# Opérateur DEFI_FISS_XFEM

## 1 But

Définir une fissure ou un interface 2D ou 3D par le biais de fonctions de niveaux.

Pour les fissures, on définit deux fonctions de niveaux (level sets) permettant de caractériser une fissure quelconque (fissure plane ou non). Pour les interfaces, une seule level-set est nécessaire. Les level sets peuvent être définies de quatre façons différentes :

- soit à partir de deux groupes de mailles (le groupe de mailles de la fissure et le groupe de mailles du fond de fissure) ;
- soit à partir de deux fonctions analytiques.
- soit à partir d'un catalogue de formes prédéfinies.
- soit en relisant des champs aux nœuds déjà existant.

Pour le cas 2D, les mailles du fond de fissure sont des mailles POI ; elles ne doivent pas être données en tant que groupe de nœud mais en tant que groupe de mailles.
Pour le cas 3D, le fond de fissure est automatiquement orienté. La liste ordonnée des points du fond est automatiquement affichée dans le fichier de messages, lors du passage dans la commande DEFI_FISS_XFEM. La méthode d'orientation et le cas spécifique où elle ne fonctionne pas, sont décrits dans la documentation [R7.02.12].

Cet opérateur ne permet pas de définir une fissure coïncidant avec un bord de la structure.

Les caractéristiques liées au contact sur les lèvres de la fissure ne sont pas données ici mais dans une charge de contact spécifique avec l'opérateur DEFI_CONTACT. On peut également définir des forces de cohésion entre les lèvres de la fissure, par ce même opérateur. La fissure est alors généralement déclarée comme étant de type COHESIF.

L'aspect théorique de la méthode X-FEM est abordée dans [R7.02.12], et il est conseillé à l'utilisateur qui n'est pas familier avec l'usage des level sets de lire les quelques lignes explicatives relatives à la représentation de surfaces par la méthode des level sets.

De plus, cet opérateur s'utilise aussi avec des fissures maillées, en préalable au calcul des facteurs d'intensité de contraintes en 3D par l'opérateur CALC_G [U4.82.03] avec l'option CALC_K_G.



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

L'opérateur produit un concept de type `fiss_xfem`.



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

# 2 Syntaxe

```
[fiss_xfem]   =  DEFI_FISS_XFEM(

♦ MAILLAGE        =  ma,                                  [maillage]

# Définition d'une grille à associer à la fissure
◊  / MAILLAGE_GRILLE   =  magri,                          [maillage]
   / FISS_GRILLE       = fiss_orig,                       [fiss_xfem]

# choix d'une fissure ou d'une interface
◊  TYPE_DISCONTINUITE     =  /  'FISSURE',                [DEFAUT]
                             /  'COHESIF'
                             /  'INTERFACE'

# Définition de la fissure ou de l'interface

♦  DEFI_FISS =  _F  (

# Soit on définit le groupe de mailles d'une lèvre et le groupe de mailles du fond de fissure :
      ♦  /  ♦  GROUP_MA_FISS    =  grma,                  [l_gr_maille]
            ◊  GROUP_MA_FOND    =  grma,                  [l_gr_maille]

# Soit on définit deux fonctions analytiques ; une pour caractériser la surface de la fissure, et l'autre pour caractériser le fond de fissure :

         /  ♦  FONC_LN          =  fonc,                  [fonction]
            ◊  FONC_LT          =  fonc,                  [fonction]
# Pour une fissure de type COHESIF :

         /  ♦  GROUP_MA_BORD    =  group_ma,              [l_gr_maille]

# Soit on utilise une forme de fissure ou d'interface pré-définie et on donne les caractéristiques géométriques :

         /  ♦  FORM_FISS    =  form,              [K8]

            # Si form = 'ELLIPSE'
            ♦  DEMI_GRAND_AXE   =  a,                    [R]
            ♦  DEMI_PETIT_AXE   =  b,                    [R]
            ♦  CENTRE           =  (x0, y0, z0),         [l_R]
            ♦  VECT_X           =  (vxx, vxy, vxz),      [l_R]
            ♦  VECT_Y           =  (vyx, vyy, vyz),      [l_R]
            ◊  COTE_FISS        =  /  'IN',              [DEFAUT]
                                   /  'OUT',
            # Si form = 'RECTANGLE'
            ♦  DEMI_GRAND_AXE   =  a,                    [R]
            ♦  DEMI_PETIT_AXE   =  b,                    [R]
            ◊  RAYON_CONGE      =  /  0                  [DEFAUT]
                                   /  r,                 [R]
            ♦  CENTRE           =  (x0, y0, z0),         [l_R]
            ♦  VECT_X           =  (vxx, vxy, vxz),      [l_R]
            ♦  VECT_Y           =  (vyx, vyy, vyz),      [l_R]
            ◊  COTE_FISS        =  /  'IN',              [DEFAUT]
                                   /  'OUT',

            # Si form = 'CYLINDRE'
```



