L}\sum_{\alpha=1}^{4} c_k^\alpha\varphi_k(x)F^\alpha(lsn(x),lst(x))$$

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

3.2.1 Enrichissement avec une fonction de sélection de domaine (2ème terme)

Supposons que l'interface Γ_c partitionne le domaine tel que $\Omega=\Omega_+\cup\Omega_-$. Si Γ_c est une fissure, on partitionne de la même manière le domaine Ω en étendant virtuellement Γ_c .

Afin de représenter le saut de déplacement à travers Γ_c , on introduit la fonction de sélection de domaine ou fonction caractéristique de domaine $H_j(x)$ [bib76] définie par :

$$Si\quad x_j\in\Omega_+,\quad H_j(x)=\begin{cases}0 & \text{si } x\in\Omega_+\\ -2 & \text{si } x\notin\Omega_+\end{cases}$$

$$Si\quad x_j\in\Omega_-,\quad H_j(x)=\begin{cases}0 & \text{si } x\in\Omega_-\\ +2 & \text{si } x\notin\Omega_-\end{cases}$$

En se servant de la level set normale, la quantité $H_j(lsn(x))$ vaut 0 si le point x et le nœud x_j se trouvent du même coté de la fissure et ± 2 sinon. Le coefficient « 2 » est introduit pour avoir une écriture plus simple du saut moyen de déplacement le long de l'interface.

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).

Remarque : par abus de langage, on appellera aussi les fonctions de sélection de domaine, fonctions « Heaviside ». Mais il faudra se référer à la définition ci-dessus, pour représenter l'approximation de la discontinuité du champ de déplacement.

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

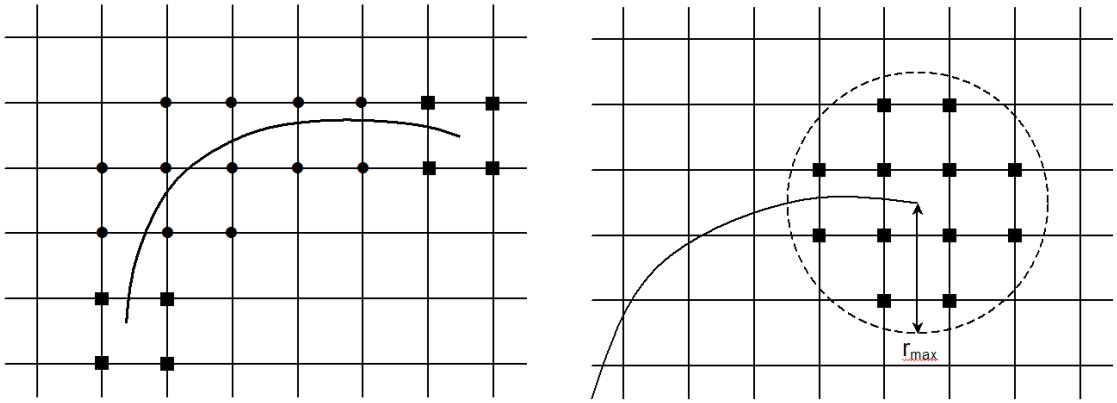


Figure 3.2.1-1 : À gauche, les nœuds « ronds » sont enrichis par la fonction Heaviside et les nœuds « carrés » par les fonctions singulières (enrichissement topologique). À droite, les nœuds « carrés » sont enrichis par les fonctions singulières (enrichissement géométrique).

3.2.2 Enrichissement avec une fonction de sélection de domaine pour un branchement de fissures

Supposons un branchement de deux discontinuités, qui partitionne l'espace en trois domaines tel que $\Omega=\Omega_1\cup\Omega_2\cup\Omega_3$. S'il s'agit d'un branchement de deux fissures, on partitionne de la même manière le domaine Ω en étendant virtuellement les trois branches de fissures.

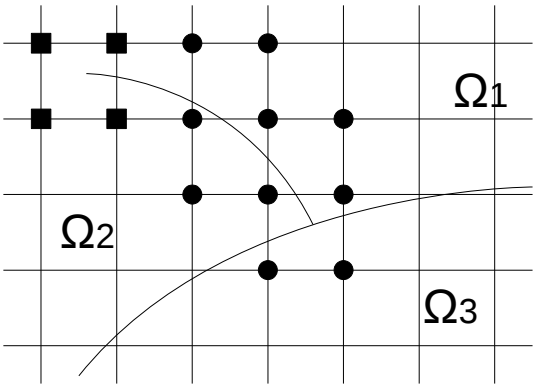


Figure 3.2.2-1 partitionnement du domaine par une jonction de fissure

Pour un branchement simple, la généralisation des fonctions de sélection de domaines conduit à l'écriture suivante:

$$u^h(x)=\sum_{i\in N_n(x)} a_i\varphi_i(x)+\sum_{j\in K\cap\Omega_1} b_{j,1}\varphi_j(x)\chi_{\Omega_2}+b_{j,2}\varphi_j(x)\chi_{\Omega_3}$$
$$+\sum_{j\in K\cap\Omega_2} b_{j,1}\varphi_j(x)\chi_{\Omega_1}+b_{j,2}\varphi_j(x)\chi_{\Omega_3}$$
$$+\sum_{j\in K\cap\Omega_3} b_{j,1}\varphi_j(x)\chi_{\Omega_1}+b_{j,2}\varphi_j(x)\chi_{\Omega_2}$$

Où χ_{Ω_i} sont des fonction de sélection de domaines définies par :

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

$$\chi_{\Omega_i}(x)=\begin{cases} 2 & \text{si } x\in\Omega_i \\ 0 & \text{si } x\notin\Omega_i \end{cases}$$

Toutefois, la description topologique des trois domaines connexes (Ω_1, Ω_2 et Ω_3) n'est pas triviale, compte-tenu de la seule information fournie par les level-sets. Les levet-sets permettent de représenter un changement de signe scalaire à travers une discontinuité. Alors que nous souhaitons représenter le partitionnement de l'espace au voisinage de la discontinuité.

D'un point de vue "élémentaire", nous constatons que le champ de signe de level-sets lu sous forme vectorielle, colle approximativement au partitionnement que nous voulons réaliser. Ce partitionnement, grâce à la vectorisation des level-sets scalaires, permet alors de réutiliser avantageusement les structures de données du Code_Aster. Toutefois, le seul partitionnement élémentaire est insuffisant, pour représenter le "ddl domaine" ($\chi_\Omega\phi i$).

Remarque :
En fait, le "ddl domaine" correspond à la collection des partitions du domaine Ω sur l'ensemble du support du nœud i. Sur le support de chaque nœud, il est donc indispensable de concaténer les partitions élémentaires correspondant à un "ddl domaine" donné. Cette opération n'est pas triviale puisque les champs de signe évoluent d'un élément à l'autre (comme les champs de signes scalaires ne sont pas prolongés sur tous les éléments, pour localiser sur quelques éléments d'intérêt la définition d'une fissure, appelés aussi "bande proche" d'enrichissement). Un algorithme dédié à la concaténation du partitionnement élémentaire, a été développé pour résoudre cette difficulté fondamentale.

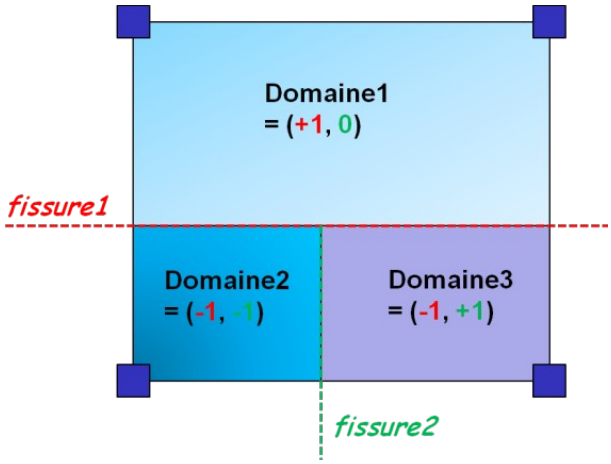


Figure 3.2.2-2 : partitionnement d'un élément multi-fissuré à l'aide du champ de signe "vectorisé". Pour chaque domaine, la première composante scalaire du vecteur signe, correspond à la caractérisation de la discontinuité de la fissure n°1 et la deuxième composante scalaire du vecteur signe, correspond à la discontinuité de la fissure n°2.

Construction des fonctions de signe ou jonctions :

Rappelons la construction des fonctions de signe Heaviside ou fonctions de type jonction [bib70]. Il s'agit d'une fonction Heaviside qui est « tronquée » au niveau du branchement. Les nœuds enrichis sont représentés sur la Figure 3.2.2-3.

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

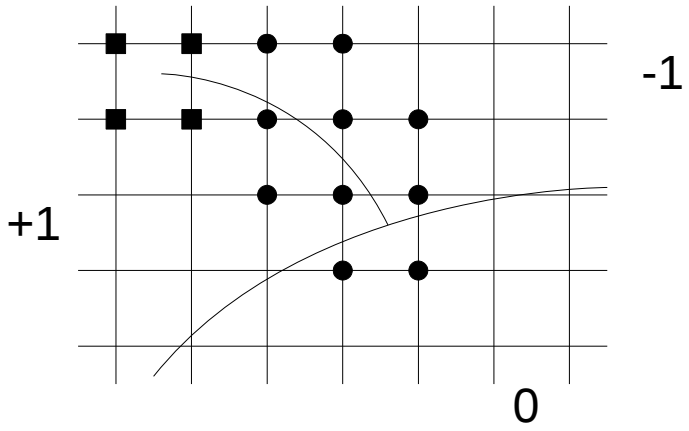


Figure 3.2.2-3 : Enrichissement pour la deuxième fissure. Les nœuds ronds sont enrichis par la fonction jonction, qui vaut +1, -1, 0.

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 :

$$J_N(x)=\begin{cases} H(lsn_N(x)) & \text{si } \forall i\in P,\ sign(lsn_i(fiss_n))\,H(lsn_i(x))\geq 0 \\ 0 & \text{sinon} \end{cases}$$

Un exemple de configuration est présenté Figure 3.2.2-4. 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 $J_5(x)=H(lsn_5(x))$ sur le domaine hachuré de la figure, et zéro ailleurs.

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

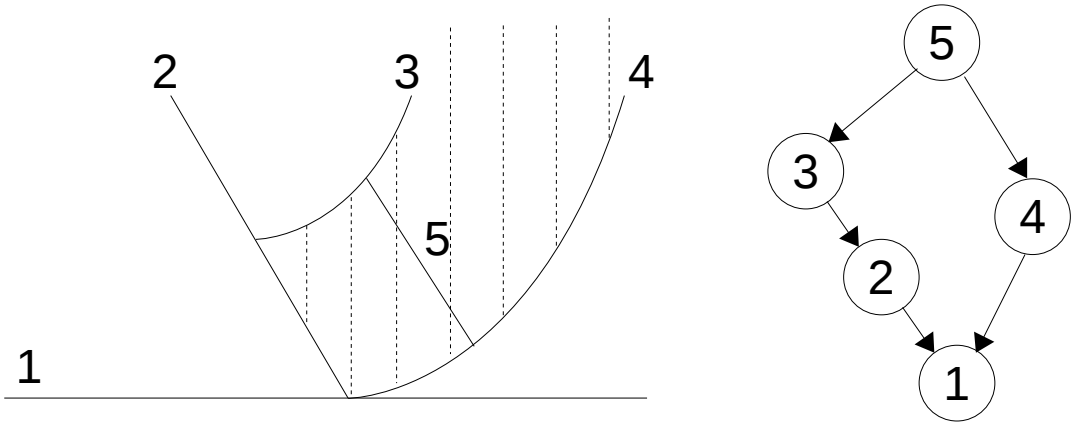
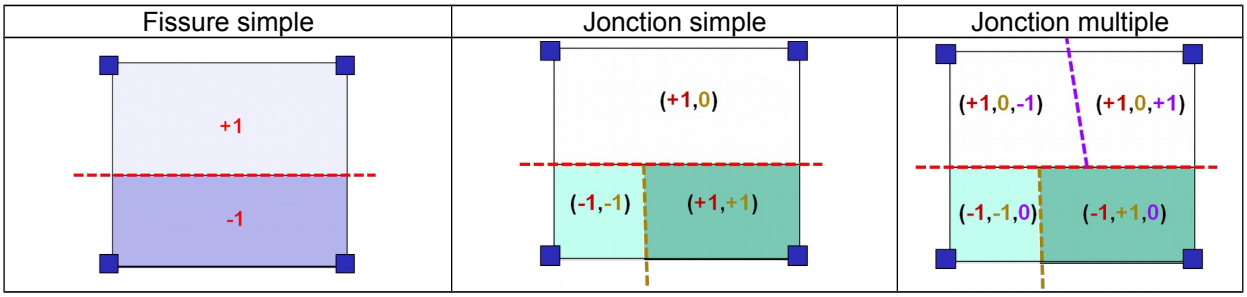


Figure 3.2.2-4 : Réseau de fissures à gauche, arbre de hiérarchie à droite, la zone hachurée correspond au domaine où la fonction d'enrichissement de la fissure 5 n'est pas nulle.

Exploitation des fonctions de signe pour l'assemblage des ddls domaines:

Pour construire les ddls domaines définis au paragraphe précédent, nous proposons de réutiliser l'information du champ de signes Heaviside pour définir un partitionnement élémentaire. L'algorithme étant beaucoup trop complexe à expliquer en termes de structures de données, nous donnons une traduction graphique de l'algorithme. Voici par exemple les champs de signes Heaviside concaténés qu'on souhaite exploiter pour construire le partitionnement élémentaire :



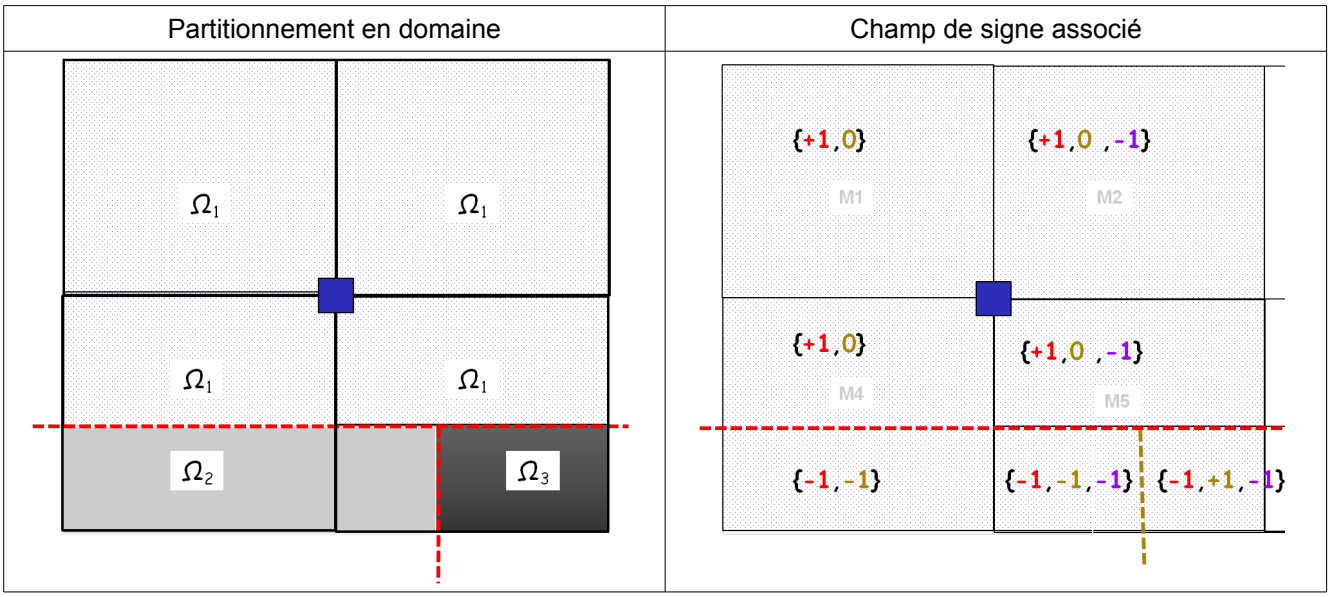
Le cas d'une mono-fissure étant assez facile, nous allons nous attarder en détail sur cas d'une jonction simple. Dans l'explication graphique suivante, nous proposons de changer de perspective par rapport à la description élémentaire suggérée ci-dessus. L'algorithme sera décrit du point de vue du support du nœud, qui est plus adapté pour décrire les fonctions d'enrichissement nodales.

Du point de vue du nœud, l'information sur les champs de signes est plus riche que le cas idyllique décrit ci-dessus. En effet, le nœud voit l'information sur les fissures proches, c'est -à-dire, situées dans la première et la deuxième bande d'éléments attenants au support du nœud. Dans le support du nœud, il existe donc un statut de fissures (cf. [D4.10.02]) pour discriminer les fissures intersectant le support du nœud et les fissures proches n'intersectant pas le support du nœud.

Cette difficulté rend non triviale l'association entre les champs de signes et les partitions de domaines. Cette association est nécessairement locale et doit prendre en compte la pollution des champs de signe liée aux bandes proches citées précédemment. Sur l'exemple donné dans le tableau ci-dessous, il faut associer les domaines au champ de signe de la manière suivante :

- on associe le domaine Ω_1 $\leftrightarrow$ au champ de signe $\{+1\,,0\,,*\}$
- on associe le domaine Ω_2 $\leftrightarrow$ au champ de signe $\{-1\,,-1\,,*\}$
- on associe le domaine Ω_3 $\leftrightarrow$ au champ de signe $\{-1\,,+1\,,*\}$

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



Pour compresser l'information des vecteurs signes, on préfère transformer les vecteurs signes à l'aide de la fonction de codage réversible suivante :

$$code(\underline{P}) = \sum_{ifiss=1}^{nfiss} 3^{nfiss-ifiss}\left(He(\underline{P}, ifiss)+1\right)$$

où,

- **ifiss** est le numéro dans le vecteur champ de signe,
- **P** désigne le point courant (un nœud ou un point de gauss),
- **nfiss** est la longueur du champ de signe.

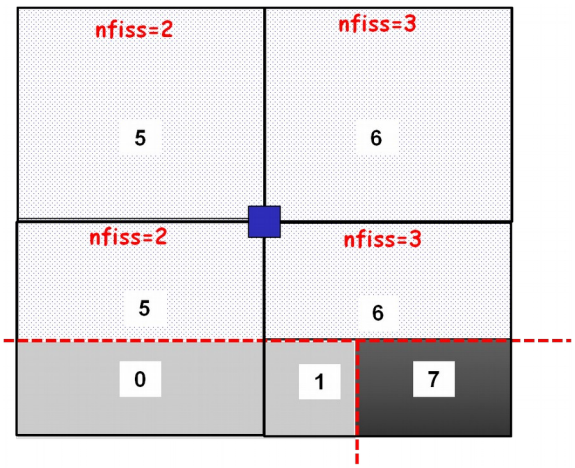


Figure 3.2.2-5 : les vecteurs signes ont des longueurs variables, ce qui conduit à des identifiants différents pour un même domaine. Ces identifiants constituent alors un deuxième niveau de sous-partitionnement.

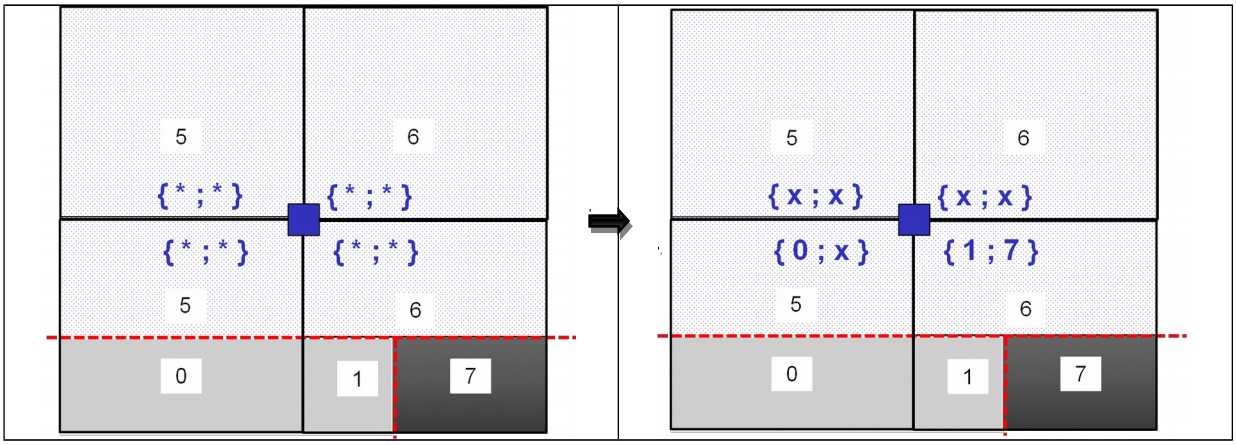
Compte-tenu de la variabilité de la longueur *nfiss* des vecteurs signes, un domaine peut avoir un code différent d'un élément à un autre, comme illustré Figure 3.2.2-5. Par exemple, le domaine Ω_1 possède deux identifiants (5 et 6). Ces identifiants peuvent être vus comme deux sous-partitions du domaine Ω_1 .

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

Ce sous-partitionnement n'est pas gênant tant que l'information sur ce sous-partitionnement est regroupée au niveau du nœud, pour l'assemblage des « ddls » domaine. Pour identifier ces deux ddls domaines dans chaque élément, il suffit alors de concaténer l'information sur les sous-partitions :

- Si une sous-partition du domaine complémentaire est trouvée, l'information est stockée dans l'emplacement dédié (structure de données par nœud et par élément),
- Sinon, il n'y a pas de sous-partition pour cet élément.

Le résultat d'une telle boucle élémentaire est résumé dans le tableau ci-dessous. Dans chaque élément on recherche une sous-partition de chaque domaine complémentaire au nœud bleu. Si aucune sous-partition n'est trouvée dans l'élément, on marque dans l'emplacement dédié une croix 'x' (ou -1 par exemple).



Pour le nœud en bleu appartenant au domaine Ω_1 , l'évaluation des fonctions caractéristiques χ_{Ω_2} ou χ_{Ω_3} est évidente compte-tenu de l'information sur le partitionnement du tableau ci-dessus. Dans chaque élément, la première composante (du champ concaténé) renseigne sur l'identifiant du premier ddl domaine χ_{Ω_2} , la deuxième composante (du champ concaténé) renseigne sur l'identifiant du deuxième ddl domaine χ_{Ω_3} :

- Si l'identifiant du domaine auquel appartient le point de Gauss correspond à la composante $k \in [1,2]$ stockée au nœud, alors la fonction caractéristique associée à la composante k prend la valeur « +2 ». Précisons à nouveau que la composante $k=1$ est associée au ddl domaine χ_{Ω_2} et la composante $k=2$ est associée au ddl domaine χ_{Ω_3} . Il est important de souligner que cette position ne change pas d'un élément à un autre, pour assurer la contiguïté du partitionnement. Par exemple, soit la collection des identifiants {0,1} correspondante au domaine Ω_2 . C'est exactement l'information qui est stockée en première position dans la structure de donnée au nœud « bleu », dans les éléments intersectant le domaine Ω_2 .
- Sinon, la fonction caractéristique prend la valeur zéro si l'identifiant au point de gauss ne correspond pas à la composante au nœud dans l'élément considéré.