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

```
                ♦  DEMI_GRAND_AXE   =  a,                  [R]
                ♦  DEMI_PETIT_AXE   =  b,                  [R]
                ♦  CENTRE           =  (x0, y0, z0),       [l_R]
                ♦  VECT_X           =  (vxx, vxy, vxz),    [l_R]
                ♦  VECT_Y           =  (vyx, vyy, vyz),    [l_R]

                #  Si form = 'ENTAILLE'
                ♦  DEMI_LONGUEUR    =  a,
                ♦   RAYON_CONGE     =  r,
                ♦  CENTRE           =  (x0, y0, z0),
                ♦  VECT_X           =  (vxx, vxy, vxz),
                ♦  VECT_Y           =  (vyx, vyy, vyz),

                #  Si form = 'DEMI_PLAN'
                ♦  PFON             =  (x0, y0, z0),       [l_R]
                ♦  NORMALE          =  (vnx, vny, vnz),    [l_R]
                ♦  DTAN             =  (vtx, vty, vtz),    [l_R]

                #  Si form = SEGMENT'
                ♦  PFON_ORIG        =  (x0, y0, z0),       [l_R]
                ♦  PFON_EXTR        =  (x1, y1, z1),       [l_R]

                #  Si form = 'DEMI_DROITE'
                ♦  PFON             =  (x0, y0, z0),       [l_R]
                ♦  DTAN             =  (vtx, vty, vtz),    [l_R]

                #  Si form = 'DROITE'
                ♦  POINT            =  (x0, y0, z0),       [l_R]
                ♦  DTAN             =  (vtx, vty, vtz),    [l_R]

                      ),
```

# Soit on relit des champs aux nœuds déjà existants

```
          /  ♦  CHAM_NO_LSN     =  ch_lsn,          [cham_no]
             ◊  CHAM_NO_LST     =  ch_lst,          [cham_no]
```

# Définition de la zone d'enrichissement

```
◊  GROUP_MA_ENRI       =  grma,                    [l_gr_maille]
◊  CHAM_DISCONTINUITE  =  'DEPL',                  [DEFAUT]
◊  TYPE_ENRI_FOND      =  /  'GEOMETRIQUE'         [DEFAUT]
                          /  'TOPOLOGIQUE'

#  Si TYPE_ENRI_FOND   =  'GEOMETRIQUE'
   ◊  RAYON_ENRI       =  Renri,                   [R]

   # Si RAYON_ENRI  =   None
   ◊  NB_COUCHES       =  /  4,                       [DEFAUT]
                          =  /  N ,                    [I]
```

# Définition d'une jonction sur la fissure

# Si (**MAILLAGE**_GRILLE = None et FISS_GRILLE = None),

```
◊  JONCTION  =  _F (

      ♦  FISSURE      =  (fiss1, fiss2,..., fissi),   [l_fiss_xfem]
      ♦  POINT        =  (Pox , Poy , Poz),           [l_R]
```



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

# Impression d'informations

```
◊ INFO = / 1,                                          [DEFAUT]
         / 2,
         / 3,
  )
```



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

# 3 Opérandes

## 3.1 Opérande MAILLAGE

♦ MAILLAGE = ma

ma : nom du maillage sur lequel on va définir la fissure. Ce maillage est obligatoirement un maillage 2D ou 3D.

**Attention, les éléments X-FEM quadratiques manquent de robustesse (surtout en 3D) on conseille d'utiliser des éléments X-FEM linéaires. Il est donc préférable que ma soit un maillage linéaire.**

## 3.2 Définition d'une grille à associer à la fissure

Une grille auxiliaire peut être associée à la fissure:
- soit en donnant un maillage grille (voir §3.2.1),
- soit en donnant une fissure déjà associée à une grille précédente (voir §3.2.2),

Dans le cas d'utilisation de la méthode upwind dans PROPA_FISS (doc U4.82.11), son utilisation est presque obligatoire.
Ce mot clé est utilisé typiquement suite à un raffinement par Homard du maillage où la fissure a été définie.

### 3.2.1 Opérande MAILLAGE_GRILLE

◊ MAILLAGE_GRILLE = magri [maillage]

magri : nom du maillage de la grille à associer à la fissure

Cet opérande est utilisé pour associer le maillage grille magri à la fissure. Les deux fonctions de niveau (level set) et leurs gradients sont calculés à la fois sur le maillage ma donné par MAILLAGE (voir §3.1) et sur le maillage magri selon la méthode choisie par DEFI_FISS (voir §3.4) et ils sont stockés avec la fissure. Donc plusieurs fissures peuvent être définies sur la même grille magri.
Le maillage magri est créé par l'opérateur DEFI_GRILLE [U4.24.02].

La grille auxiliaire utilisée doit s'étendre seulement sur la zone prévue de propagation de la fissure de façon à pouvoir correctement représenter le fond de la fissure pendant toute la simulation de la propagation. Elle peut être en partie à l'extérieur du maillage ma (mot clé MAILLAGE, voir §3.1) et aucun rapport n'existe entre ce maillage et celui de la grille magri. En présence de plusieurs fissures et avec une utilisation de grille différente pour chacune, les grilles sont indépendantes et peuvent se superposer partiellement ou totalement. Dans ce cas on pourrait aussi utiliser une seule grille pour toutes les fissures.

### 3.2.2 Opérande FISS_GRILLE

Dans le cas où l'on a défini une fissure sur un maillage avec une grille associée et où on veut la redéfinir sur un autre maillage sans changer de grille, on peut économiser du temps de calcul en évitant de recalculer les fonctions de niveau sur la grille. En effet, dans ce cas les fonctions de niveau définies sur la grille ne changent pas.
Cela est toujours le cas quand on utilise Homard sur le maillage où la fissure a été définie parce que la grille n'est pas affectée par les modifications de Homard et les fonctions de niveau définies initialement ne changent pas. On peut donc redéfinir la même fissure sur le maillage raffiné avec la même grille associée en calculant les fonctions de niveau seulement sur le maillage et en les gardant à l'identique sur la grille.