Remarques:

1. *On peut remarquer que le ddl domaine χ_{Ω_1} n'est pas traiter dans l'exemple ci-dessus. En effet, pour représenter les deux discontinuités introduites par le branchement de deux fissures il est nécessaire d'enrichir avec deux fonctions discontinues les nœuds dont le support est intersecté par la double discontinuité. On choisit alors d'enrichir avec les ddls associés aux domaines complémentaires au domaine auquel appartient le nœud. Ce choix est bien adapté, compte-tenu des problèmes de conditionnement détaillés au §3.2.6.*
2. *Dans Code_Aster, on préfère stocker l'information du signe Heaviside et celle de l'identifiant de domaine au point de Gauss dans les sous-éléments d'intégration.*

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

3.2.3 Enrichissement avec les fonctions singulières (3ème 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 [bib25]. Ces expressions ont été déterminées pour une fissure plane en milieu infini.

$$
\begin{aligned}
u_1 &= \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\left(K_1\cos\frac{\theta}{2}(\kappa-\cos\theta)+K_2\sin\frac{\theta}{2}(\kappa+2+\cos\theta)\right)\\
u_2 &= \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\left(K_1\sin\frac{\theta}{2}(\kappa-\cos\theta)+K_2\cos\frac{\theta}{2}(\kappa-2+\cos\theta)\right)\\
u_3 &= \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}K_3\sin\frac{\theta}{2}\\
\mu &= \frac{E}{2(1+\nu)} \quad \text{et} \quad \kappa = 3-4\nu \quad \text{en déformations planes}.
\end{aligned}
\qquad \textbf{éq. 3.2.3-1}
$$

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 suivante[2] :

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

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 $\mathrm{atan2}(lsn,lst)$ 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 lèvre inférieure $(lsn=0)$, la fonction $\mathrm{atan2}(0,lst)$ donne un angle valant π . En théorie, $\mathrm{atan2}(-0,lst)$ 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 :

$$\theta=H(lsn)\left|\mathrm{atan}\,2(lsn,lst)\right|,\quad \theta\in[-\pi,\pi]$$

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.

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

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

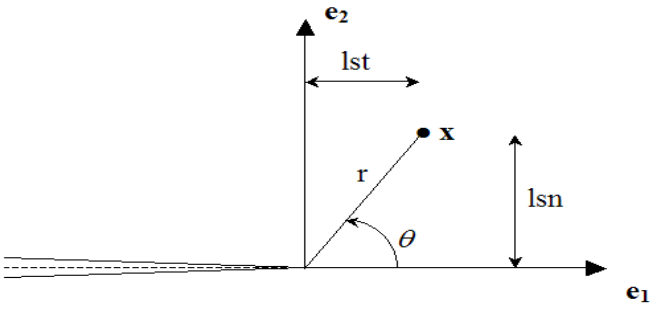
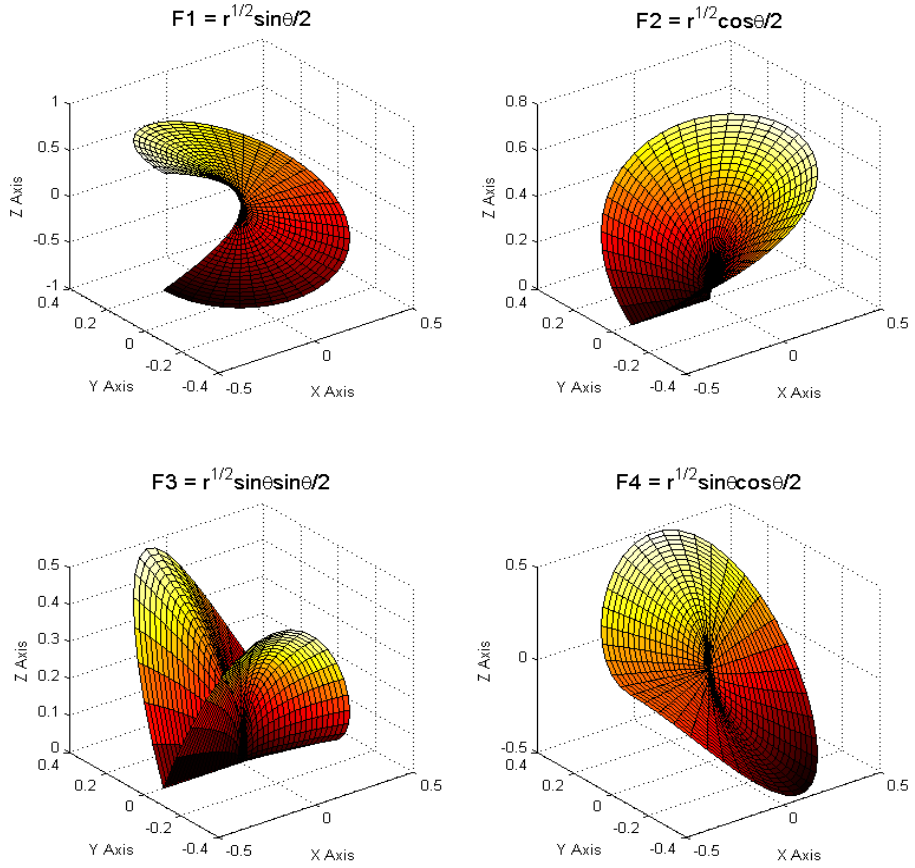


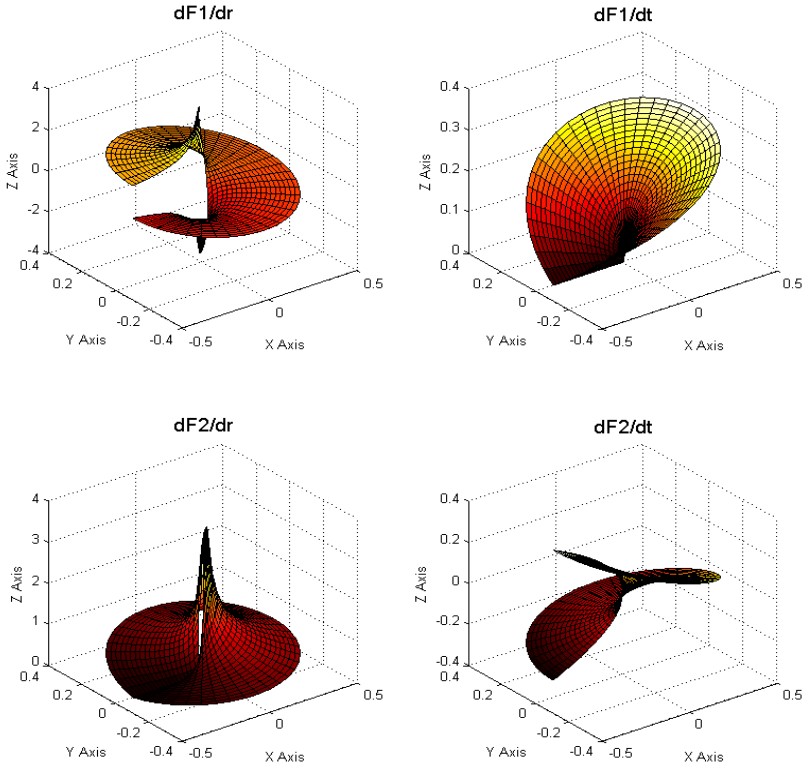
Figure 3.2.3-1 : Coordonnées polaires dans la base locale

On note que seule la 1ère 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 [bib16] [bib18], au moins pour les fissures dont le fond est assez régulier. Ces fonctions sont dites « singulières » car leurs dérivées sont singulières en $r=0$.



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

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



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

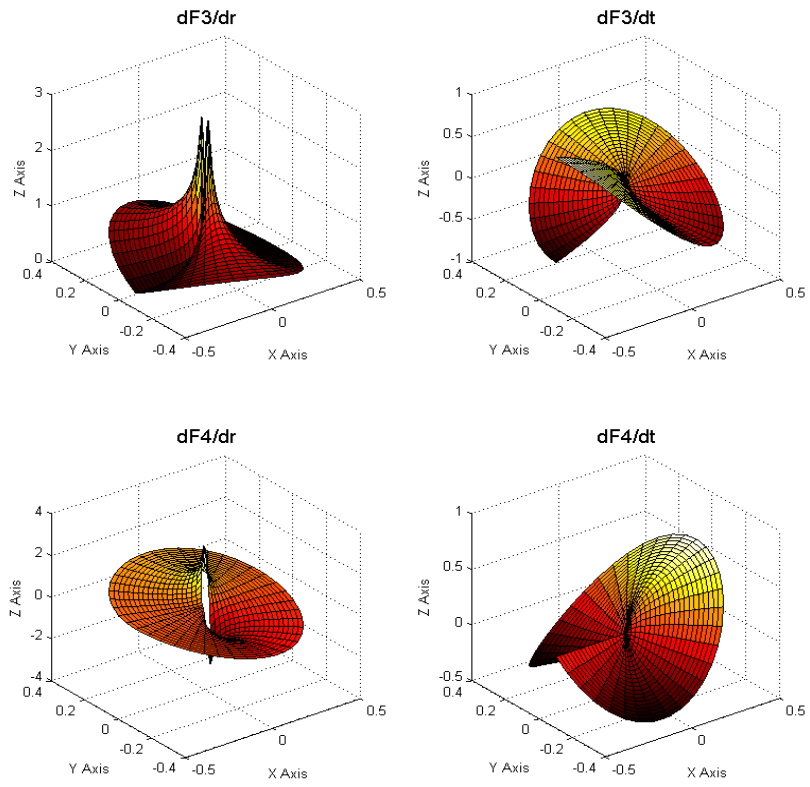


Figure 3.2.3-3 : Dérivées des fonctions d'enrichissement

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 [bib24], 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) :

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

Les premières études de convergence ont été effectuées en 2000 dans le cadre de la GFEM [bib26], 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 :

$$e_{rel}=\|u-u_h\|_{H^1}\underset{h\to 0}{\approx} Ch^{\alpha}$$

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.

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

Stazi *et al.* [bib27] 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.* [bib28] confirme cette observation et montre qu'une zone d'enrichissement fixe permet de retrouver un taux de convergence presque optimal.

Parallèlement, Laborde *et al.* [bib29] 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 3ème 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 Pk). 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. [bib29] comme beaucoup d'autres utilise des fonctions de forme de degré k pour les termes classiques et enrichis par les fonctions de discontinuité (2eme terme), de manière à ce que le saut soit de degré k . Par contre, pour les termes enrichis par les fonctions singulières (3eme terme), 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*

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

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.

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

Des idées similaires apparaissent dans [bib17], mais il semblerait que certains cas de figures n'aient pas été pris en compte. Ces expressions sont l'aboutissement des premiers efforts [bib30] 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 :

$$k\in L\Leftrightarrow\sqrt{lsn(x)^2+lst(x)^2}\leq r_{\max}$$

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 [bib31]. 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

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

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

• dans le cas d'une interpolation linéaire

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

• dans le cas d'une interpolation quadratique
On détermine la coordonnée dans l'espace de référence ξ du point d'intersection C avec la courbe $lsn=0$ sur l'arête. Sa lst est alors donnée par :

$$lst(C)=\frac{lst(A)*\xi*(\xi-1)}{2}+\frac{lst(B)*(\xi+1)*\xi}{2}-lst(M)*(\xi-1)*(\xi+1)$$

éq. 3.2.5.1-2

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 $(\mathrm{minlsn}.\mathrm{maxlsn}<0)$ et $(\mathrm{maxlst}\leq 0)$ alors $P\in K$

cas d'enrichissement topologique
si $(\mathrm{minlsn}.\mathrm{maxlsn}\leq 0)$ et $(\mathrm{minlst}.\mathrm{maxlst}\leq 0)$ 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. 3.2.5.1-1], 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 :

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

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

Pour obtenir, dans le cas quadratique, l'équation [éq. 3.2.5.1-2], ainsi que la valeur de la level set tangente au point C , on détermine au préalable la coordonnée dans l'espace de référence ξ du point C
La lsn est interpolée quadratiquement le long de l'arête. Il s'agit donc de résoudre un polynôme du second degré :

$$0=\frac{lsn(A)*\xi*(\xi-1)}{2}+\frac{lsn(B)*(\xi+1)*\xi}{2}-lsn(M)*(\xi-1)*(\xi+1)$$

L'équation $lsn=0$ possède une unique solution sur l'arête car $lsn(A)\,lsn(B)<0$

Une fois obtenue la coordonnée ξ du point C , on interpole les lst des points A , B et M pour obtenir la lst du point C :

$$lst(C)=\frac{lst(A)*\xi*(\xi-1)}{2}+\frac{lst(B)*(\xi+1)*\xi}{2}-lst(M)*(\xi-1)*(\xi+1)$$

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 .
$\text{si } (\exists j\in Ni\text{, tel que } j\in K) \text{ alors } i\in GRMAEN1$
$\text{si } (\exists j\in Ni\text{, tel que } j\in L) \text{ alors } i\in GRMAEN2$
$\text{si } (\exists (j,k)\in Ni^2\text{, tels que } j\in K \text{ et } k\in L) \text{ ou } (\exists j\in Ni\text{, tel que } j\in K\cap L) \text{ alors } i\in GRMAEN3$

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 [bib73].

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

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.

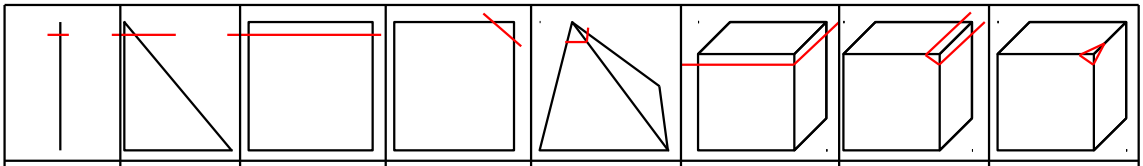
La technique d'annulation de ces degrés de liberté est expliquée en détail dans [R5.03.54, §4.4].

3.2.6 Conditionnement lié à l'enrichissement

3.2.6.1 Généralités sur le conditionnement XFEM

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 [bib29], [bib28]. Béchet et al. [bib28] 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. [bib29] 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 [bib29], [bib28]. 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 3.2.6.1-1, 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 3.2.6.1-1.



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

Figure 3.2.6.1-1 : 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.

3.2.6.2 Les critères estimatifs pour améliorer le conditionnement

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

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

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 [bib22] 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 :

$$\frac{min\left(V_{-1},V_{+1}\right)}{V_{support}}\leq 10^{-\alpha} \qquad \textbf{éq, 3.2.6.2-1}$$

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

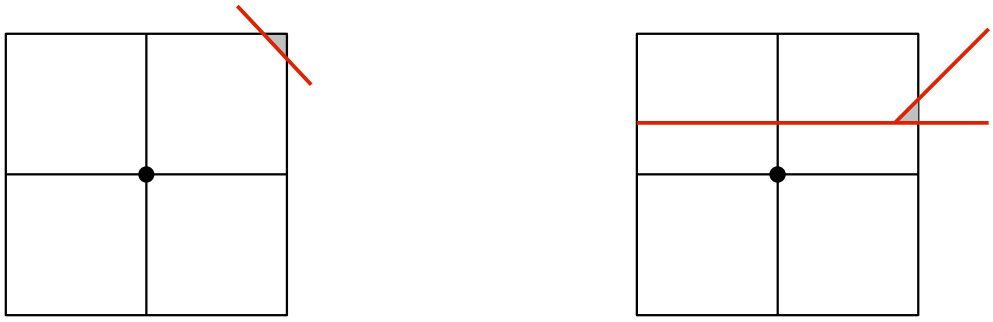


Figure -3.2.6.2-1 : illustration de la mise en place du critère volumique. Le support du nœud coupé par la level set est illustré. On compare les volumes gris et blanc par rapport au volume total du support du nœud (réunion des volumes gris et blanc) dans le cas d'une fissure (gauche) et d'une jonction (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, héxaèdres) 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 :

$$\frac{min\left(\sum_{se_{-1}}\int_{\Omega_{se}}\|\Phi_{n,X}\|^2\,d\,\Omega_{se},\sum_{se_{+1}}\int_{\Omega_{se}}\|\Phi_{n,X}\|^2\,d\,\Omega_{se}\right)}{\sum_{se}\int_{\Omega_{se}}\|\Phi_{n,X}\|^2\,d\,\Omega_{se}}\leq 10^{-\delta}$$

éq 3.2.6.2-2

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 éq 4.3-2, le critère ayant été

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

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 éq 4.3-2, 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 ([bib72]):

$$\int_{\Omega_{se}} \|\Phi_{n,X}\|^2 d\Omega_{se} \approx \|\Phi_{n,X}(G^{se})\|_\infty^2 V^{se} \qquad \text{éq 3.2.6.2-3}$$

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

Au 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 éq 4.3-2, 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:

$$\int_{\Omega_{se}} \|\Phi_{n,X}\|^2 d\Omega_{se} \approx max_{P_i \in \Omega_{se}} \left(\|\Phi_{n,X}(P_i)\|_\infty^2 \right) V^{se} \qquad \text{éq 3.2.6.2-4}$$

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).

$$\{P_i\} = \{G^{se}\} \cup \{\frac{Node_i + G^{se}}{2}\} \qquad \text{éq 3.2.6.2-5}$$

Toutefois les critiques suivantes, peuvent être faites par rapport à ces critères estimatifs :

- d'une part, ils ne garantissent pas le contrôle du conditionnement juste avant l'élimination. Tant que l'élimination n'est pas effectuée, le conditionnement augmente en $10^{n \times \gamma}$ (figure 3.2.6.1-1), ce qui peut conduire à l'arrêt du solveur direct, de manière arbitraire. Ce phénomène est exacerbé avec les éléments quadratiques, voire figure 3.2.6.1-1.
- d'autre part, une analyse de ces différents critères est faite dans [bib72]. 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 [bib28]. Afin de lever cette dernière difficulté il est nécessaire d'estimer des matrices de rigidité locales au moins en en triple précision [bib72].

Pour adresser, la première critique, c'est-à-dire endiguer l'augmentation exponentielle du conditionnement, nous avons mis en place dans le Code_Aster, le pré-conditionneur proposé par Béchet et al. [bib28].

3.2.6.3 Le pré-conditionneur XFEM pour les matrices

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

Il s'agit d'une procédure automatique et algébrique, pour « orthogonaliser » les degrés de liberté associés à un nœud XFEM. En effet, les fonctions d'enrichissement XFEM s'appuient sur les fonctions nodales Φ_i pour décrire, soit la discontinuité du déplacement par les fonctions saut $H\Phi_i$, soit le fond de fissure par les fonctions singulières $F^\alpha\Phi_i$. Les fonctions introduites par l'enrichissement XFEM ne sont pas orthogonales aux fonctions de forme en chaque nœud XFEM. De plus, les fonctions d'enrichissement XFEM et les fonctions nodales, partagent le même support : il arrive des situations où les fonctions XFEM se rapprochent des fonctions nodales, au point de devenir quasiment colinéaires.

L'information sur la colinéarité est transportée dans la matrice de rigidité K : le conditionnement de la matrice de rigidité augmente si la colinéarité augmente au moins en un nœud XFEM. D'un point de vue plus formel, la matrice de rigidité (cf.§3.5.1) dérive de la discrétisation d'une forme bilinéaire symétrique positive. Donc, il existe un produit-scalaire $\langle .,. \rangle_K$ tel que, le terme de la matrice de rigidité $K_{i,x}$ représentant la colinéarité entre la fonction nodale Φ_i et la fonction d'enrichissement $F_x\Phi_i$, se réécrit :

$$K_{i,x} = \langle \Phi_i, F_x\Phi_i \rangle_{\boldsymbol{K}} \qquad \text{éq 3.2.6.3-1}$$

Béchet [bib28] propose la construction d'un pré-conditionneur tel que :

$$K \rightarrow \widetilde{K} = P_c^T K P_c \qquad \text{éq 3.2.6.3-2}$$

On transforme alors le système matriciel original, de la manière suivante :

$$K u = f \rightarrow \begin{cases} \widetilde{K}\,\tilde{u} = \widetilde{f} \\ \widetilde{f} = P_c^T f \\ u = P_c \tilde{u} \end{cases} \qquad \text{éq 3.2.6.3-3}$$

Ainsi, Béchet [bib28] construit la nouvelle matrice $\widetilde{K}$ pour respecter la condition d'orthogonalité suivante:

$$\widetilde{K}_{i,x} = \langle \Phi_i, F_x\Phi_i \rangle_{\widetilde{\boldsymbol{K}}} = \langle P_c\Phi_i, P_c F_x\Phi_i \rangle_{\boldsymbol{K}} = 0 \qquad \text{éq 3.2.6.3-4}$$

En d'autres mots, la nouvelle base fonctionnelle $\left(P_c\Phi_i, P_c F_x\Phi_i\right)$ (par le changement de base P_c) est **orthogonale,** sur le support de chaque nœud XFEM. Par conséquent, le nouveau système matriciel à inverser $\widetilde{K}\tilde{u} = P_c^T f$ est mieux conditionné indépendamment de la position de l'interface, cf. §3.2.6.2.

La construction de $\widetilde{K}$ n'est pas unique. [bib28] propose d'utiliser la factorisation de Cholesky des matrices de rigidité locales, associées à chaque nœud XFEM :

$$K_{loc}^i = \begin{bmatrix} \langle \Phi_i, \Phi_i \rangle_{\boldsymbol{K}} & \langle \Phi_i, H\Phi_i \rangle_{\boldsymbol{K}} & \langle \Phi_i, F^\alpha\Phi_i \rangle_{\boldsymbol{K}} \\ \langle \Phi_i, H\Phi_i \rangle_{\boldsymbol{K}} & \langle H\Phi_i, H\Phi_i \rangle_{\boldsymbol{K}} & \langle H\Phi_i, F^\alpha\Phi_i \rangle_{\boldsymbol{K}} \\ \langle \Phi_i, F^\alpha\Phi_i \rangle_{\boldsymbol{K}} & \langle H\Phi_i, F^\alpha\Phi_i \rangle_{\boldsymbol{K}} & \langle F^\alpha\Phi_i, F^\alpha\Phi_i \rangle_{\boldsymbol{K}} \end{bmatrix} \qquad \text{éq 3.2.6.3-5}$$

La matrice K_{loc}^i est symétrique définie positive. Elle admet donc une factorisée de Cholesky, c'est-à-dire, qu'il existe une matrice triangulaire supérieure S_i telle que :

$$K_{loc}^i = S_i^T S_i \qquad \text{éq 3.2.6.3-6}$$

Dans une configuration de stockage optimale contiguë des degrés de liberté, [bib 28] choisit alors un pré-conditionneur diagonal par bloc, tel que :

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

$$P_c = \begin{bmatrix} P_c^1 & 0 & 0 & \dots & \dots & \dots \\ 0 & P_c^2 & 0 & \dots & 0 & \dots \\ 0 & 0 & P_c^3 & \dots & \dots & \dots \\ \dots & \dots & \dots & \ddots & \dots & \dots \\ \dots & 0 & \dots & \dots & P_c^i & \dots \\ \dots & \dots & \dots & \dots & \dots & \ddots \end{bmatrix} \quad \text{éq 3.2.6.3-7}$$

où $$P_c^i = \begin{cases} S_i^{-1} \text{ si } i \text{ est un noeud XFEM} \\ I_d \text{ sinon} \end{cases} \quad \text{éq 3.2.6.3-8}$$

avec S_i^{-1} l'inverse de la matrice factorisée de Cholesky. La nouvelle matrice de rigidité $\widetilde{K}$ vérifie à l'échelle locale du support du nœud XFEM :

$$\widetilde{K}_{loc}^i = \left(P_c^i\right)^T K_{loc}^i P_c^i = \left(S_i^{-1}\right)^T K_{loc}^i S_i^{-1} = \left(S_i^{-1}\right)^T \left(S_i^T S_i\right) S_i^{-1} = I_d \quad \text{éq 3.2.6.3-9}$$

I_d désigne la matrice identité. I_d respecte donc la condition d'orthogonalité de l'équation 3.2.6.3-4 .

Dans la pratique, il est nécessaire de mettre à l'échelle la nouvelle matrice $\widetilde{K}_{loc}^i$ avec le reste de la matrice de rigidité. On préfère donc le pré-conditionneur suivant au pré-conditionneur 3.2.6.3-8 :

$$P_c^i = \sqrt{scal} \times S_i^{-1} \quad \text{éq 3.2.6.3-10}$$

où $scal$ est un coefficient de mise à l'échelle tel que : $scal = \dfrac{max(|diag(K)|) + min(|diag(K)|)}{2}$

Par conséquent, la nouvelle matrice locale pré-conditionnée respectant la condition d'orthogonalité 3.2.6.3-4 , est :

$$\widetilde{K}_{loc}^i = \left(P_c^i\right)^T K_{loc}^i P_c^i = scal \times I_d \quad \text{éq 3.2.6.3-11}$$

Par ailleurs, l'utilisation d'une factorisation de Cholesky est risquée si le conditionnement de la matrice de rigidité locale K_{loc}^i se dégrade. En cas d'échec de la factorisation de Cholesky, on orthogonalise la matrice locale à l'aide d'une SVD (décomposition en valeurs singulières):

$$K_{loc}^i = U_i D_i U_i^T \quad \textbf{éq 3.2.6.3-12}$$

où U_i est une matrice orthogonale et D_i est une matrice diagonale à valeurs strictement positives.

On effectue alors alternativement, le choix de pré-conditionneur, suivant :

$$P_c^i = \sqrt{scal} \times U_i \sqrt{D_i^{-1}} \quad \text{éq 3.2.6.3-13}$$

Il est facile de vérifier que ce choix permet de respecter la condition d'orthogonalité 3.2.6.3-4 :

$$\widetilde{K}_{loc}^i = \left(P_c^i\right)^T K_{loc}^i P_c^i = \left(\sqrt{scal} \times U_i \sqrt{D_i^{-1}}\right)^T \left(U_i D_i U_i^T\right) \left(\sqrt{scal} \times U_i \sqrt{D_i^{-1}}\right) = scal \times I_d$$

En résumé, la construction du pré-conditionneur de Béchet [bib 28] procède en 4 étapes :

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

1. l'extraction des matrices locales K_{loc}^{i} dans la matrice de rigidité (c'est-à-dire des matrices blocs associées aux ddls portés par les nœuds XFEM n° i).
2. le calcul des matrices de pré-conditionnement locales P_{c}^{i} (cf. équation 3.2.6.3-10 ou 3.2.6.3-13),
3. l'assemblage du pré-conditionneur (cf. équation 3.2.6.3-7),
4. enfin, la transformation du système matriciel (cf. équations 3.2.6.3-2 et 3.2.6.3-3).

Remarque :

Notons que la transformation du système matriciel présente une difficulté informatique notable dans le code_aster, compte-tenu de la complexité de la structure de données de stockage des matrices (cf. [D4.06.10] et [D4.06.07]).

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 1ère fois dans [bib24] pour le 2D et dans [bib32] 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 « condense » ainsi le grand nombre de possibilités de découpe à quelques configurations. En d'autres mots, nous formulons le « clustering » d'un ensemble de grande taille (nombre de configurations d'intersection) vers un ensemble de petite taille (nombre réduit de configurations de découpe dans un tétraèdre).

En fait, l'algorithme de « condensation » construit une application bijective, du polyèdre parent, vers quelques configurations de découpe de référence. Cette application peut-être décomposée en deux opérations bijectives :

1 – Une bijection (explicite) vers la découpe d'un tétraèdre : On réalise donc une phase préalable qui consiste à découper de manière systématique les quadrangles, pyramides, pentaèdres et hexaèdres en tétraèdres. Cette bijection n'est pas unique. Comme on peut le voir sur la Figure 3.3.1-1, il existe deux bijections pour diviser un quadrangle en deux triangles (de même pour une pyramide), 6 bijections pour diviser un pentaèdre en 3 tétraèdres et 6 bijections pour diviser un hexaèdre en 2 pentaèdres qui sont ensuite divisés en tétraèdres. Dans le cas de l'hexaèdre, on obtient ainsi deux fois chaque configuration de découpe. Le nombre maximal de bijections distinctes pour diviser un hexaèdre en 6 tétraèdres est donc de $\frac{6^3}{2}=108$.

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

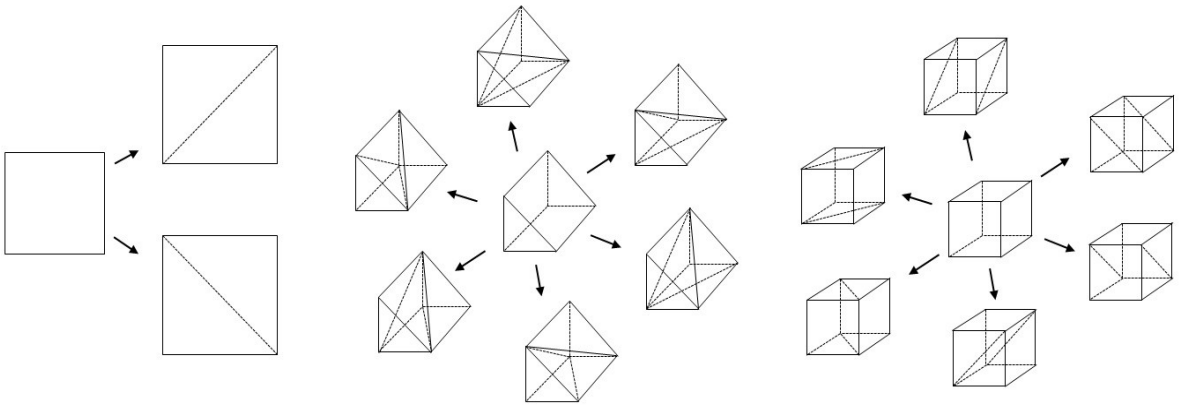


Figure 3.3-1: Division d'un quadrangle en deux triangles (à gauche), division d'un pentaèdre en 3 tétraèdres (au milieu) et division d'un hexaèdre en 2 pentaèdres (à droite)

Dans les § 3.3.1 à § 3.3.3 , on présente la bijection retenue par défaut pour la division des éléments non simpliciaux en tétraèdres (ou triangles pour le cas 2D).
Il s'agit d'une bijection, car on construit une application (inversible) qui associe la numérotation relative des nœuds sommets dans un sous-tétraèdre, à la numérotation absolue des nœuds dans le maillage. L'application directe est stockée dans la table de connectivité (voire [D4.10.02] : structures de données XFEM).
2 – Une bijection (implicite) vers une configuration de découpe de référence : d'une part, on identifie la configuration de découpe de référence correspondante au tétraèdre à découper, d'autre part, on retourne « géométriquement » le tétraèdre, pour le « superposer » à cette configuration de découpe de référence (par exemple Figure 3.3.1-2). En clair, on identifie les nœuds du sous-tétraèdre aux nœuds de l'élément de découpe de référence (c'est l'application directe de la bijection). On « condense » alors les multiples possibilités de découpe, liées aux différentes possibilités d'ordonnancement des arêtes et des points d'intersection.
Par la suite, cette opération facilite l'ordonnancement des nœuds milieux des sous-tétraèdres quadratiques pour le stockage informatique. En effet, l'ordonnancement des nouveaux nœuds milieux ne dépend plus des faces ou des arêtes où ils sont calculés, mais d'une seule configuration de découpe de référence. Par conséquent, les nouveaux nœuds milieux calculés sont nécessairement ordonnancés de la même manière, dans le sous-tétraèdre de découpe de référence. Ensuite, ces nœuds milieux sont positionnés sur les arêtes du tétraèdre par l'application inverse (de l'identification des nœuds ci-dessus). La bijection utilisée est donc tacite.
Les procédures d'ordonnancement et de calcul des points milieux sont explicitées ci dessous.

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

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

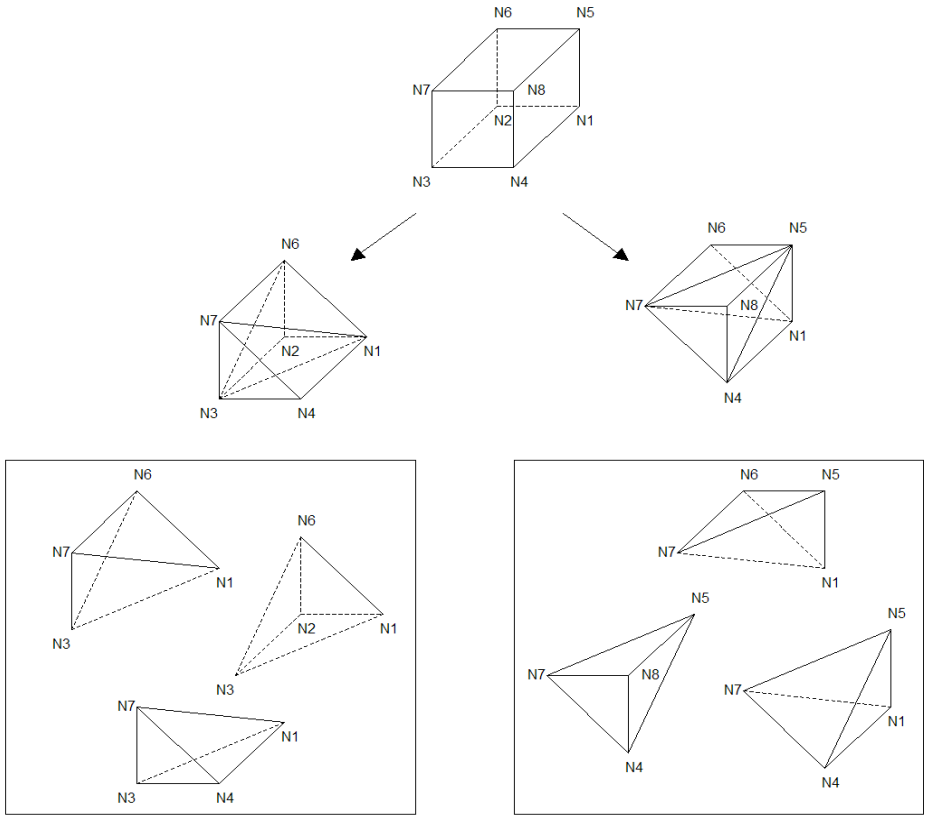


Figure 3.3.1-1 : Division d'un hexaèdre (*hexa8*) en tétraèdres

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

hexaèdre	tétraèdre
N1 N2 N3 N4 N5 N6 N7 N8	*N7 N4 N3 N1*
	N1 N6 N2 N3
	N3 N6 N7 N1
	N6 N1 N5 N7
	N4 N7 N8 N5
	N4 N5 N1 N7

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

On note que ce découpage est le choix effectué par défaut mais 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.

Si l'élément est quadratique (`HEXA20`, `PENTA15`, ...), on utilise la même subdivision en tétraèdres, que celle explicitée ci-dessus.
Par ailleurs, il faut tenir compte des nœuds milieux sur les nouvelles arêtes générées par la découpe : ces nœuds milieux coïncident avec les nœuds milieux de l'élément complet. On rajoute donc virtuellement à l'élément, les nœuds milieux nécessaires pour retrouver l'élément complet de Figure 3.3.1-2.

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

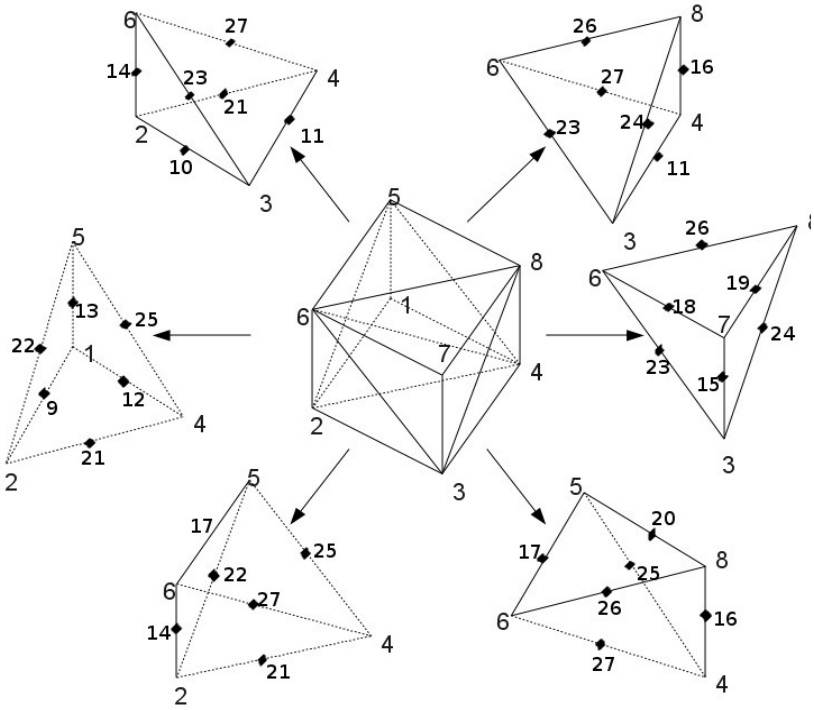


**Figure 3.3.1-2: Division d'un hexaèdre quadratique (hexa 20)
via l'élément complet (hexa 27)**

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

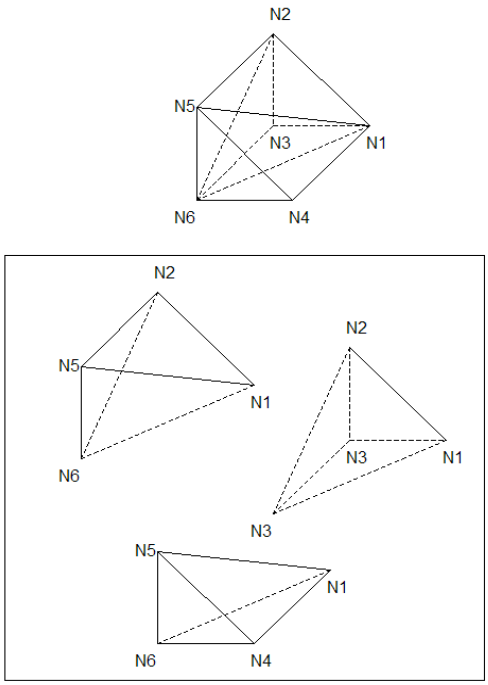


Figure 3.3.2-1 : Schéma de division d'un pentaèdre en tétraèdres

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

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

pentaèdre	tétraèdre
$N1N2N3N4N5N6$	$N5N4N6N1$
	$N1N2N3N6$
	$N6N2N5N1$

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

Pour un pentaèdre quadratique (*penta15*) on conserve la subdivision ci-dessus.
Par ailleurs, il faut tenir compte des nœuds milieux sur les nouvelles arêtes générées par la découpe : ces nœuds milieux coïncident avec les nœuds milieux de l'élément complet. On rajoute donc à l'élément, les nœuds milieux nécessaires pour retrouver l'élément complet Figure 3.3.2-2.

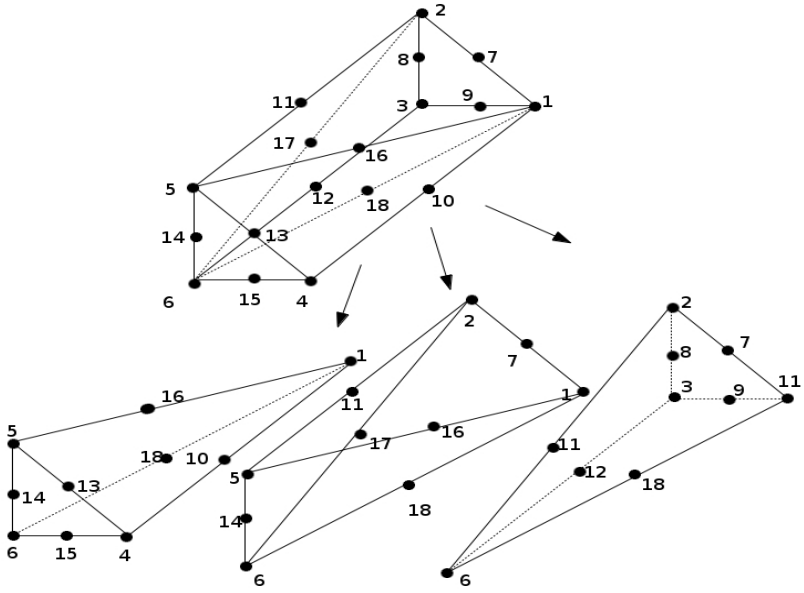


Figure 3.3.2-2: Division d'un pentaèdre quadratique

3.3.3 Phase préalable de découpe des pyramides pour se ramener à des tétraèdres

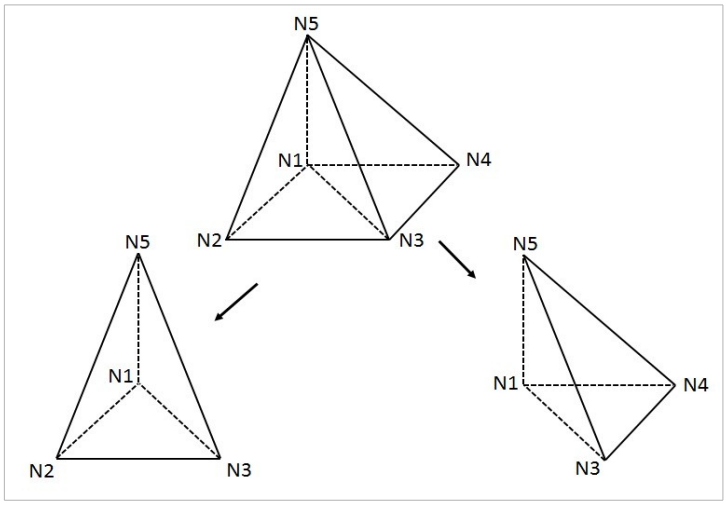


Figure 3.3.3-1 : Schéma de division d'une pyramide en tétraèdres

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

Une pyramide se divise alors en 2 tétraèdres, répertoriés dans le tableau suivant :

pyramide	tétraèdre
$N1N2N3N4N5$	$N1N3N4N5$
	$N1N2N3N5$

Tableau 3.3.3-1 : Division d'une pyramide en tétraèdres

Pour un pyramide quadratique (*pyra13*) on conserve la subdivision ci-dessus.
Par ailleurs, il faut tenir compte des nœuds milieux sur les nouvelles arêtes générées par la découpe : ces nœuds milieux coïncident avec les nœuds milieux de l'élément complet. On rajoute donc à l'élément, les nœuds milieux nécessaires pour retrouver l'élément complet Figure 3.3.2-2.

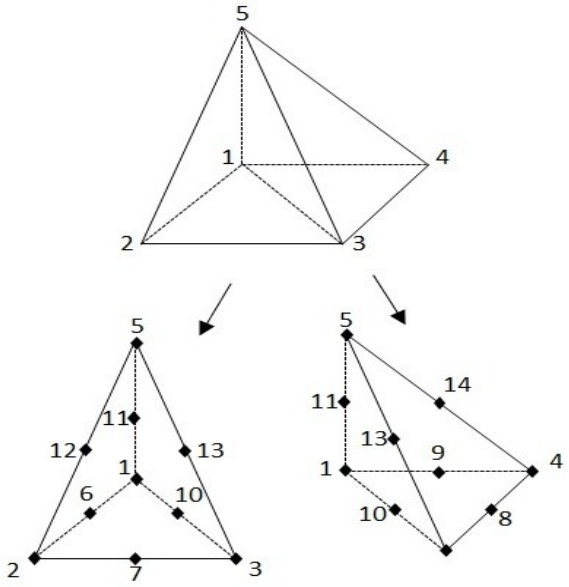


Figure 3.3.3-2: Division d'une pyramide quadratique

Remarque :

La subdivision d'un quadrangle en deux triangles est similaire à la subdivision d'une pyramide en deux tétraèdres.

3.3.4 Sous-découpage d'un tétraèdre en sous-tétraèdres

Le tétraèdre de référence est défini sur la Figure 3.3.4-1. On détermine les points d'intersection Pi entre la surface $lsn^h=0$ et les arêtes du tétraèdre.

Soit n le nombre de points d'intersection Pi .

À chaque point d'intersection Pi , on associe deux entiers : Ai et NSi

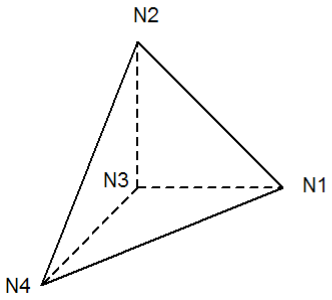
- Ai est le numéro de l'arête sur laquelle se trouve Pi (par exemple si Pi se trouve sur l'arête d'extrémités $N2-N3$, alors $Ai=4$). Dans le cas où Pi coïncide avec un nœud sommet du tétraèdre, Ai vaut 0,
- NSi est le numéro du nœud sommet dans le cas où Pi coïncide avec un nœud sommet du tétraèdre (par exemple si Pi coïncide avec $N3$, alors $NSi=3$). Dans le cas où Pi se trouve sur une arête, NS vaut 0.

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

Remarque :

Le produit de Ai par NSi vaut toujours 0.

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



arête	extrémités
1	N1 – N2
2	N1 – N3
3	N1 – N4
4	N2 – N3
5	N2 – N4
6	N3 – N4

Figure 3.3.4-1 : Tétraèdre de référence

L'approximation de la level set utilisant les fonctions de forme du tétraèdre, la surface $lsn^h=0$ est alors un plan avec des éléments linéaires, et une surface de géométrie éventuellement courbe, avec des éléments quadratiques.
Le problème se ramène donc à la découpe d'un tétraèdre par une surface. Examinons les différents cas possibles suivant la valeur de n (nombre de points d'intersection Pi). On peut déjà éliminer les cas triviaux où aucun sous-découpage n'est nécessaire :

- lorsque $n<3$ la trace de la surface dans le tétraèdre est un sommet ou une arête. Si la surface $lsn^h=0$ n'est pas un plan, la géométrie de la level-set est alors approximée par l'arête du tétraèdre.
- lorsque $n=3$ et que les 3 points d'intersection sont des points sommets, la trace de la surface dans le tétraèdre est une face du tétraèdre. Si la surface $lsn^h=0$ n'est pas un plan, la géométrie de la level-set est alors approximée par la face du tétraèdre.

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

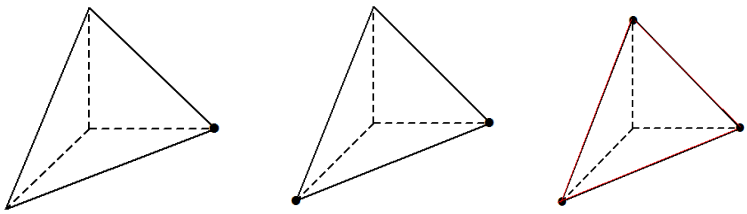


Figure 3.3.4-2 : Cas sans sous-découpage

3.3.4.1 Trois points d'intersection dont deux points sommets

$P1$ 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 $E2$.