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

◊ FISS_GRILLE = fiss_orig, [fiss_xfem]

On spécifie la fissure fiss_orig qui a été définie avant. La grille déjà présente est associée à la nouvelle fissure et les champs aux nœuds calculés qui définissent les fonctions de niveau et leur gradients sont recopiés sur la grille pour la nouvelle fissure. Cela équivaut a définir la fissure fiss_orig sur le nouveau maillage en utilisant toujours la même grille.

**Remarque :**

*Aucune vérification ne peut être faite sur la cohérence entre la fissure qui va être définie par DEFI_FISS (voir § 3.4 ) et celle qui a été définie sur la grille de fiss_orig .*

## 3.3 Opérande TYPE_DISCONTINUITE

◊ TYPE_DISCONTINUITE = / 'FISSURE', [DEFAUT]
/ 'COHESIF'
/ 'INTERFACE'

Par défaut, on définit une fissure. Il est cependant possible de définir une interface (pour traiter le cas de 2 solides en contact, ou des sous-épaisseurs, …).
Dans le cas où TYPE_DISCONTINUITE = 'FISSURE', la structure de données produite est enrichie de deux tables nommées 'FOND_FISS' et 'NB_FOND_FISS' récupérables en utilisant la commande RECU_TABLE. La table 'NB_FOND_FISS' contient une seule valeur correspondant au nombre de fonds de fissure. La table contient donc une seule colonne (nommée NOMBRE) et une seule ligne. La table 'FOND_FISS' contient la liste des coordonnées des points de tous les fond de fissure. Cette table 6 colonnes :

- la première colonne ( 'NUME_FOND' ) correspond au numéro du fond de fissure,
- la colonne suivante ( 'NUM_PT' ) correspond au numéro du point du fond de fissure, ce numéro est local au fond de fissure courant,
- la colonne suivante ( 'ABSC_CURV' ) correspond à la valeur de l'abscisse curviligne le long du fond de fissure courant,
- les 3 colonnes suivantes ( 'COOR_X' , 'COOR_Y' et 'COOR_Z' ) correspondent aux coordonnées du point.

Le type de discontinuité COHESIF permet de disposer d'informations sur le front de propagation (du fait des données précédemment mentionnées et la seconde fonction de niveau) afin de les transmettre à des algorithmes de propagation ou de suivi de la surface fissurée. En revanche, contrairement au type FISSURE , il ne contient pas d'informations liées à un enrichissement en pointe, puisque ce dernier n'a pas lieu d'être en présence de forces de cohésion. En effet, il est important de noter que pour le type de discontinuité COHESIF , le front de propagation ne coïncide pas avec la fin de l'enrichissement Heaviside, lequel peut traverser toute la structure. Le front initial doit donc être donné de façon indépendante, par le mot-clé GROUP_MA_BORD .

## 3.4 Mot clé DEFI_FISS

♦ DEFI_FISS = _F

Le mot-clé facteur DEFI_FISS permet de définir une fissure ou une interface de 4 manières différentes :

- soit à partir de deux groupes de mailles (voir §3.4.1) ;
- soit à partir de deux fonctions analytiques (voir §3.4.2) ;
- soit à partir d'un catalogue de formes prédéfinies (voir §3.4.3) ;
- soit à partir de champs existants (voir §3.4.4).

### 3.4.1 Définition de la fissure par des groupes de mailles



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

Si on a une fissure (ou une interface) déjà maillée, alors on peut définir la fissure en donnant deux groupe de mailles : GROUP_MA_FISS et GROUP_MA_FOND. Ce maillage peut être indépendant du maillage de la structure.

/ ♦ GROUP_MA_FISS = grma

avec grma groupe de mailles unique.

Ce groupe de mailles correspond à une seule des lèvres de la fissure (lèvre inférieure ou supérieure). Ce groupe de mailles doit être orienté. Dans le cas de lèvres qui ne seraient pas parfaitement collées, le fait de privilégier un côté aura une légère influence sur la base locale du fond de fissure, influence d'autant plus grande que l'angle entre les deux lèvres est important. Pour le cas 2D, le programme trie les mailles de GROUP_MA_FISS de façon à avoir un groupe de mailles contiguës quel que soit le groupe de départ. Il n'est donc pas nécessaire de donner un groupe de mailles qui se suivent.

◊ GROUP_MA_FOND = grma

avec grma groupe de mailles unique.

Cet opérande sert à définir le fond de fissure. Pour les interfaces, GROUP_MA_FOND n'est pas à renseigner.

En 2D, GROUP_MA_FOND est un groupe de mailles point.