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

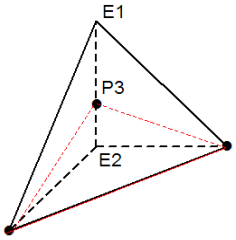


Figure 3.3.4.1-1 : Cas général où $n=3$ dont 2 points sommet

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 3.3.4.1-1 : Sous-tétraèdres

Pour réduire les possibilités de découpe, (puisqu'il existe au moins 6 possibilités de coupe compte-tenu des 6 arêtes dans le sous-tétraèdre) on construit une bijection vers un **unique** élément de découpe Figure 3.3.4.1-2.Les nœuds A, B, C, D, E, F, G, H définissent les inconnues d'ordonnancement de la découpe du sous-tétraèdre. Les inconnues d'ordonnancement vont être associées de manière unique aux nœuds de l'élément parent. Cette association construit une bijection (implicite).
Rappelons que,
- l'ordonnancement est important d'un point de vue informatique, pour assurer le stockage de la maille virtuelle de sous-découpe. En particulier, pour localiser les points d'intersection et des points milieux dans les structures de données XFEM,
- les nœuds de numéro compris entre 1000 et 1999 représentent les points d'intersections, ces nœuds sont calculés avec la procédure décrite au §4.2.3 de [D4.10.02].
- les nœuds de numéro supérieur à 2000 représentent les nouveaux nœuds milieux issus de la découpe, ces nœuds sont calculés avec la procédure décrite au §4.2.3 de [D4.10.02]

Remarque :

Pour cette configuration de découpe particulière, la construction de la bijection n'est pas entièrement déterministe. En effet, les 2 sous-tétraèdres sont symétriques par rapport au plan de découpe. D'un point de vue topologique, les inconnues $\{A, D, E\}$ sont symétriques aux inconnues $\{B, F, G\}$. Pour rendre déterministe la découpe d'un maillage donné, nous avons choisi la convention arbitraire suivante: prendre A comme le premier nœud de l'arête du premier point d'intersection.
Par analogie aux configurations moléculaires, la découpe va potentiellement générer deux configurations dites « énantiomères ».

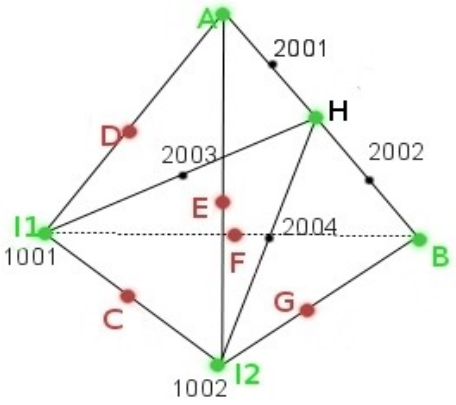


Figure 3.3.4.1-2: élément de découpe pour la configuration 4 points d'intersection dont 2 nœuds sommets

3.3.4.2 Trois 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 $E3$.

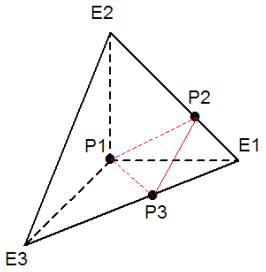


Figure 3.3.4.2-1 : Cas général où $n=3$ dont un point sommet

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 3.3.4.2-1 : Sous-tétraèdres

Pour réduire les possibilités de découpe, on construit une bijection vers un **unique** élément de découpe Figure 3.3.4.2-2. Les nœuds A , B , C , E , F , G , H définissent les inconnues d'ordonnancement de la découpe du sous-tétraèdre. Les inconnues d'ordonnancement vont être associées de manière unique aux nœuds de l'élément parent. Cette association construit une bijection (implicite).

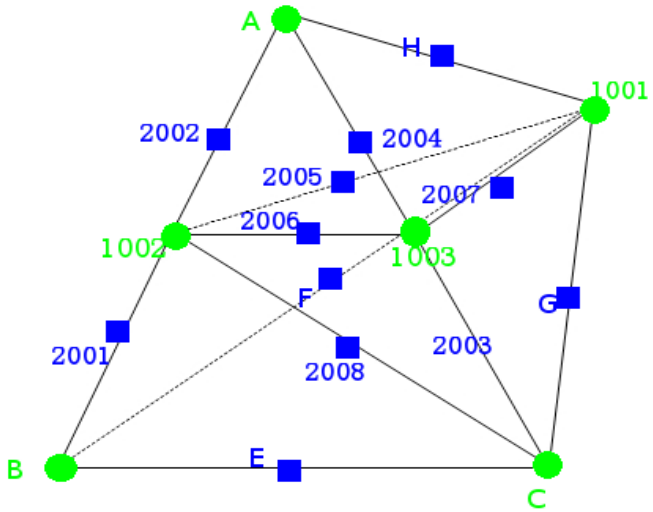


Figure 3.3.4.2-2: élément de découpe pour la configuration 3 points d'intersection dont 1 nœud sommet

3.3.4.3 Cinq points d'intersection dont deux points sommets

Ce cas de découpe correspond à une configuration rasante. C'est un cas particulier de la découpe précédente pour lequel la surface $lsn=0$ passe en plus par un autre nœud sommet. La découpe est alors strictement identique à la découpe précédente (Paragraphe [§3.3.4.2]). Cette configuration de

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

découpe est représentée Figure 3.3.4.3-1 avec à droite la configuration à 3 points d'intersection et un point sommet à laquelle on se ramène.

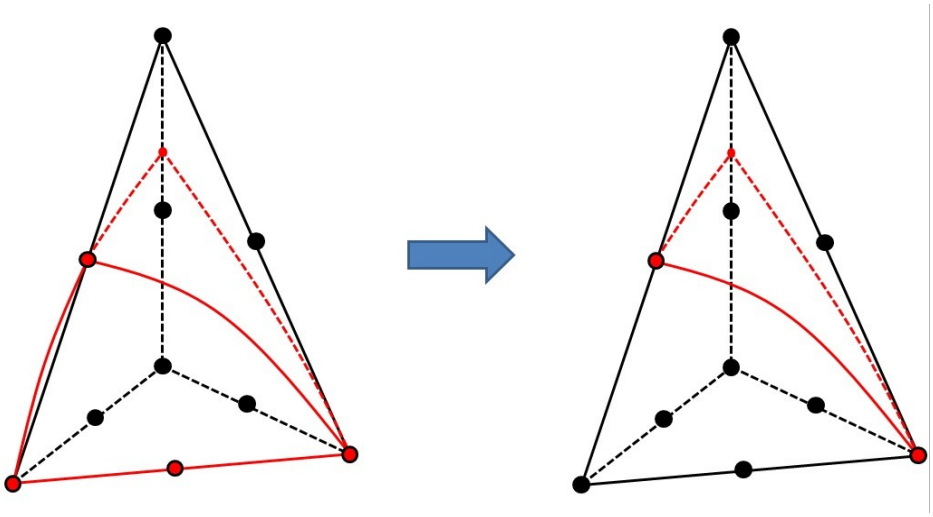


Figure 3.3.4.3-1: Configuration rasante à 5 points d'intersection (à gauche) et configuration saine correspondante (à droite)

3.3.4.4 Trois points d'intersection dont aucun point sommet

On distingue au moins 4 possibilités de coupe, répertoriées Figure 3.3.4.4-1. Ces différentes possibilités ne sont pas traitées au cas par cas.

On construit une bijection vers un **unique** élément de découpe Figure 3.3.4.2-2. Les nœuds A, B, C, D, E, F, G définissent les inconnues d'ordonnancement de la découpe du sous-tétraèdre. Les inconnues d'ordonnancement vont être associées de manière unique aux nœuds de l'élément parent. Cette association construit une bijection (implicite).

Exemple de procédure d'identification:
Dans chacune des configurations listées Figure 3.3.4.4-1, on associe les inconnues d'ordonnancement nœuds $\{A, B, C, D\}$ *aux nœuds* $\{N_1, N_2, N_3, N_4\}$ *du sous tétraèdre. Comme le nœud* A *appartient aux trois arêtes coupées, on l'identifie facilement au nœud correspondant dans chacune des configurations listées. Ensuite, on identifie alors* $\{B, C, D\}$ *puisque les nœuds* B, C, D *appartiennent respectivement à la première, deuxième, troisième arête intersectée, et sont différents du nœud* A *précédemment identifié. Il en résulte le tableau de correspondance Tableau 3.3.4.4-1*
Cette procédure d'identification permet de construire implicitement une bijection.

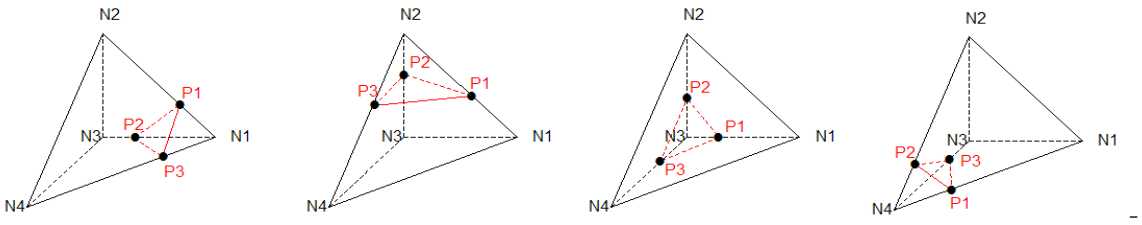


Figure 3.3.4.4-1 : Configurations d'un tétraèdre avec 3 arêtes intersectées

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

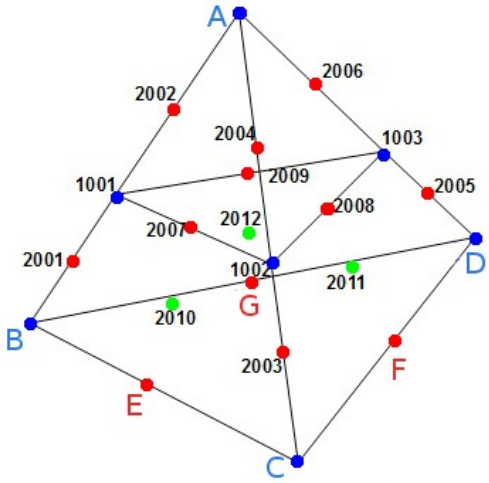


Figure 3.3.4.4-2: élément de découpe pour la configuration 3 points d'intersection et aucun nœud sommet

De Figure 3.3.4.4-2 à Figure 3.3.4.4-1	
(A,B,C,D)	(N_1,N_2,N_3,N_4)
(A,B,C,D)	(N_2,N_1,N_3,N_4)
(A,B,C,D)	(N_3,N_1,N_2,N_4)
(A,B,C,D)	(N_4,N_1,N_2,N_3)

Tableau 3.3.4.4-1 : Identification des nœuds de l'élément de découpe aux nœuds du sous-tétraèdre dans chaque configuration

3.3.4.5 Quatre points d'intersection dont un point sommet

Ce cas de découpe correspond à une configuration rasante. C'est un cas particulier de la découpe précédente pour lequel la surface $lsn=0$ passe en plus par l'un des nœuds sommets. La découpe est alors strictement identique à la découpe précédente (Paragraphe [§3.3.4.4]). Cette configuration de découpe est représentée Figure 3.3.4.5-1 avec à droite la configuration à 3 points d'intersection à laquelle on se ramène.

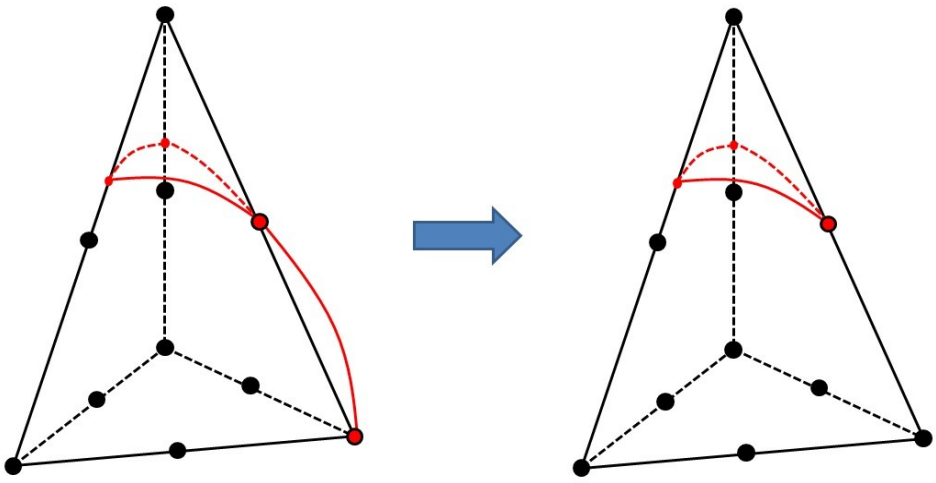


Figure 3.3.4.5-1: Configuration rasante à 4 points d'intersection (à gauche) et configuration saine correspondante (à droite)

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

3.3.4.6 Six points d'intersection dont deux points sommet

Ce cas de découpe correspond à une configuration rasante. C'est encore un cas particulier de la découpe précédente pour lequel la surface $lsn=0$ passe en plus par deux des nœuds sommets. La découpe est alors strictement identique à la découpe précédente (Paragraphe [§3.3.4.4]). Cette configuration de découpe est représentée Figure 3.3.4.7-1 avec à droite la configuration à 3 points d'intersection à laquelle on se ramène.

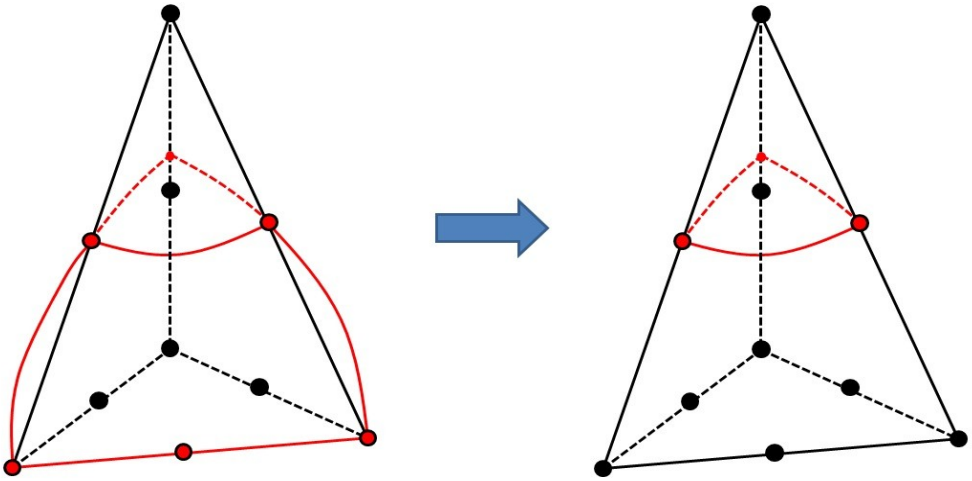


Figure 3.3.4.6-1: Configuration rasante à 6 points d'intersection (à gauche) et configuration saine correspondante (à droite)

3.3.4.7 Quatre points d'intersection

On distingue au moins 3 possibilités de coupe, répertoriées Figure 3.3.4.7-1. Ces différentes possibilités ne sont pas traitées au cas par cas.

On construit une bijection vers un **unique** élément de découpe Figure 3.3.4.7-2 . Les nœuds A , B , C , D , E , F ,définissent les inconnues d'ordonnancement de la découpe du sous-tétraèdre. Les inconnues d'ordonnancement vont être associées de manière unique aux nœuds de l'élément parent. Cette association construit une bijection (implicite).

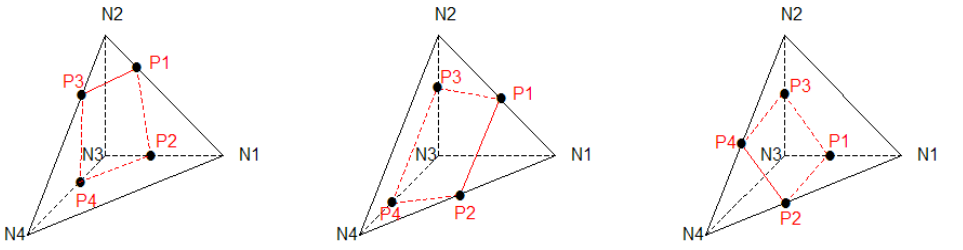


Figure 3.3.4.7-1 : Configurations d'un tétraèdre avec 4 arêtes intersectées

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

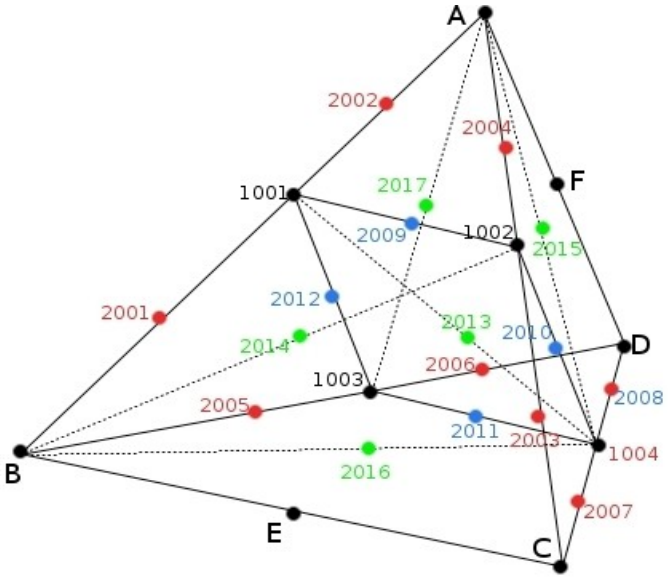


Figure 3.3.4.7-2: élément de découpe pour la configuration 4 points d'intersection et aucun nœud sommet

3.3.4.8 Cinq points d'intersection dont un point sommet

Ce cas de découpe correspond à une configuration rasante. C'est un cas particulier de la découpe précédente pour lequel la surface $lsn=0$ passe en plus par l'un des nœuds sommets. La découpe est alors strictement identique à la découpe précédente (Paragraphe [§ 3.3.4.7]). Cette configuration de découpe est représentée Figure 3.3.7-1 avec à droite la configuration à 4 points d'intersection à laquelle on se ramène.

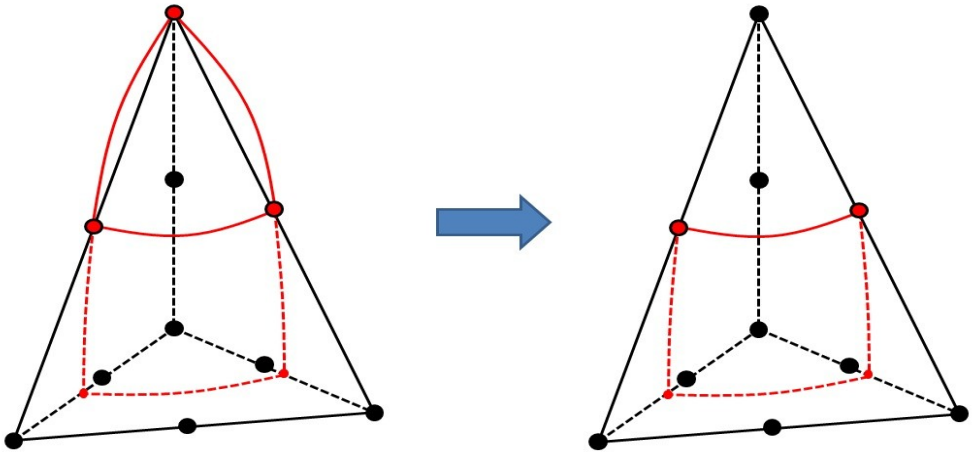


Figure 3.3.4.8-1: Configuration rasante à 5 points d'intersection (à gauche) et configuration saine correspondante (à droite)

3.3.4.9 Quatre points d'intersection dont un point sommet

Cette configuration de découpe correspond à une configuration rasante. Elle ne se ramène à aucune configuration précedemmment décrite. On la distingue de la configuration décrite dans le Paragraphe [§3.3.4.5]) (qui comporte également 4 points d'intersection dont un point sommet) car le nœud sommet non intersecté sur l'arrête rasante n'appartient qu'à deux arêtes intersectées (au lieu de 3 dans le cas

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

décrit dans le Paragraphe [§3.3.4.5]). On distingue au moins 24 possibilités de coupe, dont 3 sont répertoriées dans la Figure 3.3.4.9-1. Ces différentes possibilités ne sont pas traitées au cas par cas.

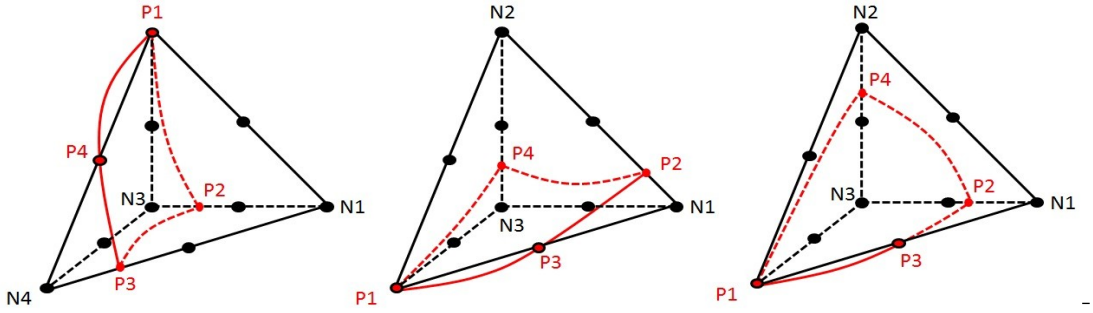


Figure 3.3.4.9-1 : Configurations d'un tétraèdre avec 4 points d'intersection dont un point sommet

On construit une bijection vers un **unique** élément de découpe Figure 3.3.4.9-2 . Les nœuds A , B , C , D , E , F ,définissent les inconnues d'ordonnancement de la découpe du sous-tétraèdre. Les inconnues d'ordonnancement vont être associées de manière unique aux nœuds de l'élément parent. Cette association construit une bijection (implicite). Cette configuration de découpe donnent naissance à 5 sous-tétraèdres enfants.

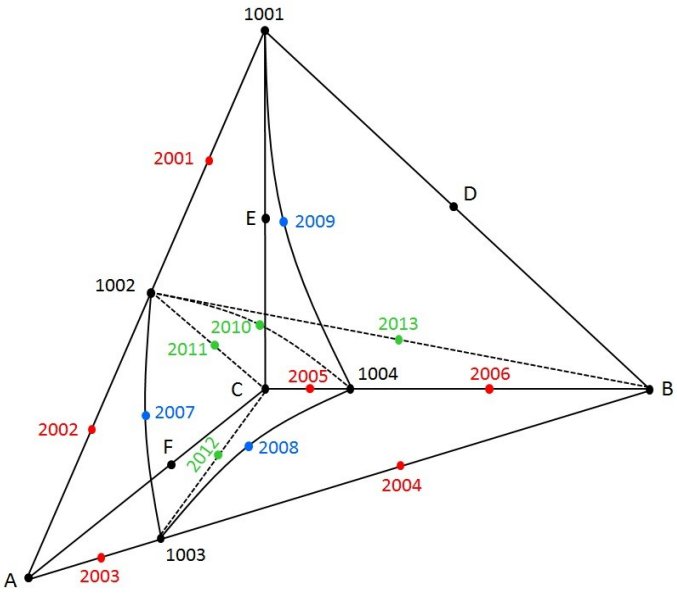


Figure 3.3.4.9-2: élément de découpe pour la configuration 4 points d'intersection dont un nœud sommet

3.3.5 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 le 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

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

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.

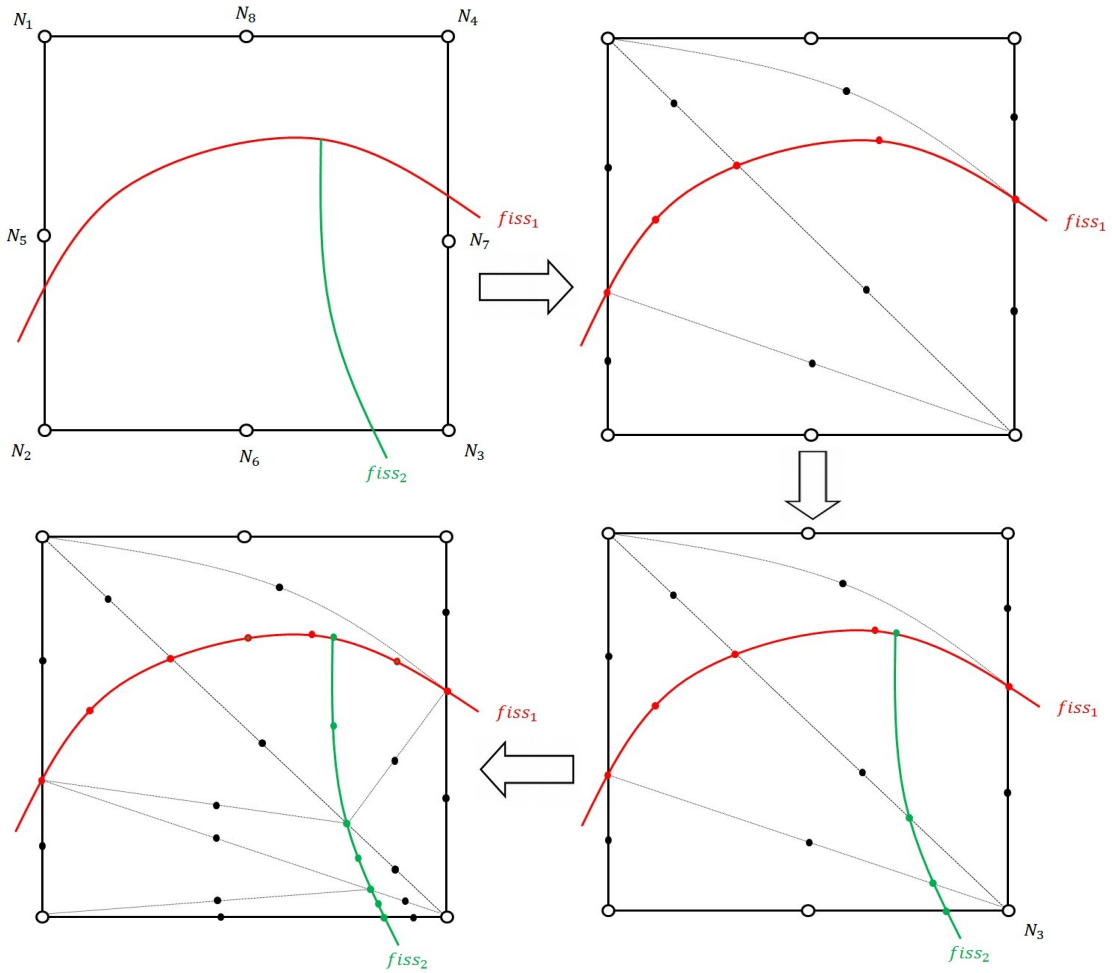


Figure 3.3.5-1: Exemple de multi-découpage en 2D

3.3.6 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.

3.3.6.1 Cas du tétraèdre

Le cas engendrant le plus grand nombre de sous-éléments est celui décrit au paragraphe [§3.3.4.7], qui aboutit à 6 sous-éléments.

3.3.6.2 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.

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

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.

3.3.6.3 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 $(6+6+4)+(6+6+4)=32$.

3.3.6.4 Cas du multi-découpage

Un élément fini XFEM peut être découpé par 4 fissures au maximum dans *CodeAster.* Le nombre maximum de sous éléments générés est alors difficile à évaluer et un majorant du nombre maximal de sous élément serait trop élevé. En effet, si on prend le nombre maximal de sous élément généré par une seule fissure, soit 32, et que l'on considère que les 32 tétraèdres peuvent donner naissance à 6 tétraèdres pour chacune des fissures suivantes, on obtient $32*6^3=6912$. Ce nombre de sous éléments d 'intégration pour un hexaèdre est tout à fait inatteignable car il va de soi que les tétraèdres issus du 1er découpage ne seront pas tous découpés par les fissures suivantes, mais il est difficile d'exhiber un meilleur majorant. On fait le choix de fixer le nombre maximal de sous éléments pour les éléments multi-fissurés à 4 fois le nombre maximal de sous éléments pour une seule fissure soit $32*4=128$ en 3D et $4*6=24$ en 2D. Ces nombres peuvent aisément être dépassés pour des éléments coupés par 3 ou 4 fissures. Dans ce cas, la procédure de découpage renvoie une erreur et l'utilisateur est invité à raffiner le maillage utilisé afin d'éviter que des éléments finis ne soient striés de fissures, ce qui entrainerait par ailleurs d'importants problèmes de conditionnement (voir paragraphe [§3.2.6]).

3.3.7 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 3.3.7-1 : 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 Pi .

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

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). La valeur de la level set normale qui lui est attribuée est obtenue par interpolation des lsn des nœuds $N1$ à $N8$. Un réajustement peut alors être nécessaire pour assurer l'unicité de la solution de l'équation $lsn=0$ le long de la diagonale (cf. réajustement des level set [§2.2.4]) lors de la découpe des éléments enfants en sous éléments d'intégration. Préalablement à toute découpe, on effectue donc une boucle sur les arêtes des éléments enfants (cela a déjà été effectué pour les arêtes des éléments parents lors du réajustement des level set) pour détecter les situations pathogènes et effectuer les réajustements qui s'imposent, exactement comme décrit au [§2.2.4].*

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

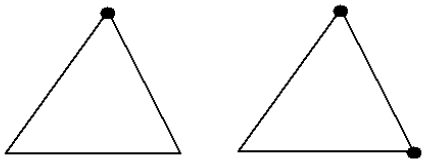


Figure 3.3.7-2 : Cas sans sous-découpage

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

Si on a 1 point sommet et 1 point non sommet, $P1$ est forcement sommet. On détermine les sommets de l'arête du 2e point $N2$ et $N3$ pour découper le triangle en 2 sous triangles $P1\,N2\,P2$ et $P1\,N3\,P2$. Si le triangle est quadratique, on détermine les points milieux de l'arête du 2e point $Q1$ et $Q2$ et celui de la fissure $Q3$.



Figure 3.3.7-3 : 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 $N1\,P1\,P2$, $P1\,P2\,N3$ et $P1\,N2\,N3$. $N1$ 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 1er point $Q1$ et $Q2$, ceux du 2e point $Q3$ et $Q4$, celui de la fissure $Q5$ et celui de la nouvelle arête $Q6$.

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



Figure 3.3.7-4 : Cas de découpage linéaire (à gauche) et quadratique (à droite) avec 2 points d'intersection sans sommet

3.3.8 Algorithmes de sous-découpage

Dans les paragraphes §3.3.1 à §3.3.7, nous avons expliqué le découpage géométrique des éléments en sous-cellules d'intégration (`Tetra10` en 3D, `Tri6` en 2D), sans expliquer comment sont calculées les coordonnés des points d'intersection et des nouveaux nœuds milieux dans le maillage virtuel. Dans ce paragraphe, nous allons donc décrire les algorithmes de calcul des coordonnées des nouveaux points, générés par la découpe.

Pour chaque type de points, il existe un algorithme de positionnement spécifique :

- Le positionnement des points d'intersection permet de délimiter une frontière explicite, bien que l'interface/fissure soit définie par une équation implicite $lsn^h = 0$. Précisément, le lieu des points d'intersection est calculé en prenant l'intersection entre les arêtes du maillage et l'iso-zéro de la level-set. Les points d'intersection sont ensuite stockés dans une structure de donnée auxiliaire, en complétion des points du maillage [`D4.10.02`].

- Le positionnement des points milieux permet d'améliorer la description d'une level-set courbe, dans le cas d'éléments quadratiques. En présence d'éléments à bords courbes, la courbure des arêtes de l'élément quadratique est conservée, en positionnant des nœuds milieux rigoureusement sur les arêtes lors de la découpe. En revanche, une découpe avec des éléments à bords droits altérerait la courbure de l'élément originel. Pour capter la courbure de la level-set et des bords courbes d'éléments quadratiques, on calcule donc plusieurs types de points milieux (voir § Error: Reference source not found). Plusieurs algorithmes sont implémentés, pour capter les différents types de points milieux envisageables. Ces nouveaux points milieux sont alors stockés dans une structure de donnée auxiliaire, en complétion des points du maillage [`D4.10.02`].

Par la suite, on adopte la notation suivante :

- $\xi = (\xi_i)_{i \in [1, dim]}$ désigne les coordonnées de référence dans l'élément parent. Ces coordonnées varient généralement entre $[0,1]$ ou entre $[-1, 1]$. Rappelons que les fonctions de formes ou polynômes d'interpolation, sont paramétrées dans Aster, grâce aux coordonnées de référence [`R3.01.01`],

- $X = (X_i)_{i \in [1, dim]}$ désigne les coordonnées d'un point dans l'espace réel. Si le point courant est un nœud, ce sont les coordonnées géométriques du nœud dans le maillage. Si le point courant n'est pas un nœud, il s'agit des coordonnées géométriques interpolées dans l'élément parent $X(\xi) = \sum_{1 \leqslant j \leqslant nnop} X^j \Phi_j(\xi)$.

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

3.3.8.1 Calcul des points d'intersection

N. Moës propose de poser le problème de recherche des points d'intersection dans l'espace de référence parent. En clair, pour calculer les coordonnées réelles d'un point d'intersection, on recherche au préalable les coordonnées dans l'espace de référence ξ. On déduit ensuite les coordonnées réelles grâce à l'interpolation des coordonnées géométriques dans l'élément parent.

Pour calculer le point d'intersection P entre une arête AB et l'iso-zéro, on résout le problème suivant :

$$P \in AB$$
$$lsn^h(P) = 0$$

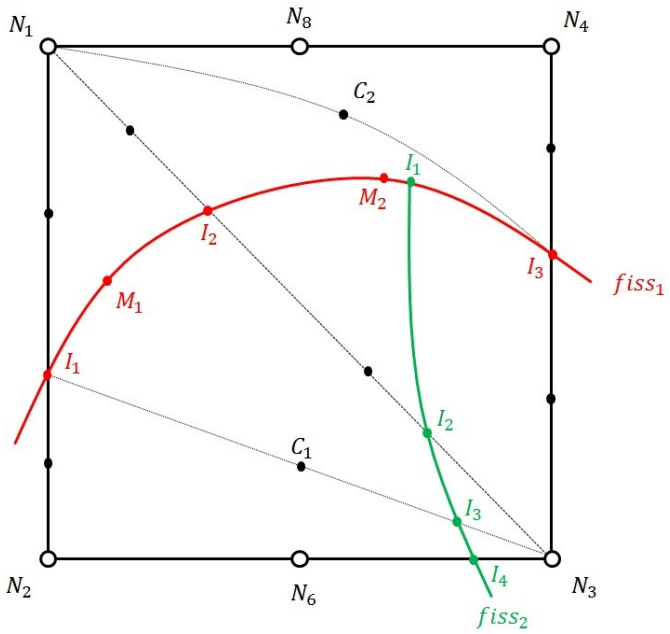


Figure 3.3.8.1-1 : Recherche du point d'intersection I_1 sur l'arête $I_2 I_3$ qui est courbe dans l'espace de référence. On prend alors en compte la courbure de $I_2 I_3$ avec le nœud milieu M_2

Exception : Pour les éléments quadratiques multi-fissurées, dans l'élément de référence parent, l'arête AB n'est pas nécessairement droite. Le nœud milieu M de l'arête AB n'est pas nécessairement sur le segment $[A,B]$ si le côté AB est le côté d'un sous élément d'intégration généré par la découpe conformément à une première fissure (voir Figure 3.3.8.1-1). Il faut alors prendre en compte l'éventuelle courbure de l'arête; on a ainsi $P \in AB \Leftrightarrow \xi_P = (2t-1)(t-1)\xi_A + t(2t-1)\xi_B + 4t(1-t)\xi_M$ avec $t \in [0,1]$.
Du coup, le problème posé revient à rechercher $t \in [0,1]$ tel que $lsn^h\left(\xi_P = (2t-1)(t-1)\xi_A + t(2t-1)\xi_B + 4t(1-t)\xi_M\right) = 0$.

Pour les modélisations linéaires ou quadratiques mono-fissurées, le point P appartient nécessairement à l'arête $[A,B]$ et donc : $\xi_P = (1-t)\xi_A + t\xi_B$ avec $t \in [0,1]$.
Du coup, le problème posé, revient à rechercher $t \in [0,1]$ tel que $lsn^h\left(\xi_P = (1-t)\xi_A + t\xi_B\right) = 0$.

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

On cherche alors la racine d'une équation polynomiale puisque les fonctions d'interpolation de la level-set sont des polynômes. Cette équation est alors résolue à l'aide de l'algorithme de Newton-Raphson (détaillé ci-dessous).

Comme $f:t \rightarrow lsn^h(t)$ est C^2 et strictement monotone au voisinage du point d'intersection (sinon, la level-set devient parallèle à l'arête AB et ne l'intersecte pas), la convergence de l'algorithme de Newton est **quadratique**, donc optimale.

Algorithme de Newton-Raphson pour le calcul d'un point d'intersection de l'iso-zéro avec l'arête AB :

- Lecture des coordonnées de référence des nœuds sommets A et B et de l'éventuel nœud M milieu : par construction, ces nœuds sont des nœuds de l'élément enfant que l'on découpe et leur coordonnées sont donc connues.

- Initialisation du Newton-Raphson :

  - Pour les modélisations linéaires, on initialise la recherche avec l'approximation linéaire le long de l'arête :

$$t_0 = \frac{lsn(A)}{lsn(A) - lsn(B)}$$

  - Pour les modélisations quadratiques, on initialise la recherche avec la solution exacte du problème le long de l'arête de l'élément enfant. Cette solution est donnée par la résolution d'une équation polynomiale du second degré. Les coefficients de ce polynôme du second degré sont donnés par les valeurs de lsn aux nœuds extrémités et milieu de l'arête du sous élément enfant. Étant donné que la lsn change de signe strictement entre les extrémités de cette arête, il existe une unique solution à l'équation $lsn=0$ sur l'arête. L'équation à résoudre a la forme suivante :

$$lsn(A) + (4*lsn(M) - 3*lsn(A) - lsn(B))*t_0 + 2*(lsn(A) + lsn(B) - 2*lsn(M))*t_0^2 = 0$$

    Parmi les deux solutions éventuelles de cette équation du second degré on ne retient que celle qui satisfait : $t_0 \in [0,1]$. Cette valeur constitue une bonne initialisation et c'est même la solution finale du problème lorsque l'arête de l'élément enfant sur lequel on effectue la recherche coïncide avec une arête de l'élément parent. En effet, la l lsn est alors interpolée uniquement par les nœuds extrémités et milieu de l'arête de l'élément parent. Cette interpolation étant quadratique, l'initialisation fournit la solution exacte. En revanche, lorsque l'arête de l'élément enfant ne coïncide pas avec une arête de l'élément parent (arête interne Figure 3.3.8.1-2) , l'initialisation ne fournit pas la solution exacte du problème car la lsn est interpolée par tous les nœuds de l'élément parent le long de l'arête de l'élément enfant.

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

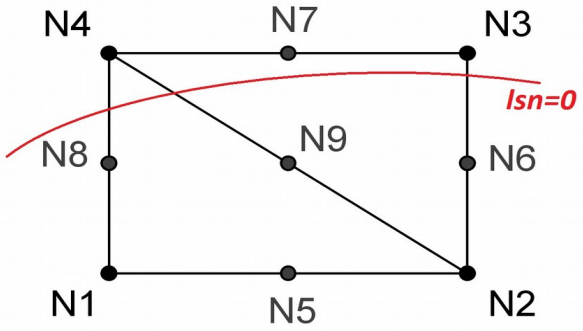


Figure 3.3.8.1-2: Intersection de la courbe *lsn=0* avec une arête de l'élément enfant

- Tant que $n \leqslant itermax$ faire :
  - $\xi_n = \begin{cases} (1-t_n)\xi_A + t_n \xi_B & en\ linéaire\ et\ mono-fissuré \\ (2t_n-1)(t_n-1)\xi_A + t_n(2t_n-1)\xi_B + 4t_n(1-t_n)\xi_M & en\ quadratique\ multi-fissuré \end{cases}$
  - Calcul de la fonction : $lsn^h(\xi_n) = \sum_{1 \leqslant j \leqslant nnop} lsn_j \Phi_j(\xi_n)$
  - Calcul de la dérivée : $dlsn^h(\xi_n)/dt = \sum_{1 \leqslant j \leqslant nnop} lsn_j \nabla \Phi_j(\xi_n) \cdot \vec{v}$
  - Vérification de la pente : si $\left|dlsn^h(\xi_n)/dt\right| < 10^{-16}$ alors sortie avec code erreur.
  - Calcul de l'incrément : $\Delta_n = \dfrac{lsn^h(\xi_n)}{dlsn^h(\xi_n)/dt}$
  - Calcul de la nouvelle position : $t_{n+1} = t_n - \Delta_n$
  - Vérification de l'hypothèse AB : si $t_n \notin [0,1]$ alors sortie avec code erreur.
  - Critère d'arrêt sur l'incrément: si $|\Delta_n| < 10^{-8}$ alors fin de la boucle.
- Calcul des coordonnées de référence du point d'intersection : $\xi_\infty = \begin{cases} (1-t_\infty)\xi_A + t_\infty \xi_B & en\ linéaire\ et\ mono-fissuré \\ (2t_\infty-1)(t_\infty-1)\xi_A + t_\infty(2t_\infty-1)\xi_B + 4t_\infty(1-t_\infty)\xi_M & en\ quadratique\ multi-fissuré \end{cases}$
- Vérification de l'hypothèse $P \in ELREFP$: si $\xi_\infty \notin ELREFP$ alors sortie avec code erreur.
- Calcul des coordonnées réelles par interpolation des coordonnées géométriques : $X(\xi_\infty) = \sum_{1 \leqslant j \leqslant nnop} X^j \Phi_j(\xi_\infty)$

Remarque sur le critère d'arrêt:
On teste l'incrément Δ_n pour évaluer la convergence du Newton, car la solution et l'incrément varient sur l'échelle de l'unité. La valeur de l'incrément par rapport à l'unité est donc un indicateur robuste de

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

la convergence. Si le critère d'arrêt portait sur la valeur de la fonction $lsn^h(\xi_n)$, il faudrait mettre à l'échelle le résultat.
Très concrètement, quand l'incrément tend vers zéro par rapport à l'unité, la solution $t_{n+1}=t_n-\Delta_n$ devient stationnaire par rapport à l'unité. D'un point de vue numérique, cela garantit la convergence sur l'intervalle $[0,1]$, compte-tenu de la précision machine.

A titre d'exemple, considérons maintenant l'interpolation d'une level-set dans l'élément quadratique (QUAD8) du paragraphe §2.2.3. Envisageons un segment AB arbitraire dans l'élément parent (voir Figure 3.3.8.1-3). En traçant l'évolution de la level-set interpolée $f:t\rightarrow lsn^h(t)$ le long du segment $A\text{-}B$ (voir Figure 3.3.8.1-4), on constate que la level-set varie de manière strictement monotone au voisinage du point d'intersection P. Tant que l'on reste au voisinage du point P, l'algorithme de Newton respectera les conditions de convergence optimale.

L'algorithme de Newton proposé ci-dessus, est initialisé par une interpolation linéaire de la level-set le long du segment $A\text{-}B$ Figure 3.3.8.1-5. Le point d'initialisation P_0 est alors suffisamment proche du point de convergence P. L'algorithme de Newton aura un comportement optimal.

En revanche, si l'algorithme avait été initialisé au point A, la proximité d'un point de stationnarité (où la dérivée première de $f:t\rightarrow lsn^h(t)$ s'annule) aurait causé l'échec du Newton. Plusieurs scénarios seraient envisageables :

- soit le Newton ne convergerait pas au bout d'un nombre d'itérations maximal donné,
- soit le Newton sortirait du segment $A\text{-}B$.

La présence du point de stationnarité a une justification géométrique : l'iso-zéro de la level-set est pratiquement parallèle au segment $A\text{-}B$ au voisinage du point A (voir Figure 3.3.8.1-3). Par la suite, la distance entre le segment $A\text{-}B$ et l'iso-zéro diminue progressivement. Par conséquent, au voisinage du point d'intersection la variation de la level-set redevient strictement monotone.

Pour prévenir cette configuration, le Newton est initialisé suffisamment proche du point d'intersection Figure 3.3.8.1-5. De plus, à chaque itération de l'algorithme, on contrôle la persistance de la solution sur le segment $A\text{-}B$ c'est-à-dire $\forall t, t\in[0,1]$. Cela apporte alors une garantie supplémentaire sur la robustesse de la solution calculée, par le Newton.

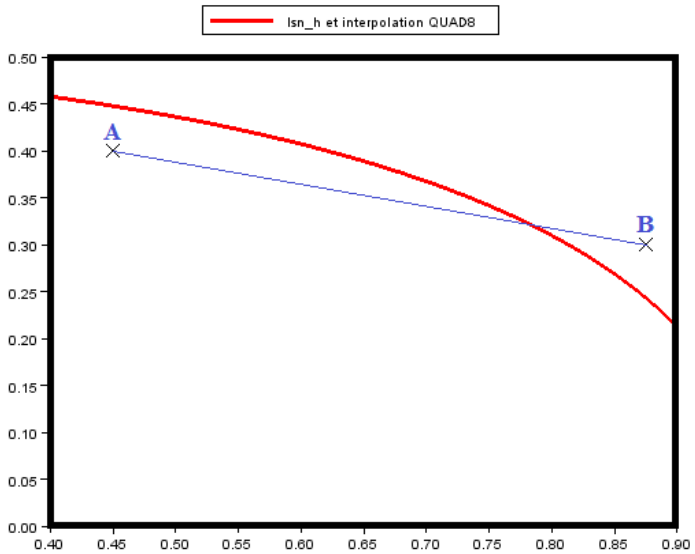


Figure 3.3.8.1-3 : recherche du point d'intersection entre l'iso-zéro de la level-set et un segment A-B (arbitraire) dans un élément QUAD8

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

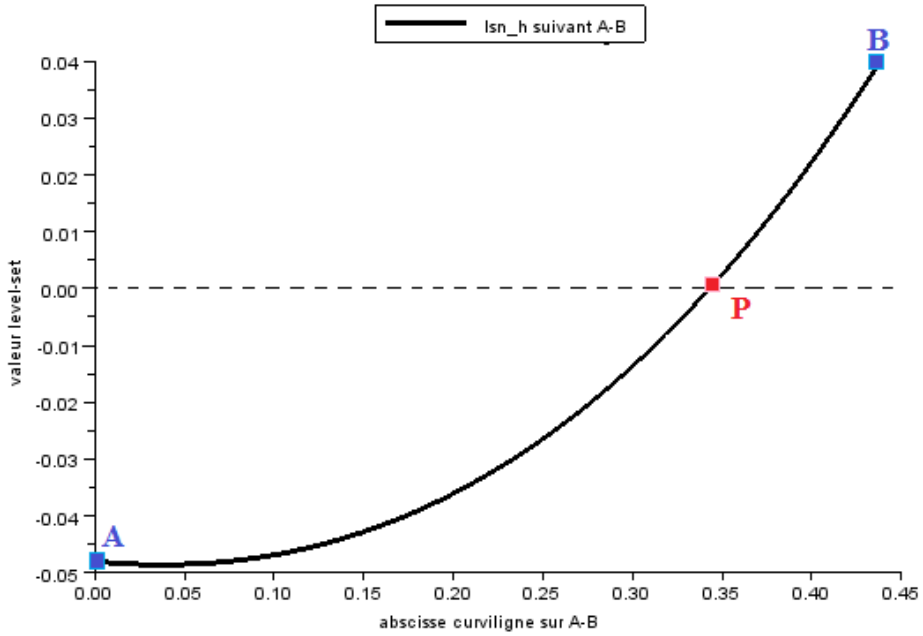


Figure 3.3.8.1-4 : évolution de fonction distance (level-set) le long du segment A-B

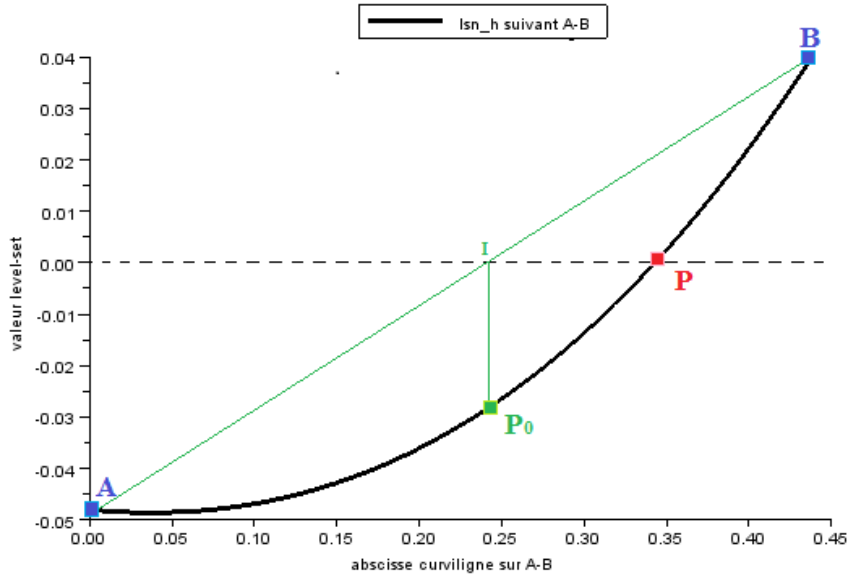


Figure 3.3.8.1-5 : Initialisation de l'algorithme de Newton-Raphson par interpolation linéaire de la level-set

3.3.8.2 Calcul des points milieu

Lors de la découpe d'un tétraèdre quadratique, les nouveaux points milieux peuvent être regroupés suivant 3 catégories Figure 3.3.8.2-1 :

- les points milieux de premier type MP_1 : situés sur les arêtes du tétraèdre (triangle), entre les points d'intersections et les nœuds sommets,

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

- les points milieux de deuxième type MP_2 : situés sur l'iso-zéro, dans la face quadratique (SE3, TRI6 ou QUAD8) délimitée par les points d'intersection. Les points milieux sur les bords de la face et le point milieu au centre de la face (si face quadrangle) sont calculés avec des procédures légèrement différentes,

- les points milieux de troisième type MP_3 : situés sur les faces quadrangles des sous-polyèdres-enfants (pentaèdre ou pyramide), générés par la découpe du tétraèdre.