### 3.4.2 Définition de la fissure par des fonctions analytiques

Le principal intérêt de cet opérateur est la possibilité de définir une fissure sans que celle-ci ne soit forcément maillée. Dans ce cas, on définit la fissure à l'aide de deux fonctions de niveaux (level sets). La première level set (dite level set « normale ») est celle qui permet de caractériser la surface de la fissure. On renseigne donc FONC_LN avec une fonction réelle de $X$ , $Y$ et $Z$ définie au préalable par l'opérateur FORMULE. La surface de la fissure sera alors l'iso-zéro de cette fonction. La seconde level set (dite level set « tangente ») est celle qui permet de caractériser le fond de fissure. On renseigne donc FONC_LT avec une fonction réelle de $X$ , $Y$ et $Z$ définie au préalable par l'opérateur FORMULE. La trace de l'iso-zéro de FONC_LT dans le plan de fissuration est le fond de fissure. Les points de la fissure sont alors caractérisés par FONC_LN = 0 et FONC_LT < 0, alors que le fond de fissure est caractérisé par FONC_LN = 0 et FONC_LT = 0.

Pour les interfaces, FONC_LT n'est pas à renseigner.

/ ♦ FONC_LN = fonc,

avec fonc une fonction ou une formule définie auparavant.

◊ FONC_LT = fonc,

avec fonc une fonction ou une formule définie auparavant.

**Remarque :**

*Pour une fissure, il est primordial que les level sets soient les vraies fonctions distances, on ne peut donc pas définir une fissure elliptique par :*
*LN=FORMULE(NOM_PARA=('X','Y','Z'),VALE='Z-H')*
*LT=FORMULE(NOM_PARA=('X','Y','Z'),VALE='(X/a)²+(Y/b)²-1')*
*où a et b seraient les demi-grand axe et demi-petit axe de l'ellipse. En effet, la fonction LT n'est pas la distance à l'ellipse. La fonction distance à une ellipse ne s'exprime pas à l'aide de fonctions usuelles. Pour introduite une telle fissure, il faut utiliser la 3 ème méthode : les formes pré-définies.*
*Pour les interfaces, n'importe quelle fonction level set est possible, et il n'est pas utile d'introduire la vraie fonction distance.*



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

**Remarque :**

*A ce jour rien n'interdirait d'avoir une level set qui définit plusieurs fonds de fissure fermés (par exemple plusieurs fissures de type « penny-shaped »). Dans ces conditions, un appel à DEFI_FISS_XFEM pourraient alors définir plusieurs fissures. Cette possibilité est interdite : chaque commande DEFI_FISS_XFEM ne définit qu'une seule fissure. Il n'est donc pas possible de définir plusieurs fonds fermés OU un fond fermé plus un ou plusieurs fonds ouverts.*

**Remarque :**

*Attention au signe de FONC_LT , toujours se rappeler que la fissure est définie du côté FONC_LT < 0.*

### 3.4.3 Définition de la fissure par un catalogue de forme pré-définies

Il existe un catalogue de formes de fissure/interface pré-définies. Le choix de la forme dépend de la dimension du maillage (2D ou 3D). Les choix possibles sont répertoriés dans le Tableau 3.4.3-1.

| | Fissures | Interfaces |
|---|---|---|
| 2D | SEGMENT, DEMI_DROITE | DROITE , ELLIPSE , RECTANGLE , ENTAILLE |
| 3D | ELLIPSE, RECTANGLE, CYLINDRE, DEMI_PLAN | |

**Tableau 3.4.3-1: Catalogue des formes pré-définies**

Pour chaque forme, il faut donner les caractéristiques géométriques spécifiques.

```
/  ♦  FORM_FISS     =  form,
```

#### 3.4.3.1 FORM_FISS = 'ELLIPSE'

Cas d'une fissure plane elliptique en 3D ou d'une inclusion de forme elliptique en 2D. Il faut renseigner la longueur du demi-grand axe par le mot-clé DEMI_GRAND_AXE et la longueur du demi-petit axe par le mot-clé DEMI_PETIT_AXE. Pour caractériser le plan de l'ellipse, on doit renseigner le centre de l'ellipse par le mot-clé CENTRE et le repère du plan par les mot-clés VECT_X et VECT_Y. Le vecteur VECT_X correspond à l'axe de l'ellipse le long duquel on définit la longueur $a$ donnée sous le mot-clé DEMI_GRAND_AXE, et le vecteur VECT_Y correspond à l'axe de l'ellipse le long duquel on définit la longueur $b$ donnée sous le mot-clé DEMI_PETIT_AXE.

*Remarque :*

*Qu'on choisisse* $0<b\leqslant a$ *ou* $0<a\leqslant b$ *,* VECT_X *reste toujours associé à* DEMI_GRAND_AXE *et* VECT_Y *reste toujours associé à* DEMI_PETIT_AXE *.*

Dans le cas d'une fissure, on peut de plus spécifier de quel coté se trouve la fissure (à l'intérieur ou à l'extérieur de l'ellipse) par le mot-clé COTE_FISS . La Figure 3.4.3.1-1 représente une fissure/inclusion elliptique, de demi-grand axe $a$ , de demi-petit axe $b$ et de centre $C$ . Le vecteur $\vec{e}_1$ correspond au mot-clé VECT_X et le vecteur $\vec{e}_2$ correspond au mot-clé VECT_Y .

```
♦  DEMI_GRAND_AXE   =  a,
♦  DEMI_PETIT_AXE   =  b,
♦  CENTRE           =  (Cx, Cy, Cz),
♦  VECT_X           =  (e1x, e1y, e1z),
♦  VECT_Y           =  (e2x, e2y, e2z),
◊  COTE_FISS        =  /  'IN',
                       /  'OUT',
```



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

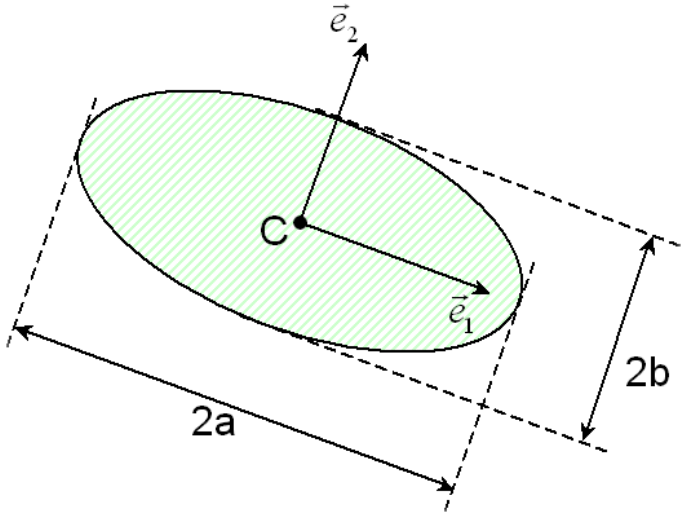



**Figure 3.4.3.1-1: Schéma d'une fissure/inclusion elliptique**

### 3.4.3.2 FORM_FISS = 'RECTANGLE'

Cas d'une fissure plane de forme rectangulaire en 3D, ou d'une inclusion de forme rectangulaire en 2D, avec possibilité de coins arrondis. C'est une forme similaire à l'ellipse, il faut renseigner les même mot-clés : la valeur du demi-grand axe (ou demi-longueur) par le mot-clé DEMI_GRAND_AXE et la valeur du demi-petit axe (ou demi-largeur) par le mot-clé DEMI_PETIT_AXE . Pour caractériser le plan du rectangle, on doit renseigner le centre du rectangle par le mot-clé CENTRE et le repère du plan. Le demi-grand axe correspond au vecteur VECT_X et le demi-petit axe au vecteur VECT_Y . De plus, pour le cas d'une fissure, on peut spécifier de quel coté se trouve la fissure (à l'intérieur ou à l'extérieur du rectangle) par le mot-clé COTE_FISS . Le seul mot-clé en plus par rapport à la forme ellipse est celui servant à donner la valeur du rayon du congé pour les coins arrondis ( RAYON_CONGE ). La Figure 3.4.3.2-1 représente une fissure/inclusion de forme rectangulaire à coins arrondis, de demi-grand axe $a$ , de demi-petit axe $b$ et de centre $C$ . Le vecteur $\vec{e}_1$ correspond au mot-clé VECT_X et le vecteur $\vec{e}_2$ correspond au mot-clé VECT_Y. Le rayon de congé des coins arrondis est $r$ . Par défaut le rayon de congé est nul.

```
♦  DEMI_GRAND_AXE   =  a,
♦  DEMI_PETIT_AXE   =  b,
◊  RAYON_CONGE      =  r,
♦  CENTRE           =  (Cx, Cy, Cz),
♦  VECT_X           =  (e1x, e1y, e1z),
♦  VECT_Y           =  (e2x, e2y, e2z),
◊  COTE_FISS        =  /  'IN',
                       /  'OUT',
```



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

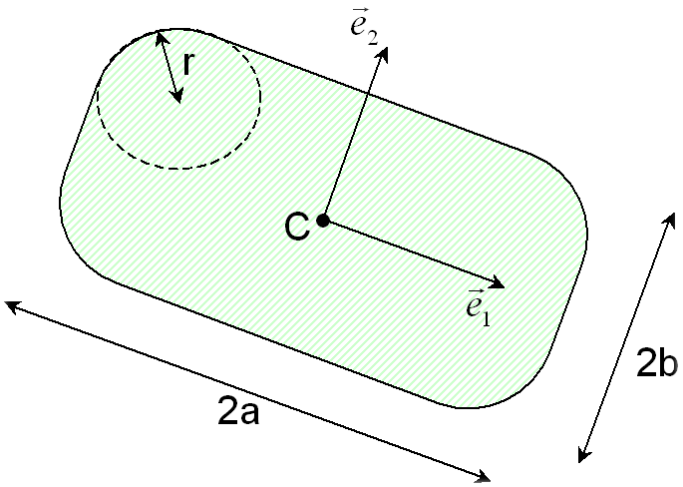



**Figure 3.4.3.2-1: schéma d'une fissure/inclusion rectangulaire à coins arrondis**

### 3.4.3.3 FORM_FISS = 'CYLINDRE'

Cas d'une fissure tridimensionnelle cylindrique à section elliptique. Pour caractériser le fond (elliptique) de la fissure, il faut renseigner la longueur du demi-grand axe par le mot-clé DEMI_GRAND_AXE et la longueur du demi-petit axe par le mot-clé DEMI_PETIT_AXE . On doit également renseigner le centre de l'ellipse par le mot-clé CENTRE ainsi que le repère du plan ( VECT_X , VECT_Y ) . Remarque : le demi-grand axe correspond au vecteur VECT_X et le demi-petit axe au vecteur VECT_Y . La génératrice du cylindre et la direction de propagation potentielle de la fissure sont données par le produit vectoriel de VECT_X par VECT_Y . Ce cylindre est semi-infini suivant cet axe. La Figure 3.4.3.3-1illustre une fissure de type cylindre de demi-grand axe $a$ , de demi-petit axe $b$ et de centre $C$ . Le vecteur $\vec{e}_1$ correspond au mot-clé VECT_X et le vecteur $\vec{e}_2$ correspond au mot-clé VECT_Y .

```
♦  DEMI_GRAND_AXE   =  a,
♦  DEMI_PETIT_AXE   =  b,
♦  CENTRE           =  (x0, y0, z0),
♦  VECT_X           =  (vxx, vxy, vxz),
♦  VECT_Y           =  (vyx, vyy, vyz),
```