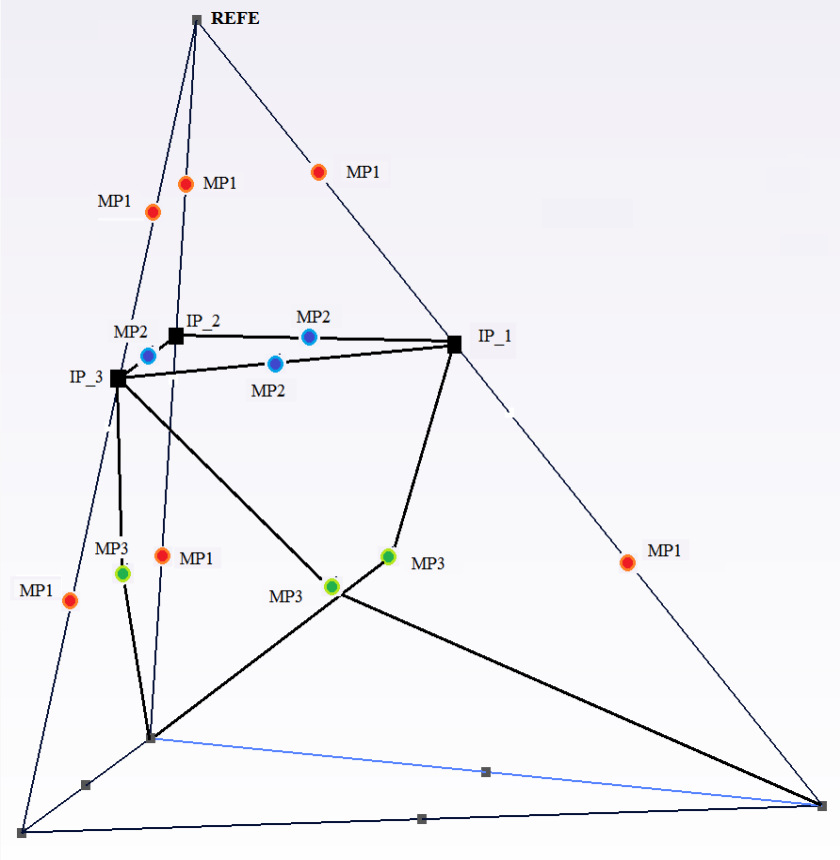


Figure 3.3.8.2-1 : les différents types de nœuds milieux MP_i à calculer lors de la découpe des éléments quadratiques

Définition des points milieux

On propose de définir systématiquement les points milieux dans le repère de référence de l'élément parent. Comme l'élément de référence est à bords droits et à faces planes, cela facilite grandement le calcul des points milieux notamment les nœuds milieux de premier type. Soulignons d'avantage ce point essentiel, les calculs à venir s'appuieront souvent implicitement sur l'hypothèse que les faces des tétraèdres à découper sont planes dans l'élément de référence parent.
Cependant, avec cette définition, il est possible de calculer un nœud milieu qui ne soit pas « au milieu » de l'arête dans le système de coordonnées réelles. Cela n'est pas gênant, à condition de choisir une technique d'intégration adaptée.

Dans l'élément de référence, nous proposons la définition suivante des points milieux :

- pour un point milieu du **premier type** , situé entre le nœud sommet XX et le point d'intersection IP : $\xi_{PM_1} = \xi_{Milieu(XX\text{-}IP)} = \frac{\xi_{XX} + \xi_{IP}}{2}$

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

Comme représenté sur la Figure 3.3.8.2-3 .

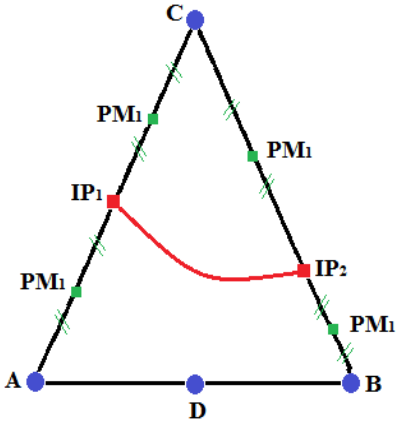


Figure 3.3.8.2-2 : calcul dans le repère de référence, d'un point milieu de deuxième type, situé entre 2 points d'intersection

Exception : Pour les éléments quadratiques multi-fissurées, dans l'élément de référence parent, l'arête intersectée et sur laquelle on souhaite positionner les points milieux n'est pas nécessairement droite. Si l'arête intersectée est le côté d'un sous élément d'intégration généré par la découpe conformément à une première fissure (voir Figure 3.3.8.2-3), il faut prendre en compte l'éventuelle courbure de l'arête pour positionner les points milieux de premier type :

$$\xi_{M_6}=\frac{\xi_{I_4}+\xi_{I_2}}{2}$$

$$M_6=\frac{\xi_{M_6}(\xi_{M_6}-1)}{2}I_2+\frac{\xi_{M_6}(\xi_{M_6}+1)}{2}I_3+(1-\xi_{M_6})(1+\xi_{M_6})I_4$$

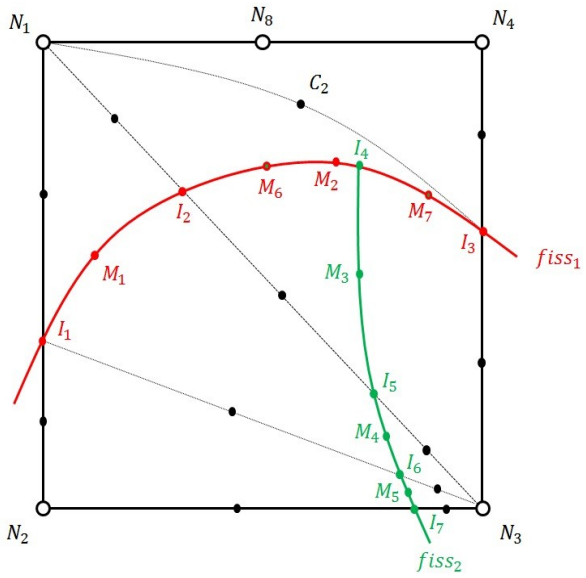


Figure 3.3.8.2-3 : détermination des points milieux M_6 *et* M_7 sur l'arête I_2I_3

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

- pour un point milieu du **deuxième type** , situé entre 2 points d'intersection sur une face du tétraèdre Figure 3.3.8.2-4 , on opère le choix suivant :

$$\xi_{PM_2}=\xi_M+t\,\vec{v}$$
$$lsn^h\left(\xi_{PM_2}\right)=0$$

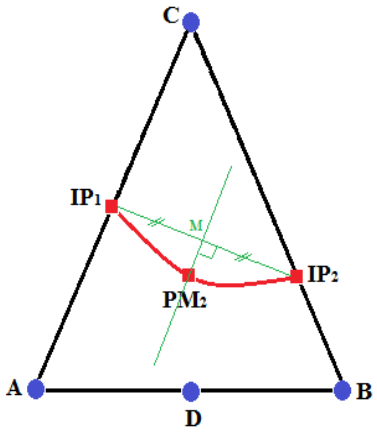


Figure 3.3.8.2-4 : calcul dans le repère de référence, d'un point milieu de deuxième type, situé entre 2 points d'intersection

  - M est le milieu du segment joignant les 2 points d'intersection $\xi_M=\dfrac{\xi_{IP_1}+\xi_{IP_2}}{2}$
  - $\vec{v}$ est le vecteur unitaire normal à $\overrightarrow{IP_1\text{-}IP_2}$, appartenant à la face (A,B,C) (plane dans l'élément de référence). Concrètement pour calculer le vecteur normal $\vec{v}$, on pose les équations suivantes :

$$\left\{\begin{array}{c}\vec{v}\in(A,B,C)\\ \vec{v}\perp\overrightarrow{IP_1\text{-}IP_2}\\ \|\vec{v}\|=1\end{array}\right\}\Rightarrow\left\{\begin{array}{c}\vec{v}=\alpha_1\overrightarrow{AB}+\alpha_2\overrightarrow{AC}\\ \vec{v}\cdot\overrightarrow{IP_1\text{-}IP_2}=0\\ \|\vec{v}\|=1\end{array}\right\}$$

Après résolution :

$$\vec{v}=\pm\alpha_1\left(k_2\overrightarrow{AB}-k_1\overrightarrow{AC}\right)\text{ avec }\begin{array}{c}\alpha_1=1/\sqrt{n_1^2k_2^2+n_2^2k_1^2-2kk_1k_2}\\ n_1=\|\overrightarrow{AB}\|\text{ et }n_2=\|\overrightarrow{AC}\|\\ k_1=\overrightarrow{AB}\cdot\overrightarrow{IP_1\text{-}IP_2}\\ k_2=\overrightarrow{AC}\cdot\overrightarrow{IP_1\text{-}IP_2}\\ k=\overrightarrow{AB}\cdot\overrightarrow{AC}\end{array}$$

Remarques :

1. *Si* $k_2=0$ *c'est-à-dire* $\overrightarrow{AC}\perp\overrightarrow{IP_1\text{-}IP_2}$ *, alors,* $\vec{v}=\pm\dfrac{\overrightarrow{AC}}{\|\overrightarrow{AC}\|}$

2. *Il y a 2 vecteurs directeurs unitaires possibles pour une droite donnée, ce qui explique les 2 solutions calculées pour le vecteur* $\vec{v}$ *. On choisit arbitrairement l'une des 2 solutions. Ce choix n'a aucune influence sur la convergence de l'algorithme de Newton.*

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

- pour un point milieu du **deuxième type** , situé au centre d'une face quadrangle, on opère le choix suivant :

$$\xi_{PM_2}=\xi_M+t\,\vec{v}$$
$$lsn^h\left(\xi_{PM_2}\right)=0$$

  - M est le point milieu de l'arête $\left[IP_1\text{-}IP_4\right]$ c'est-à-dire $\xi_M=\frac{\xi_{IP1}+\xi_{IP4}}{2}$. Si l'iso-zéro est plane la découpe génère alors des sous-tétraèdres à bords droits.

  - $\vec{v}$ est le gradient de la lsn au point M : $\vec{v}=\overrightarrow{grad_{lsn}}(M)$. C'est la « meilleure » direction de recherche car le gradient de la lsn est par définition orthogonal à la surface $lsn=0$ que l'on cherche à atteindre.

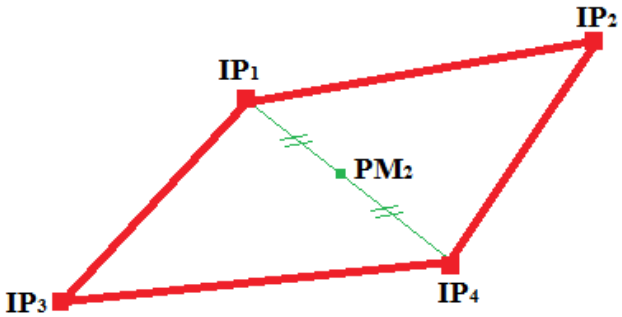


Figure 3.3.8.2-5 : calcul dans le repère de référence, d'un point milieu de deuxième type, situé sur une face quadrangle

  - **Calcul des points milieux de deuxième type :**
    Les points milieux de deuxième type vérifient l'équation d'intersection d'un droite avec l'iso-zéro suivante :

$$\xi_{PM_2}=\xi_M+t\,\vec{v}$$
$$lsn^h\left(\xi_{PM_2}\right)=0$$

En fonction de la localisation du point milieu (soit situé entre 2 points d'intersection, soit situé à l'intérieur de la face quadrangle sur l'iso-zéro), nous avons explicité les coordonnées exactes de ξ_M et $\vec{v}$ dans l'élément de référence.

On résout la recherche du zéro de l'équation $lsn^h\left(\xi_M+t\,\vec{v}\right)=0$ par un Newton comme au paragraphe §2.2.3. Afin de s'assurer que l'algorithme de Newton « reste » respectivement dans la face TRIA pour un point milieu de deuxième type situé entre deux points d'intersection et dans le sous élément TETRA pour un point milieu de deuxième type situé au centre d'une face quadrangle, on renforce l'algorithme de Newton par la méthode de Dekker. Si une itération de l'algorithme de Newton nous envoie en dehors de la zone de recherche autorisée, une dichotomie sera préférée pour se rapprocher de la solution. En effet, il arrive que l'équation $\begin{matrix}\xi_{PM_2}=\xi_M+t\,\vec{v}\\ lsn^h\left(\xi_{PM_2}\right)=0\end{matrix}$ possède plusieurs solutions, mais il n'y en a qu'une seule qui est dans la face TRIA de recherche ou dans le sous élément TETRA de recherche. Cette solution peut s'avérer difficile à capter par l'algorithme de Newton, surtout lorsque la courbe $lsn=0$ a une courbure importante ou « rase » une arête de la face TRIA ou une face du sous élément TETRA (voir Figure 3.3.8.2-6 et Figure 3.3.8.2-7).

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

Figure 3.3.8.2-6 : existence de solutions rapprochées pour le point milieu de type 2 dans une face TRIA

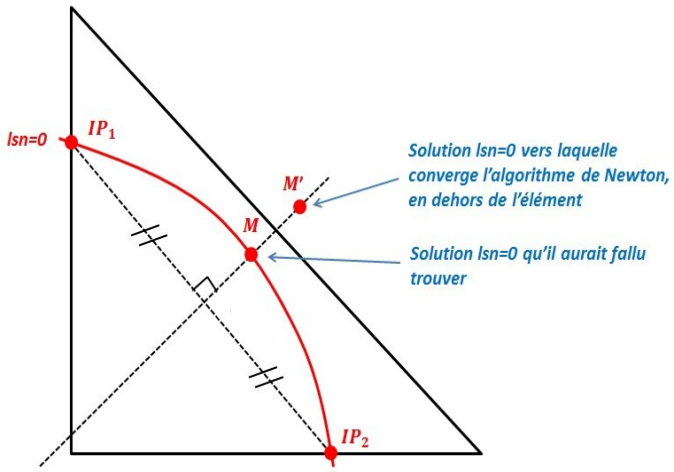
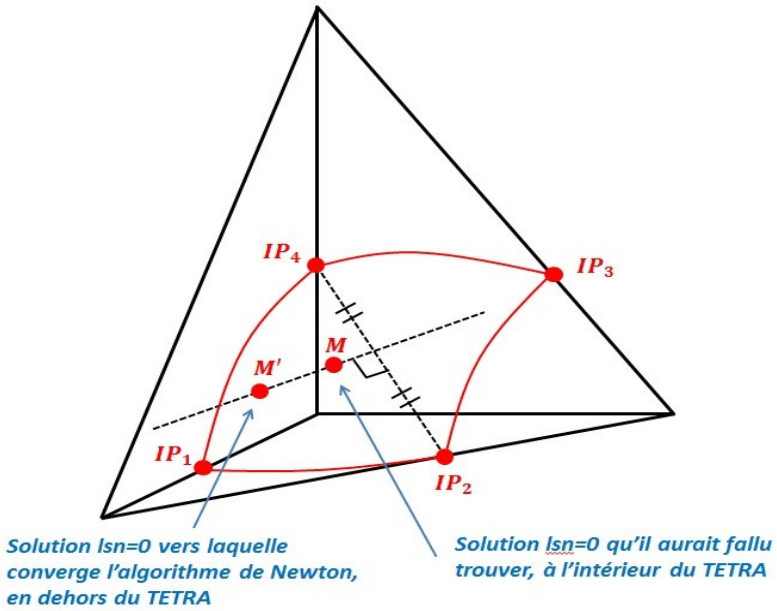


Figure 3.3.8.2-7 : existence de solutions rapprochées pour le point milieu de type 2 situé sur une face quadrangle dans un sous élément TETRA



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

Figure 3.3.8.2-8 : limites de « l'espace de recherche » dans la face TRIA pour un point milieu de deuxième type situé entre deux points d'intersection

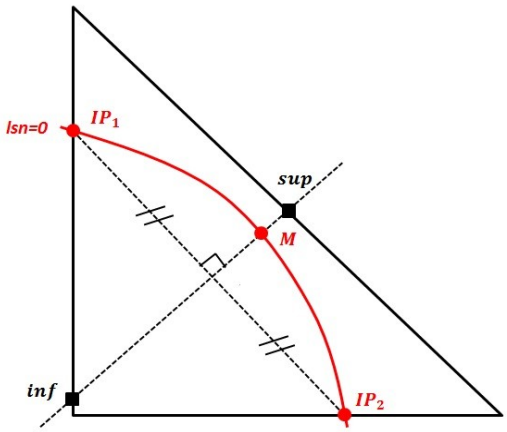
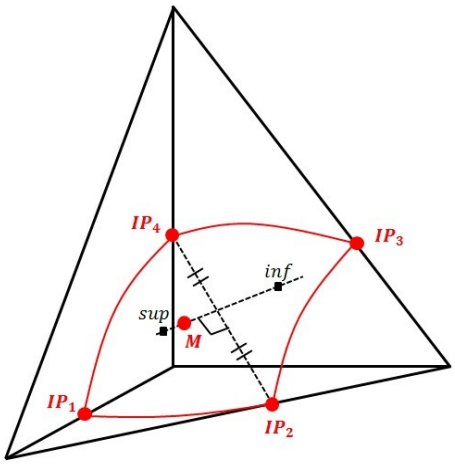


Figure 3.3.8.2-9 : limites de « l'espace de recherche » dans le sous élément TETRA pour un point milieu de deuxième type situé au centre d'une face quadrangle



Algorithme de Newton-Raphson pour le calcul d'un point milieu situé sur l'isozéro de la level-set :

- Définition de ξ_M point de départ de la recherche et de $\vec{v}$ direction de la recherche.
- Initialisation du Newton -Raphson : $t_0=0$
- Détermination des bornes inférieures et supérieures t_{min} et t_{max} de l'espace de recherche (voir Figure 3.3.8.2-8 et Figure 3.3.8.2-9)
- Tant que $n \leqslant itermax$ faire :
  - $\xi_n=t_n \vec{v}+\xi_M$

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

- Calcul de la fonction : $lsn^h(\xi_n)=\sum_{1\leqslant j\leqslant nnop} lsn_j\,\Phi_j(\xi_n)$
- Calcul de la dérivée : $dlsn^h(\xi_n)/dt=\sum_{1\leqslant j\leqslant nnop} lsn_j\,\nabla\Phi_j(\xi_n)\cdot\vec{v}$
- Vérification de la pente : si $\left|dlsn^h(\xi_n)/dt\right|<10^{-16}$ alors sortie avec code erreur.
- Calcul de l'incrément : $\Delta_n=\dfrac{lsn^h(\xi_n)}{dlsn^h(\xi_n)/dt}$
- Calcul de la nouvelle position : $t_{n+1}=t_n-\Delta_n$
- Si $t_{n+1}\in[t_{min},t_{max}]$, alors on continue, sinon :
  - Si $t_{n+1}>t_{max}$, alors $t_{n+1}=\dfrac{t_n+t_{max}}{2}$
  - Si $t_{n+1}<t_{min}$, alors $t_{n+1}=\dfrac{t_n+t_{min}}{2}$
- Critère d'arrêt sur l'incrément: si $|\Delta_n|<10^{-8}$ alors fin de la boucle.