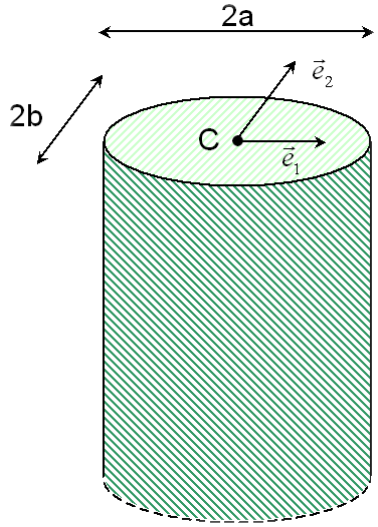



**Figure 3.4.3.3-1: Schéma d'une fissure cylindre**



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

#### 3.4.3.4 FORM_FISS = 'ENTAILLE'

Cas d'une inclusion de type entaille en 2D. C'est un cas particulier de l'inclusion de type rectangle à coins arrondis dans le où la demi-hauteur est égale au rayon de congé. Pour définir l'entaille il faut renseigner la valeur de la demi-longueur par le mot-clé DEMI_LONGUEUR , la valeur du rayon du congé par le mot-clé RAYON_CONGE et le centre de l'entaille par le mot-clé CENTRE . De plus, la direction de l'entaille est donnée par les mots-clé VECT_X et VECT_Y . La Figure 3.4.3.4-1 représente une inclusion de type entaille, de demi-longueur $a$ , de rayon de congé $r$ et de centre $C$ . Le vecteur $\vec{e}_1$ correspond au mot-clé VECT_X et le vecteur $\vec{e}_2$ correspond au mot-clé VECT_Y .

```
♦ DEMI_LONGUEUR    =  a,
♦ RAYON_CONGE      =  r,
♦ CENTRE           =  (Cx, Cy, Cz),
♦ VECT_X           =  (e1x, e1y, e1z),
♦ VECT_Y           =  (e2x, e2y, e2z),
```

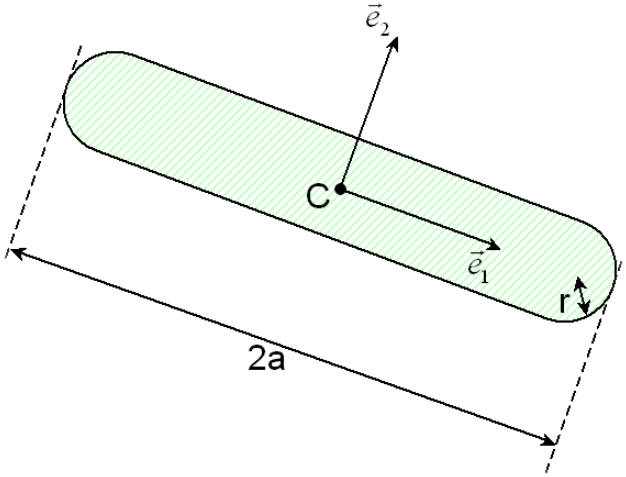



**Figure 3.4.3.4-1: schéma d'une fissure/inclusion rectangulaire à coins arrondis**

#### 3.4.3.5 FORM_FISS = 'DEMI_PLAN'

Cas d'une fissure définie par un demi-plan. PFON désigne un point du fond de fissure. NORMALE définit la normale au demi-plan de la fissure et DTAN fournit un vecteur dans le plan de fissure, orthogonale au fond et dirigé dans la direction de propagation potentielle. La Figure 3.4.3.5-1 représente une fissure de demi-plan définie par $P$ un point du fond, le vecteur $\vec{e}_1$ correspond au mot-clé DTAN et le vecteur $\vec{e}_2$ correspond au mot-clé NORMALE.

```
♦ PFON             =  (Px, Py, Pz),
♦ DTAN             =  (e1x, e1y, e1z),
♦ NORMALE          =  (e2x, e2y, e2z),
```



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

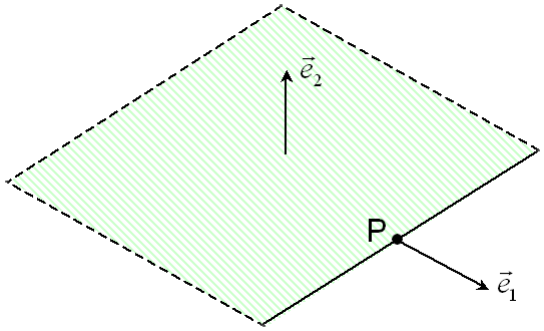



**Figure 3.4.3.5-1: schéma d'une fissure demi-plan**

#### 3.4.3.6 FORM_FISS = 'SEGMENT'

Cas d'une fissure 2D définit par un segment. PFON_ORIG et PFON_EXTR désignent les extrémités du segment. La Figure 3.4.3.6-1 présente une fissure 2D de type segment où $P1$ désigne le point d'origine PFON_ORIG et $P2$ le point extrémité PFON_EXTR .

- PFON_ORIG = (P1x, P1y, P1z),
- PFON_EXTR = (P2x, P2y, P2z),

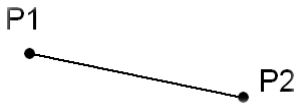



**Figure 3.4.3.6-1: schéma d'une fissure 2d de type segment**

#### 3.4.3.7 FORM_FISS = 'DEMI_DROITE'

Cas d'une fissure 2D définie par une demi-droite. PFON désigne le point du fond de fissure. DTAN correspond à un vecteur directeur de la demi-droite orienté dans la direction de propagation potentielle. La Figure 3.4.3.7-1 présente une fissure 2D de type demi-droite pour laquelle $P$ est le point du fond (PFON) et de vecteur directeur $\vec{e}_1$ (DTAN).

- PFON = (x0, y0, z0),
- DTAN = (e1x, e1y, e1z),

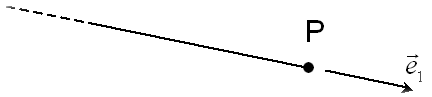



**Figure 3.4.3.7-1: schéma d'une fissure 2D de type demi-droite**

si form = DROITE

Cas d'une interface droite dans une structure 2D définie. POINT désigne une point de l'interface. DTAN correspond à un vecteur directeur de la droite. L'orientation n'est pas importante.

- POINT = (x0, y0, z0),
- DTAN = (vtx, vty, vtz),

### 3.4.4 Définition de la fissure par relecture de champs aux nœuds déjà existants



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

```
/   ♦   CHAM_NO_LSN      =   ch_lsn,              [cham_no]
    ◊   CHAM_NO_LST      =   ch_lst,              [cham_no]
```

Les champs ch_lsn et ch_lst sont des champs aux nœuds déjà existants. Ils doivent avoir été créés sur le même maillage que le maillage ma donné en entrée sous le mot-clé MAILLAGE (cf. §3.1).

Ces champs nodaux peut être obtenu successivement par la lecture d'une table (LIRE_TABLE), puis par la création du champ par extraction de la table (CREA_CHAMP/OPERATION='EXTR').
Ils ont aussi pu être générés dans une étape de raffinement de maillage (MACR_ADAP_MAIL/ MAJ_CHAM).

### 3.4.5 Définition du front de propagation initial pour une fissure cohésive

```
/   ♦   GROUP_MA_BORD    =   group_ma,            [l_gr_maille]
```

Pour une étude de propagation en présence de forces cohésives, il est nécessaire de donner le front de propagation initiale, c'est à dire la ligne à partir de laquelle la fissuration s'amorce. Il s'agit généralement d'un coin de la structure, ou d'un fond d'entaille, ou plus généralement d'une ligne située sur un bord. Elle est donnée soit sous forme d'un groupe de maille 1D si elle est maillée, soit sous forme d'une groupe de mailles 2D de bord, dont l'intersection avec la level-set normale fournira le front de propagation initial.

### 3.4.6 Exemples

- si on souhaite définir un barreau fissuré de part en part dans son plan à la hauteur $z=H$ (voir [Figure 3.2-a], sur laquelle la fissure est hachurée) :

```
LN=FORMULE(NOM_PARA=('X','Y','Z'),VALE='Z-H');
LT=FORMULE(NOM_PARA=('X','Y','Z'),VALE='Y-a');
```

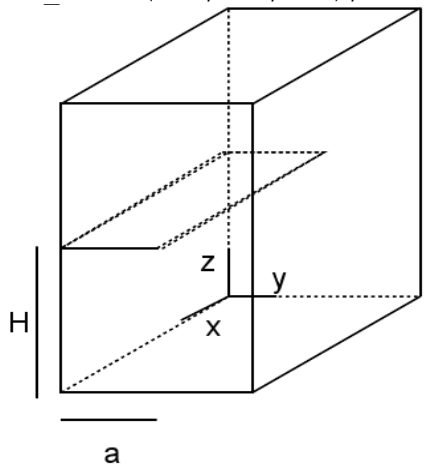



**Figure 3.2-a : Barreau fissuré**

- si on souhaite définir une fissure elliptique dans un massif infini dans un plan à la hauteur $z=H$ (voir [Figure 3.2-b], sur laquelle la fissure est hachurée) :

```
DEFI_FISS =  _F (  FORM_FISS     =  'ELLIPSE',
                   DEMI_GRAND_AXE  =  a,
                   DEMI_PETIT_AXE  =  b,
                   CENTRE          =  (0, 0, H),
                   VECT_X          =  (1, 0, 0),
```



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

```
VECT_Y          =  (0, 1, 0),
COTE_FISS       =  'IN',),
```

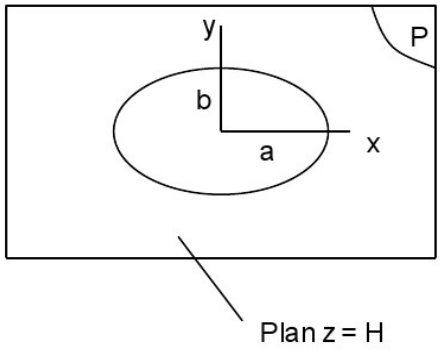



**Figure 3.2-b : Fissure elliptique**

## 3.5 Opérande GROUP_MA_ENRI

◊ GROUP_MA_ENRI = grma, [l_gr_maille]