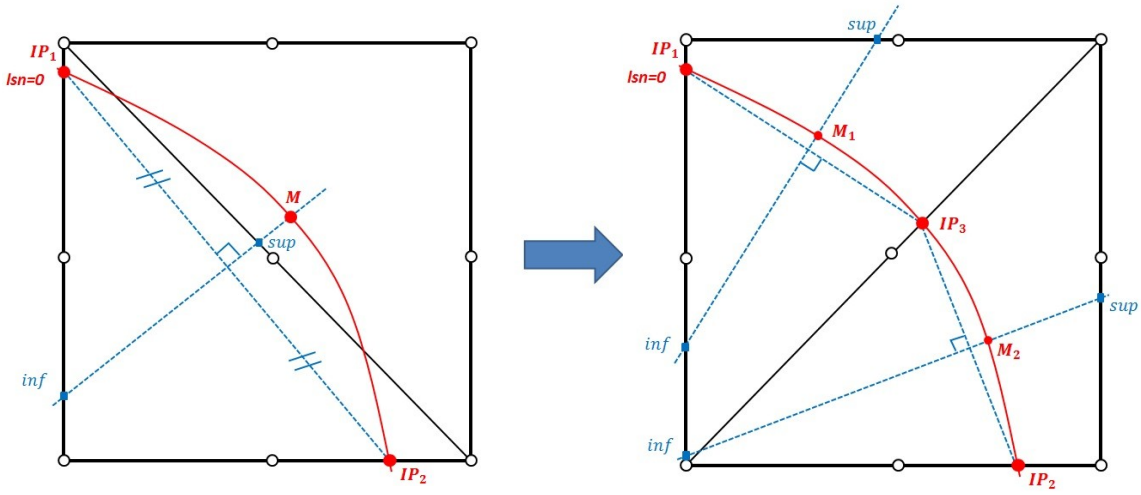
- Calcul des coordonnées de référence du point d'intersection : $\xi_\infty=t_\infty\vec{v}+M$
- Vérification de l'hypothèse $\xi_\infty\in ELREFP$: si $\xi_\infty\notin ELREFP$ alors sortie avec code erreur.
- Calcul des coordonnées réelles par interpolation des coordonnées géométriques :
$$X(\xi_\infty)=\sum_{1\leqslant j\leqslant nnop} X^j\Phi_j(\xi_\infty)$$

Pour les éléments non simpliciaux, il existe une situation pour lesquelles la recherche du point milieu de type 2 est vaine dans l'espace de recherche autorisé (entre les bornes inf et sup). Comme indiqué au paragraphe § 3.3.1 , la découpe en sous éléments se base sur une réduction des éléments non simpliciaux en éléments simpliciaux (TRIA en 2D et TETRA en 3D). Or cette réduction peut amener des situations qui rendent la recherche des points milieux de type 2 impossible dans dans la subdivision des éléments non simpliciaux. Un exemple est présenté sur la figure Figure 3.3.8.2-11 à gauche. On recherche le point milieu de type 2 entre les points d'intersection IP_1 et IP_2 . Or dans l'intervalle de recherche autorisé (entre les bornes inf et sup), le champ de lsn est de signe constant. L'algorithme de Newton-Raphson est donc mis en échec. L'arête diagonale interne est en effet intersectée par la courbe $lsn=0$ alors que cette dernière était détectée comme non coupée car la valeur de la lsn en chacun de ses 3 nœuds est strictement de même signe. Mais à la différence d'une arête de l'élément parent, le champ lsn le long de cette arête interne dépend de la valeur de la lsn en chacun des 8 nœuds de l'élément parent QUAD8, et non pas seulement de la valeur de la lsn en ses 3 nœuds. Voilà pourquoi le champ de lsn change de signe le long de cette arête interne. Cette difficulté peut être contournée en choisissant une autre bijection pour la division du quadrangle en deux triangles (confer Figure 3.3-1). Sur la Figure 3.3.8.2-11 à droite, on utilise l'autre bijection pour la découpe du quadrangle en deux triangles de manière à éviter un changement de signe du champ lsn le long de l'arête interne. On détermine alors les points milieux de deuxième type M_1 et M_2 dans les intervalles autorisés. Donc chaque fois que la recherche d'un point milieu de deuxième type dans l'intervalle $[inf,sup]$ est mise en échec au sein d'un élément simplicial, on relance la procédure de découpe de l'élément en choisissant une bijection différente pour la division en sous cellule triangulaires (ou tétraédriques en 3D). Dans le cas où toutes les bijections possibles n'auraient

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

pas permis d'apporter une solution à ce problème, on effectue alors une approximation linéaire de la courbe $lsn=0$, où le point milieu est choisi comme étant le milieu du segment $[IP_1\,IP_2]$ dans l'espace de référence.

Figure 3.3.8.2-10 : cas d'avortement de la recherche d'un point milieu de type 2 pour les éléments non simpliciaux.



Ces cas de figure sont rares et ne surviennent que lorsque la courbure de l'interface est importante au regard de la taille des mailles. En raffinant suffisamment le maillage, on est assuré de ne pas se heurter à ce problème.

- pour un point milieu du **troisième type** , situé sur une face quadrangle d'un polyèdre-enfant (pyramide ou pentaèdre), l'arête virtuelle de découpe relie le premier point d'intersection IP_1 de la face et un nœud sommet du tétraèdre B (voir Figure 3.3.8.2-11). Concernant le positionnement du point milieu sur la face quadrangle:
  - si la level-set est droite, on opère le choix suivant : $\xi_{PM_3}=\xi_{Milieu(IP_1\text{-}B)}=\frac{\xi_{IP_1}+\xi_B}{2}$ pour que la découpe génère des sous-tétraèdres à bords droits.
  - si la level-set est courbe , on opère le choix suivant : $\xi_{PM_3}=\xi_{Milieu(PM_2\text{-}D)}=\frac{\xi_{PM_2}+\xi_D}{2}$ pour que le point milieu calculé ne sorte pas de la face quadrangle Figure 3.3.8.2-11 .

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

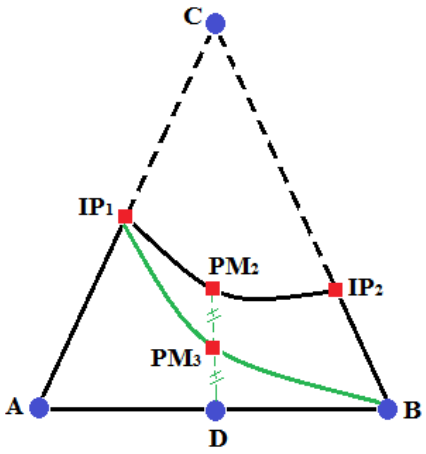
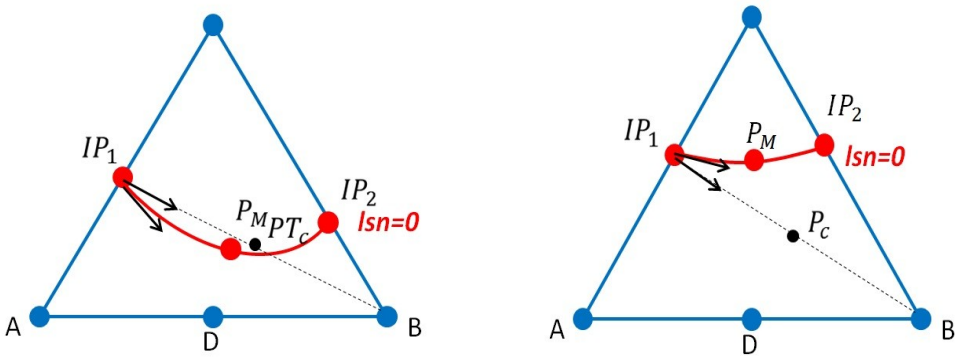


Figure 3.3.8.2-11 : calcul dans le repère de référence, d'un point milieu de troisième type pour une iso-zéro de forme courbe

Remarque sur le critère de courbure

Pour juger de la courbure de la lsn, on effectue la démarche suivante. On considère d'abord le point milieu PM_3 *candidat tel que* $\xi_{PM_3}=\xi_{Milieu(IP_1\text{-}B)}=\frac{\xi_{IP_1}+\xi_B}{2}$ *. On calcule alors le sinus de l'angle entre les vecteurs* $\overrightarrow{IP_1 PM_2}$ *et* $\overrightarrow{IP_1 PM_3}$ *, le plan étant orienté par rapport à la normale sortante venant vers le lecteur. S'il est négatif, ce point milieu donnera naissance à des éléments distordus Figure 3.3.8.2-12, sinon les deux TRIA résultants seront sains. En pratique, on considère que la lsn est trop courbe lorsque le sinus est inférieur à 0.1. On choisit alors* PM_3 *tel que .*

Figure 3.3.8.2-12 : critère pour déterminer si la lsn est jugée courbe : situation « courbe » à gauche et « saine » à droite



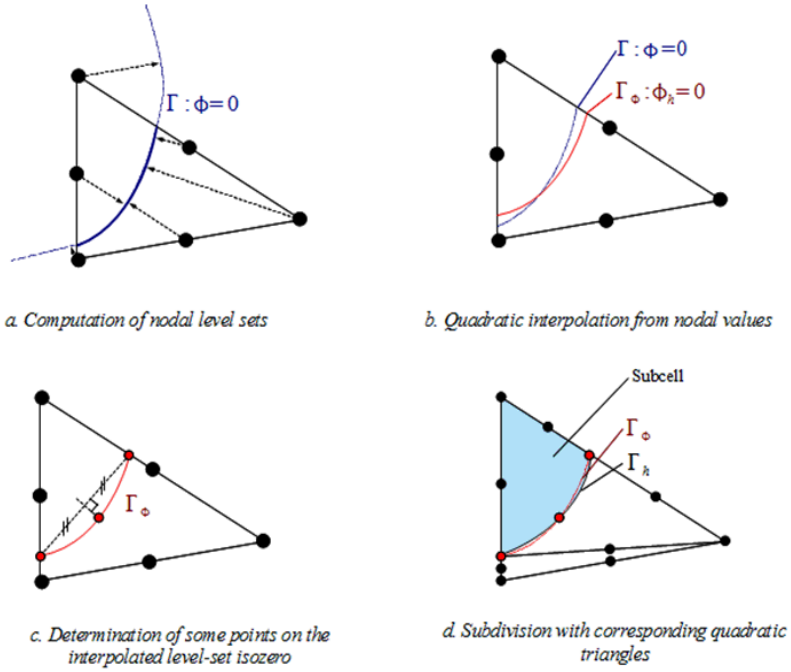
Sous-découpage et erreur de discrétisation

Dans la procédure de découpage expliquée au paragraphe §3.3.8, la levet-set est interpolée systématiquement dans l'élément de référence parent. Par conséquent, les points d'intersection et points milieux positionnés sur l'iso-zéro, vérifient rigoureusement l'équation $lsn^h(\xi^{ELREFP})=0$. En

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

clair, l'algorithme de positionnement des points sur la surface de l'iso-zéro n'introduit pas d'erreur supplémentaire par rapport à l'erreur d'interpolation de la level-set.

En revanche, la forme géométrique de la level-set est approximée en reliant les points calculés pour former la frontière des sous-domaines d'intégration (voir Figure 3.3.8.2-13, où la fonction $\Phi^h = lsn^h(\xi^{ELREFP}) = 0$ est une conique alors que le bord de la sous-cellule quadratique coïncidant au mieux avec cette conique est parabolique).



a. Computation of nodal level sets

b. Quadratic interpolation from nodal values

c. Determination of some points on the interpolated level-set isozero

d. Subdivision with corresponding quadratic triangles

Figure 3.3.8.2-13 : approximation de l'iso-zéro

3.4 Récupération des facettes de contact

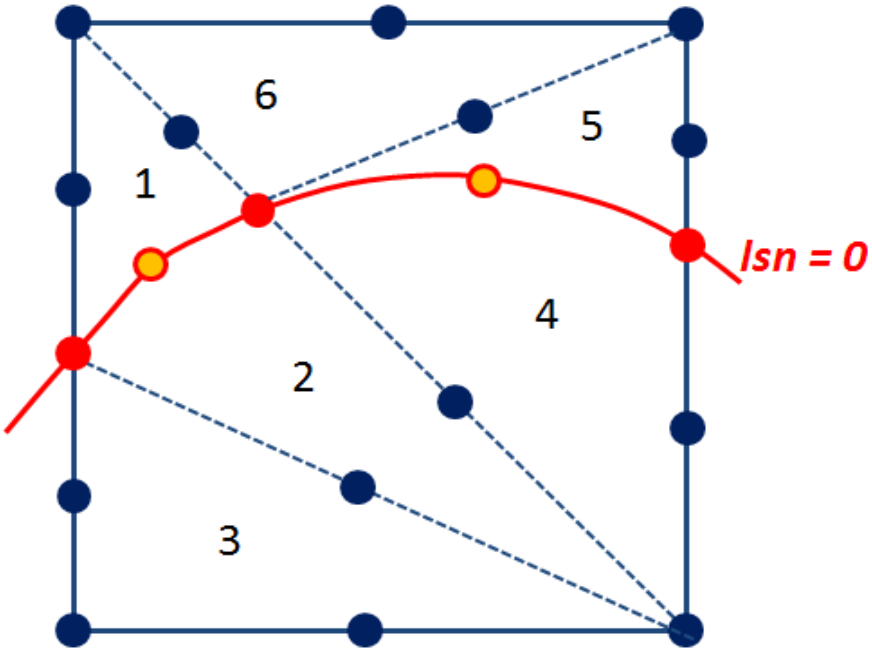
3.4.1 Éléments XH

La récupération des facettes de contact est utile à la fois pour le contact mais aussi pour appliquer une pression mécanique dans les lèvres de fissures. Pour obtenir ces facettes, on s'appuie sur le travail de découpage en sous éléments décrit en §3.3. En effet, la topologie des sous éléments d'intégration contient déjà toutes les informations sur les facettes de contact, qui ne sont autres que les côtés (en 2D) ou les faces (en 3D) des sous éléments d'intégration qui vérifient $lsn = 0$. Il suffit donc simplement de parcourir les côtés ou les faces des sous éléments d'intégration et de sélectionner ceux qui sont confondus avec la discontinuité (voir Figure 3.4.1-1). Or les nœuds sommets des sous éléments d'intégration qui sont des points d'intersection sont repérés par leur connectivité comprise entre 1000 et 1999. Dans le cas d'un point d'intersection confondu avec un nœud de l'élément parent, on aura directement $lsn = 0$ pour ce nœud.

Les facettes que l'on récupère sont des `SEG2` en 2D linéaire, des `SEG3` en 2D quadratique, des `TRIA3` en 3D linéaire et des `TRIA6` en 3D quadratique. Afin de ne récupérer les facettes qu'une et une seule fois, on n'effectue la recherche que sur les côtés ou les faces des sous éléments d'intégration tels que $lsn < 0$.

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

Figure 3.4.1-1 : sous éléments d'intégration et facettes de contact dans un QUAD8



Dans l'exemple ci-dessus, le QUAD8 est divisé en 6 TRIA6. On effectue la recherche des facettes de contact sur les sous éléments 2, 3 et 4 qui vérifient $lsn<0$. On récupère alors deux SEG3 (en rouge).

Algorithme de récupération des facettes de contact pour les éléments XH:

- Boucle sur les nse sous éléments de l'élément parent XH
  - Si le sous élément vérifie $lsn<0$, alors
    - Boucle sur les côtés (en 2D) ou les faces (en 3D) du sous-élément
      - Boucle sur les nœuds sommets du côté ou de la face courante

        - si la connectivité du nœud est comprise entre 1000 et 1999, c'est un point d'intersection avec la discontinuité, on le sélectionne.

        - si la connectivité du nœud sommet est strictement inférieure à 1000, alors c'est un nœud sommet de l'élément parent, on le sélectionne uniquement s'il vérifie $lsn=0$.
      - Fin de la boucle sur les nœuds sommets de la face courante
      - Si tous les nœuds sommets du côté ou de la face courante ont été sélectionnés, il s'agit d'une facette de contact, dont on stocke la connectivité et les coordonnées de ses nœuds
    - Fin de la boucle sur les côtés (en 2D) ou les faces (en 3D) du sous-élément

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

- Fin de la condition

- Fin de la boucle sur les nse sous éléments de l'élément parent XH

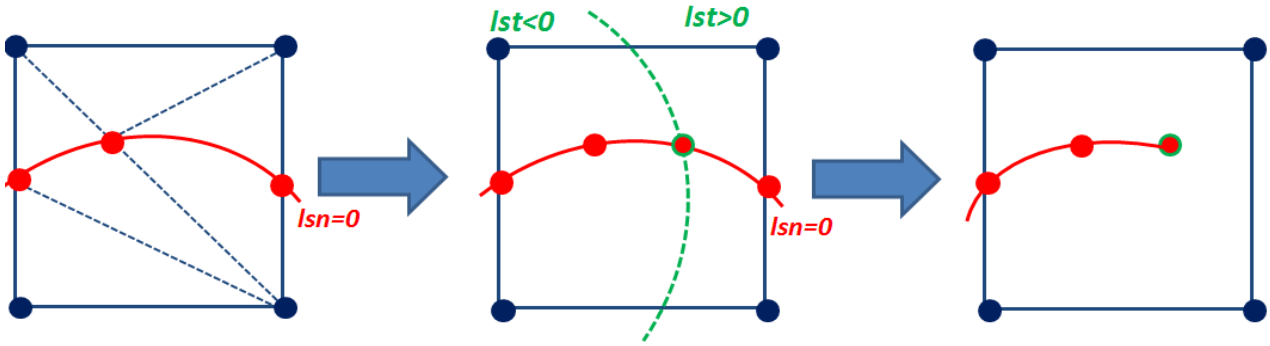
3.4.2 Éléments XHT et XT

Lorsque l'élément est XHT ou XT, un travail supplémentaire est nécessaire par rapport aux éléments XH. En effet, la découpe en sous-éléments est effectuée conformément à la lsn, mais pas conformément à la lst. Les facettes de contact que l'on récupère vérifient donc bien $lsn=0$, mais pas nécessairement $lst<0$, car les éléments XHT et XT sont susceptibles de contenir le fond de fissure. Un redécoupage conforme à la lst est alors nécessaire.

Une fois obtenues les facettes de contact telles que $lsn=0$, on les redécoupe si nécessaire selon la courbe $lst=0$ (voir Figure 3.4.2-1)

Pour ce faire, on utilise exactement la même démarche que lors de la découpe des éléments principaux en sous éléments d'intégration (découpe conforme à la surface $lsn=0$). Dans l'exemple ci-dessous, le SEG2 de droite est découpé suivant la courbe $lst=0$.

Figure 3.4.2-1 : redécoupage des facettes au niveau du fond de fissure



3.4.3 Les éléments multi-Heaviside

Pour les éléments multi-fissurés, la procédure est la même. En effet, la découpe en sous éléments est effectuée de manière séquentielle, fissure après fissure, de manière à ce que les sous-éléments d'intégration finaux soient conformes à chacune des fissures (voir Figure 3.4.3-1). On utilise alors, pour chaque fissure, le même procédé que pour les éléments XH. Cependant, pour les fissures sur lesquelles une autre fissure se branche, on effectue pas nécessairement la recherche des facettes de contact dans les éléments tels que $lsn<0$. Afin de s'assurer que les facettes de contact soient conformes à la jonction, on effectue la recherche des facettes de contact du côté où la fissure se branche (voir Figure 3.4.3-1).

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

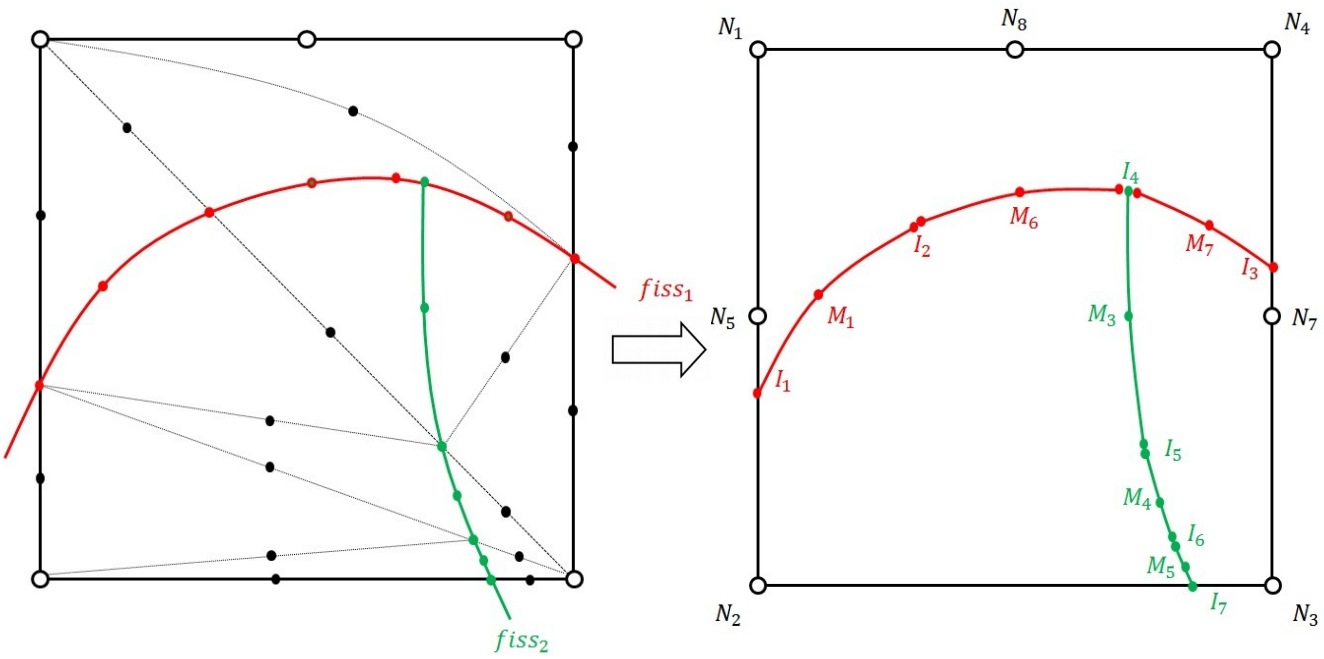


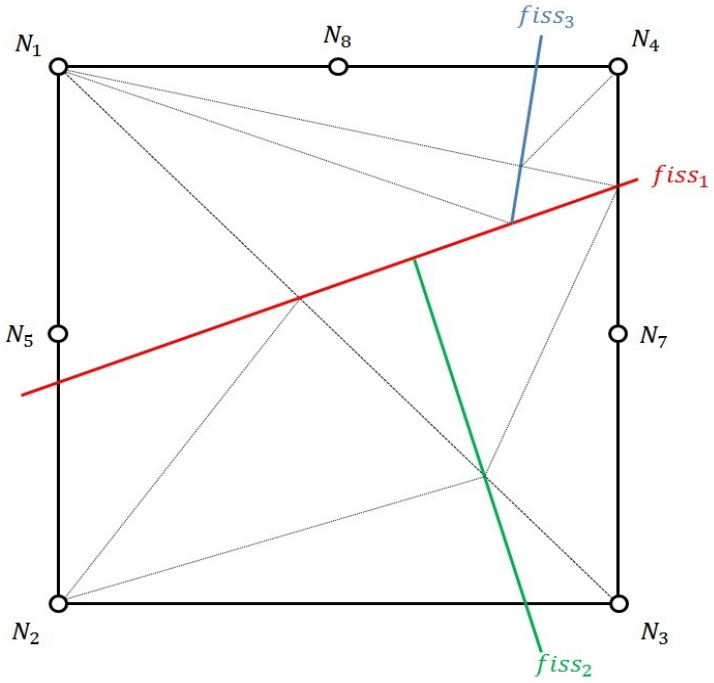
Figure 3.4.3-1 : récupération des facettes de contact pour un élément multi-fissurés. Les facette de la fissure 1 sont récupérées du côté ou la fissure 2 se branche.

Lorsque plusieurs fissures se branchent sur une même fissure au sein d'un élément, si tous les branchements ne se font pas du même côté, il n'existe pas en général de facettes de contact conformes à chaque fissure(voir Figure 3.4.3-2). L'algorithme de découpage émet alors un message d'alarme qui invite l'utilisateur à raffiner le maillage de manière à ne plus avoir de jonctions multiples au sein d'un même élément.

Il n'y a pas de redécoupage nécessaire car les éléments multi-fissurés ne peuvent pas contenir de fond de fissure, ce sont des éléments d'interface uniquement.

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

Figure 3.4.3-2 : récupération des facettes de contact impossible pour un élément multi-fissuré



dans lequel deux fissures se branchent de part et d'autre d'une première fissure.

Algorithme de récupération des facettes de contact pour les éléments multi-fissurés:

- Boucle sur les $nfiss$ fissures qui traversent l'élément

  $signe^{ref}=-1$

  Si une ou plusieurs fissures se branchent sur $ifiss$, on détermine le signe correspondant au côté où les fissures se branchent $signe^{ref}=signe(côté)$. Si les fissures ne se branchent pas toutes du même côté, on émet le message d'alarme suivant :

  « Plusieurs fissures sont branchées de part et d'autre d'une même fissure au sein du même élément. La récupération de facettes de contact conformes à chaque jonction risque d'échouer. Raffinez le maillage afin de que cette situation n'arrive plus »

  - Boucle sur les nse sous éléments de l'élément parent XH
    - Si le sous élément vérifie $signe^{sous\,élément}(ifiss)=signe^{ref}$, alors
      - Boucle sur les côtés (en 2D) ou les faces (en 3D) du sous éléments
        - Boucle sur les nœuds sommets du côté ou de la face courante

          - si la connectivité du nœud est comprise entre 1000 et 1999, c'est un point d'intersection avec l'une des fissures, on le présélectionne. On le sélectionne définitivement si $|lsn^{ifiss}|<loncar*10^{-8}$ avec $loncar$ la longueur caractéristique de l'élément parent. Cette partie de la programmation évoluera de façon à se débarrasser de ce critère. En effet, deux nœuds d'un même côté, de connectivités comprises entre

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

1000 et 1999 réalisent le minimum d'une seule level set qu'il est donc facile d'identifier;

- si la connectivité du nœud sommet est strictement inférieure à 1000, alors c'est un nœud sommet de l'élément parent, on le sélectionne uniquement s'il vérifie $lsn^{ifiss}=0$.

- Fin de la boucle sur les nœuds sommets de la face courante

- Si tous les nœuds sommets du côté ou de la face courante ont été sélectionnés, il s'agit d'une facette de contact pour la fissure $ifiss$, dont on stocke la connectivité et les coordonnées de ses nœuds

- Fin de la boucle sur les côtés (en 2D) ou les faces (en 3D) du sous éléments

- Fin de la condition

- Fin de la boucle sur les nse sous-éléments de l'élément parent XH

Fin de la boucle sur les $nfiss$ fissures qui traversent l'élément

3.4.4 Nombre maximum de facettes récupérées

3.4.4.1 Cas 2D

Dans le cas 2D, on peut avoir au maximum 3 facettes de contact dans un quadrangle (voir Figure 3.4.4.1-1).

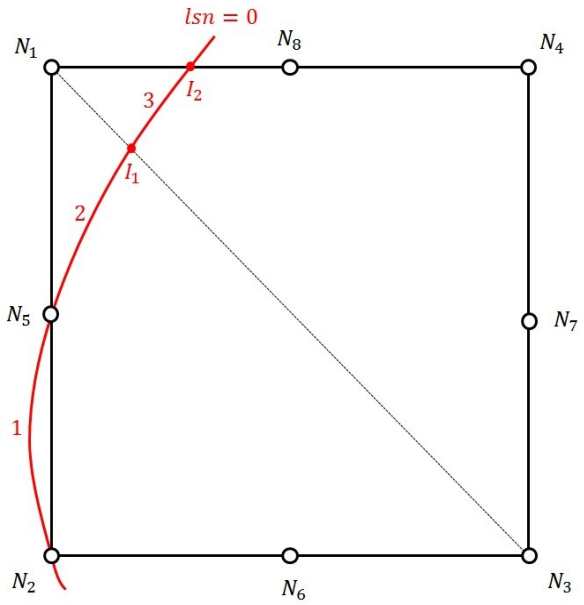


Figure 3.4.4.1-1 : un quadrangle à 8 nœuds et ses 3 facettes de contact

3.4.4.2 Cas 3D

Dans le cas 3D, pour un hexaèdre, on peut majorer le nombre maximum de facettes de contact pour une seule fissure par 18. On est en effet susceptible d'atteindre ce chiffre pour un élément qui contient le fond de ffissure. Lors du redécoupage des facettes triangulaires conformément à la lst , certaine facettes donnent naissance à deux facettes (voir Figure 3.4.4.2-1) , ce qui explique que l'on puisse avoir au final près de 18 facettes.

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

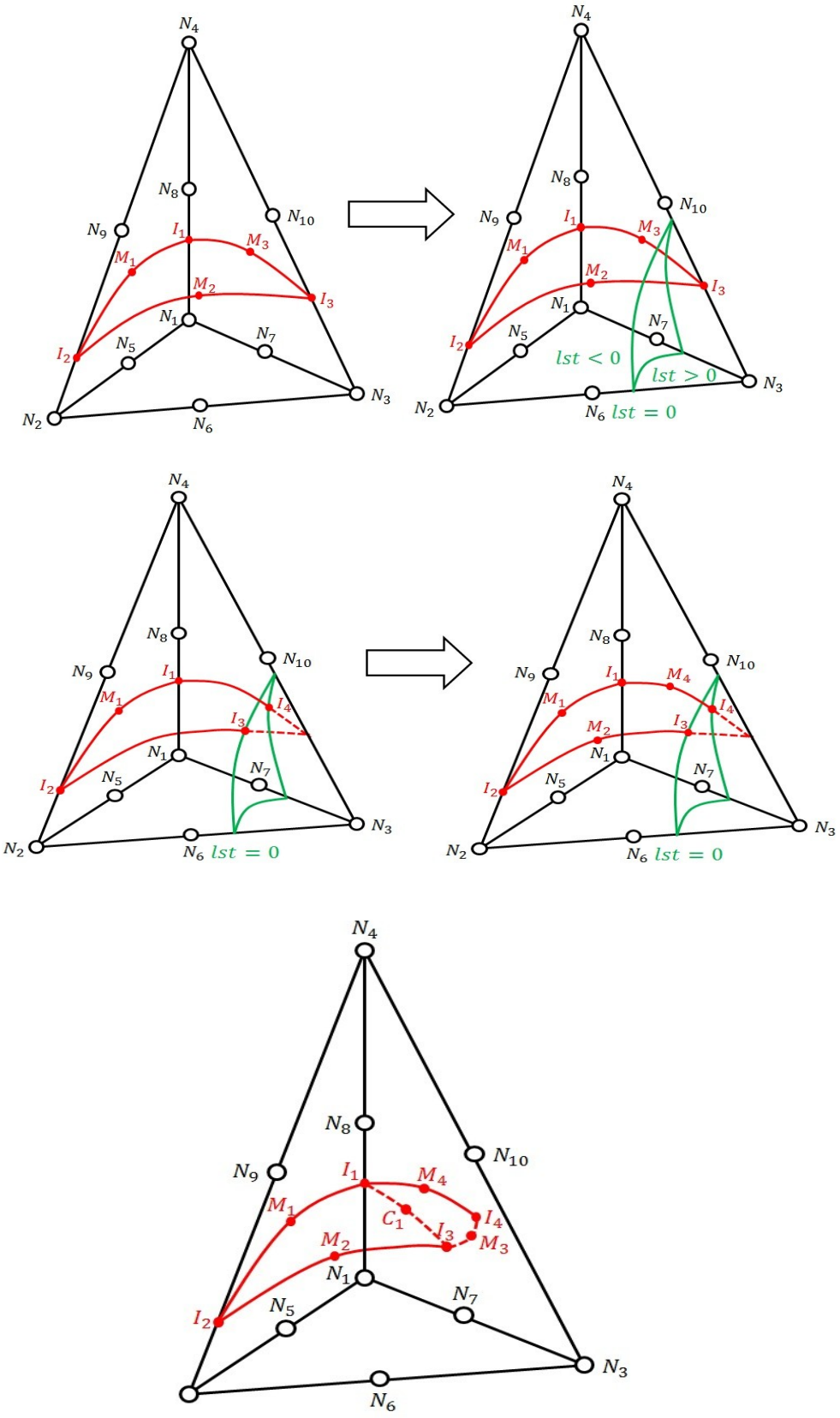


Figure 3.4.4.2-1 : redécoupage d'une facette triangulaire au niveau du fond de fissure

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

3.4.4.3 Cas multi-Heaviside

Dans le cas multi-fissuré, le nombre maximum de facettes de contact par fissure est plus élevé que le nombre maximum de facettes de contact pour un élément traversé par une seule fissure. En effet, le découpage séquentiel génère plus de sous éléments pour chaque nouvelle fissure et donc plus de facettes de contact car ces dernières s'appuient sur les sous éléments. En 3D, pour un hexaèdre, on fixe le nombre maximal de facettes de contact par fissures à 30. Un élément 3D traversées par 4 fissures peut donc avoir un total de $30*4=120$ facettes. En 2D, on fixe le nombre maximum de facettes de contact par fissures à 7. Un élément 2D traversé par 4 fissures peut donc avoir un total de $4*7=28$ facettes.

3.5 Intégration de la rigidité

3.5.1 Intégrande du terme de rigidité mécanique

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

$$\left\{u^h\right\}=\left\{\varphi_i \quad \varphi_j H_j \quad \varphi_k F^1 \quad \varphi_k F^2 \quad \varphi_k F^3 \quad \varphi_k F^4\right\}\cdot\begin{Bmatrix} a_i \\ b_j \\ c_k^1 \\ c_k^2 \\ c_k^3 \\ c_k^4 \end{Bmatrix}$$

Soit encore :

$$\left\{u^h\right\}=\left\{N\right\}\cdot\left\{u\right\}$$

où $\left\{N\right\}$ est le vecteur de la base enrichie des fonctions de forme et $\left\{u\right\}$ le vecteur des ddls nodaux de déplacement.
Les déformation s'écrivent :

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

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

$$\left[B\right]=\left\{\left(\varphi\right)_i' \quad \left(\varphi_j\right)' H_j \quad \left(\varphi_k\right)' F^\alpha+\varphi_k\left(F^\alpha\right)'\right\},\quad \alpha=1,4$$

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) :

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

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

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

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 $H_j(x)$ y est constante, et les fonctions Φ_i', F^α et $\left(F^\alpha\right)'$ y sont continues.

3.5.2 Intégrande du terme de rigidité géométrique

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

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 :

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

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 :

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

et en utilisant le fait que :

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

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

et :

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

la variation du travail virtuel devient:

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

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

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

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

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

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

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

3.5.3 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) :

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

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

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

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

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

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

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

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

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 :

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

3.5.4 Schémas d'intégration

3.5.4.1 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-Legrendre 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 [bib33]. 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 3.5.4.1-1). 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 [bib34] pour déterminer le schéma adapté permettant une intégration exacte[3] 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 3.5.4.1-1 : Schémas d'intégration selon le type de l'élément parent

3.5.4.2 Intégration des termes singuliers

3 Une intégration préservant l'ordre d'approximation élément fini serait en fait suffisante

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

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 [bib28]. Une dizaine de points de Gauss dans chaque direction suffit à atteindre les limites de la précision numérique d'un calculateur. 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 [bib28], l'effort d'implémentation à fournir pour les structures 3D nous paraît considérable [bib35]. 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.

3.6 Intégration des seconds membres surfaciques

3.6.1 Intégrande du second membre des efforts surfaciques

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

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

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 [§3.3.7].

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 $n = \dfrac{\overrightarrow{AB} \wedge \overrightarrow{AC}}{\|\overrightarrow{AB} \wedge \overrightarrow{AC}\|}$. La base locale s'écrit :

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

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.

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

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

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

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

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.

Pour les éléments 2D, les options `CHAR_MECA_PRES_R`, `CHAR_MECA_PRES_F`, `CHAR_MECA_PCISA_R et CHAR_MECA_CISA_F` permettent d'appliquer des efforts mécaniques de pression répartie et de cisaillement sur les facettes de contact. Seules les options `CHAR_MECA_PRES_R` et `CHAR_MECA_PRES_F` sont actuellement programmées pour les éléments 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.

La figure ci dessous (Figure 3.6.2-1) illustre l'intégration surfacique numérique sur les facettes de contact d'un quadrangle. En chaque point de Gauss de la facette, on détermine un vecteur unitaire normal et un vecteur unitaire tangent uniquement à l'aide de la topologie des facettes de contact.

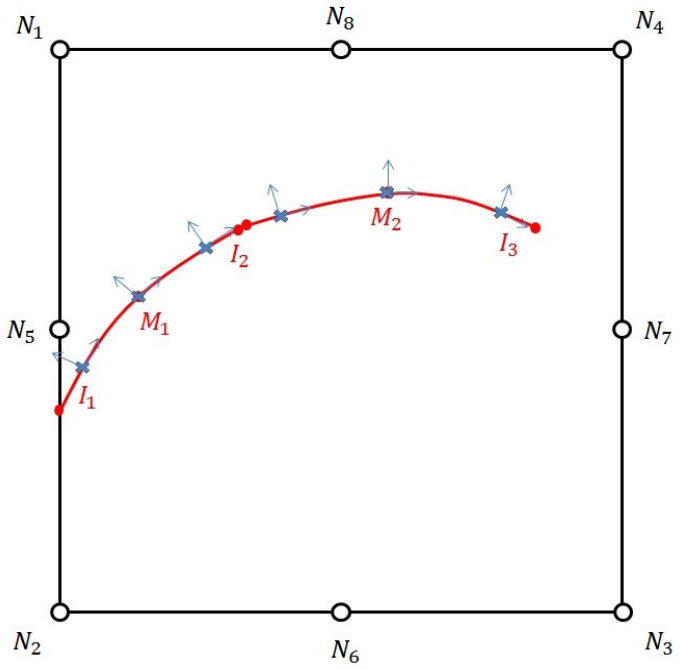


Figure 3.6.2-1 : intégration surfacique sur les facettes d'un quadrangle

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

En 3D, on détermine de même en chaque point de Gauss un vecteur normal unitaire uniquement à partir de la topologie des facettes de contact. On peut éventuellement créer une base orthonormée en chaque point de Gauss en complétant ce vecteur normal par deux vecteurs unitaires tangents à la discontinuité et orthogonaux entre eux.

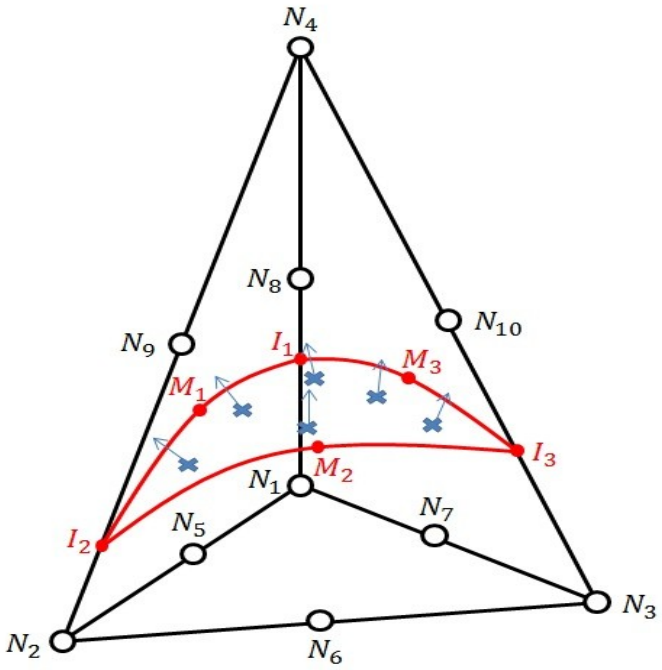


Figure 3.6.2-2 : intégration surfacique sur les facettes d'un tétraèdre.

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

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 3 « Problème de fissuration avec X-FEM ». 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 [bib74]. 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

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

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 $n_{dim}=2 \text{ ou } 3$. On note $\boldsymbol{n}$ la normale sortante à sa frontière $\partial\Omega$, Γ_1^c et Γ_2^c les lèvres de la fissure de normales sortantes respectives $\boldsymbol{n_1}$ et $\boldsymbol{n_2}$, et Γ^d la partie de $\partial\Omega$ soumise à une température imposée (voir Figure 3.3.8.1-3). On note $\boldsymbol{x}$ 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.

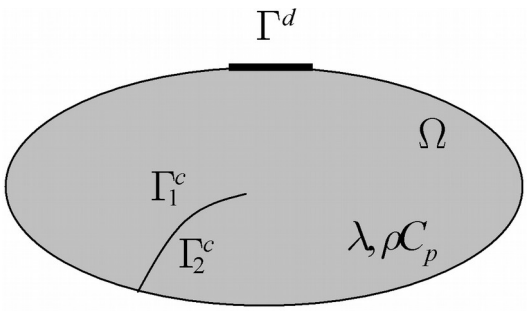


Figure 4.3-1: Notations

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 :

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

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 :

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

avec la la condition initiale :

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

On introduit ensuite les espaces fonctionnels suivants :

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

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

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

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

La discrétisation en temps est réalisée par un θ -schéma (voir [bib74]). 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 :

$$T^+=T(\boldsymbol{x}, t_{n+1}) \text{, et } T^-=T(\boldsymbol{x}, t_n)$$
$$V^+=\left\{v\in H^1(\Omega), v_{|\Gamma^d}=\bar{T}^+\right\} \text{ et } V^-=\left\{v\in H^1(\Omega), v_{|\Gamma^d}=\bar{T}^-\right\}$$

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

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

4.4 Approximation X-FEM du champ de température

On utilise les notations précédemment adoptées pour la mécanique au paragraphe 3.2 « Enrichissement de l'approximation du déplacement ». On rappelle l'approximation éléments finis classique :

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

L'approximation enrichie s'écrit quant à elle :

$$T^h(\boldsymbol{x})=\sum_{i\in N_n(x)} T_i^C \phi_i(\boldsymbol{x}) + \sum_{j\in N_n(x)\cap K} T_j^H \phi_j(\boldsymbol{x}) H_j\left(lsn(\boldsymbol{x})\right) + \sum_{k\in N_n(x)\cap L} T_k^{CT} \phi_k(\boldsymbol{x}) F^1\left(lsn(\boldsymbol{x}), lst(\boldsymbol{x})\right)$$

Cette expression est composée de 3 termes. Le 1er 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 de selection de domaine H_j 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.

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

On suppose que le domaine est partionné Ω par l'interface de la fissure de la fissure Γ^c en deux domaines Ω_- du côté de Γ_1^c et Ω_+ du côté de Γ_2^c .

On rappelle ci-dessous l'expression de H_j :

$$Si \quad x_j \in \Omega_+, \quad H_j(x) = \begin{cases} 0 & \text{si } x \in \Omega_+ \\ -2 & \text{si } x \notin \Omega_+ \end{cases} \qquad Si \quad x_j \in \Omega_-, \quad H_j(x) = \begin{cases} 0 & \text{si } x \in \Omega_- \\ +2 & \text{si } x \notin \Omega_- \end{cases}$$

On rappelle ci-dessous l'expression de F^1 :

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

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

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 75] 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 $+$), 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 $n_{dim} = 2 \text{ ou } 3$, 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 :

$$\{\mathbf{T}\}_E^T = \left[T_1^C \;\; T_1^H \;\; T_1^{CT} \;\; \ldots \;\; T_n^C \;\; T_n^H \;\; T_n^{CT} \right] \text{ de taille } 3n$$

$$[\mathbf{N}]_E = \left[N_1 \;\; H_1 N_1 \;\; F^1 N_1 \;\; \ldots \;\; N_n \;\; H_n N_n \;\; F^1 N_n \right] \text{ de taille } (1, 3n)$$

$$[\mathbf{B}]_E = \left[\nabla N_1 \;\; \nabla(H_1 N_1) \;\; \nabla(F^1 N_1) \;\; \ldots \;\; \nabla N_n \;\; \nabla(H_n N_n) \;\; \nabla(F^1 N_n) \right] \text{ de taille } (n_{dim}, 3n)$$

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 90. On donne ci-dessous l'expression de ces quantités élémentaires.

matrice de masse (option MASS_THER) :

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

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

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 94.

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 :

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

Remarque :

D'une matrice à l'autre, les coefficients correspondant à l'enrichissement classique sont égaux, et les coefficients correspondants à l'enrichissement Heaviside et « Crack-Tip » sont reliés par : $H_{|\Gamma_2^c} - H_{|\Gamma_1^c} = 2$, et $F^1_{|\Gamma_2^c} - F^1_{|\Gamma_1^c} = 2\, F^1_{|\Gamma_1^c}$ *car l'angle polaire vaut* $\pm\pi$.

On introduit ensuite les matrices de couplage suivantes :

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

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) :

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

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

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 :

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

C est le tenseur de Hooke (tenseur du 4ème 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) :

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

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 :

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

$$\delta W=\int_{\Omega}\frac{\partial \Psi}{\partial \varepsilon_{ij}}\delta\varepsilon_{ij}\,d\Omega-\int_{\Omega}f_i\delta v_i\,d\Omega-\int_{S}g_i\delta v_i\,d\Gamma$$
$$=\int_{\Omega}\sigma_{ij}\frac{1}{2}\left(\delta v_{i,j}+\delta v_{j,i}\right)d\Omega-\int_{\Omega}f_i\delta v_i\,d\Omega-\int_{S}g_i\delta v_i\,d\Gamma$$
$$=\int_{\Omega}\sigma_{ij}\delta v_{i,j}\,d\Omega-\int_{\Omega}f_i\delta v_i\,d\Omega-\int_{S}g_i\delta v_i\,d\Gamma=0$$

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

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

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 introduite 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 :

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

Soit m la normale unitaire au fond de fissure, située dans le plan tangent de la fissure.
D'après [bib50], le taux de restitution d'énergie local G est solution de l'équation variationnelle suivante :

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

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 :

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

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 :

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

En fait, d ans 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,

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

$$-G(\theta) = \frac{\partial}{\partial \eta}\left(\int_\Omega \Psi - f_i u_i \, d\Omega - \int_S g_i u_i \, d\Gamma\right)\Big|_{\eta=0}$$

$$= \int_\Omega \dot{\overbrace{\Psi - f_i u_i}} + \left(\Psi - f_i u_i\right) div(\theta) \; d\Omega - \int_S \dot{\overbrace{g_i u_i}} + g_i \, u_i \left(div(\theta) - \frac{\partial \theta}{\partial n_k} \, n_k\right) d\Gamma$$

où:

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

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

donc :

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

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

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,

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

donc

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

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

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

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

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

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 $J(\theta)$:

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

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 $\left(p_j(s)\right)_{1\le j\le N}$:

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

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

$$\overline{\theta}^i(s) = \sum_{k=1}^{M} \theta^i_k q_k(s) \qquad \textbf{éq. 5.1.3-1}$$

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

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

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

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

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

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 [bib49].

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 :

$$g(u,v)=\frac{1}{4}\left(G(u+v)-G(u-v)\right) \qquad \textbf{éq. 5.2.1-1}$$

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 :

$$\varepsilon(u)=\nabla_s u, \quad \varepsilon(v)=\nabla_s v$$
$$\sigma(u)=C:\varepsilon(u), \quad \sigma(v)=C:\varepsilon(v)$$

En reprenant les notations de [bib49], le terme classique de $J(\theta)$ [éq. 5.1.2-1] est :

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

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

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

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 :

$$S1=C1.S11+C2.S12+C3.S13$$

en posant :

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

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

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 :

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

L'expression de la partie thermique est :

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

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

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

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 :

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

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

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

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

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

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

L'expression de la partie thermique est :

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

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

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

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

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 :

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

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

Il reste donc :

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

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

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

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 :

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

5.2.1.4 Terme thermique

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

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

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

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 :

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

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

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

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 [bib53] 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 :

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

On retrouve l'analogue de ces expressions écrites avec des intégrales d'interaction dans [bib54] et [bib53].

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

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

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 $\left(e_1,e_2,e_3\right)$ issue des gradients des level sets , or les dérivées intervenants dans $g\left(u\,,u^S_{I\,,II\,,III}\right)_\theta$ sont faites par rapport au repère global $\left(E_1,E_2,E_3\right)$ (voir les expressions des termes $S1$ et $S2$ au paragraphe précédent). Il faut donc procéder à un changement de base.

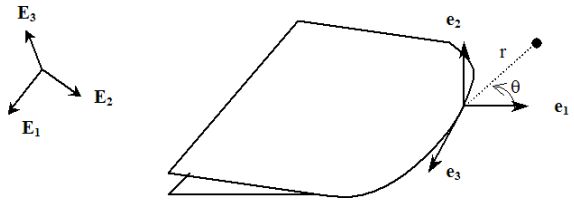


Figure 5.2.2-1 : Coordonnées polaires, base locale et base globale

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 $\left(r\,,\theta\right)$. Les dérivées de u par rapport aux variables $\left(r\,,\theta\right)$ 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 $\left(e_1,e_2,e_3\right)$:

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

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

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

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

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

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

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 :

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

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

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

Pour les champs thêta, 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 :

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

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.

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

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

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

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. 2 89-314, 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

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

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, 2 nd 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

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

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

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

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 approac h 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

76 A. Hansbo, P. Hansbo. A finite element method for the simulation of strong and weak discontinuities in solid mechanics, Comput. Methods Appl. Mech. Engrg., volume 193, 3523–3540, 2004.

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
D	11	M. NDEFFO, P. MASSIN, EDF R&D LaMSID	Rajout des éléments 3D quadratiques

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