Cet opérande permet de connaître la zone sur laquelle la procédure d'enrichissement va s'effectuer. Les nœuds enrichis doivent appartenir à ce groupe de maille.
Si GROUP_MA_ENRI n'est pas renseigné, alors la procédure d'enrichissement va s'effectuer sur tout le maillage ma donné en entrée sous le mot-clé MAILLAGE (cf. §3.1)

Une utilisation de ce mot-clé permet donc de délimiter la zone d'enrichissement, donc de la fissure. L'exemple suivant présente une fissure de forme elliptique dans un cylindre creux. La fissure est définie par la donnée de groupes de mailles (mots-clé GROUP_MA_FISS et GROUP_MA_FOND). Deux level sets en sont déduites, en considérant un prolongement du fond de fissure par continuité. Si la zone d'enrichissement contient toute la structure, cela conduit alors à une fissure dans la partie droite du cylindre, ce qui n'est pas du tout ce que l'on souhaite. Pour résoudre ce problème, il suffit de limiter la zone d'enrichissement et de renseigner le mot-clé GROUP_MA_ENRI avec les mailles de la moitié gauche du cylindre par exemple.

## 3.6 Opérande CHAM_DISCONTINUITE

◊ CHAM_DISCONTINUITE = 'DEPL', [DEFAUT]

Ce mot-clé permet de choisir le type de champs à enrichir. Pour le moment, seul le champ de déplacement peut être enrichi.

## 3.7 Opérandes TYPE_ENRI_FOND, RAYON_ENRI, NB_COUCHES

```
◊ TYPE_ENRI_FOND  =  /  'GEOMETRIQUE'          [DEFAUT]
                     /  'TOPOLOGIQUE'
```

Ce mot-clé permet de choisir le type d'enrichissement en fond de fissure. On active l'enrichissement topologique (une seule couche) si TYPE_ENRI_FOND = ' TOPOLOGIQUE'. On active l'enrichissement géométrique (plusieurs couches) si TYPE_ENRI_FOND = ' GEOMETRIQUE' (pour plus de renseignements concernant les effets de l'enrichissement sur la qualité des résultats, voir [R7.02.12] et [U2.05.02]). L'enrichissement géométrique est piloté par les mots-clé RAYON_ENRI et NB_COUCHES .



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

◊ RAYON_ENRI = Renri, [R]

Ce mot-clé ne peut être renseigné que si TYPE_ENRI_FOND = 'GEOMETRIQUE'. Si cette opérande est renseignée, il permet de préciser un critère géométrique Renri tel que tous les nœuds dont la distance au fond de fissure est inférieure à ce critère seront enrichis par les fonctions singulières. Cet enrichissement est appelé « géométrique ». Des études ont montré qu'un tel enrichissement améliorait grandement la précision des calculs, notamment lorsque le maillage est raffiné en fond de fissure. Il est conseillé de choisir Renri environ égal à 1/10 de la longueur de la fissure. Le conditionnement est très peu affecté par l'utilisation de l'enrichissent géométrique grâce à l'introduction de fonctions d'enrichissement vectorielles en fond de fissure [R7.02.12].

Pour éviter les désagréments causés par un trop grand rayon d'enrichissement (augmentation du coût de calcul), on peut définir la zone à enrichir en stipulant le nombre de couches d'éléments à enrichir autour du fond de fissure à la place du rayon d'enrichissement.

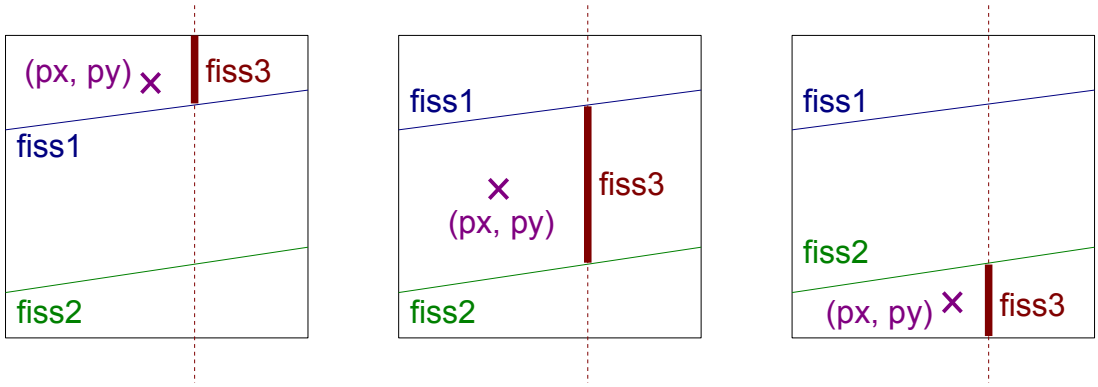
```
# Si RAYON_ENRI  =  None
◊   NB_COUCHES      =  /  2,                    [DEFAUT]
                       /  N ,                   [I]
```

Si RAYON_ENRI n'est pas renseigné (et que TYPE_ENRI_FOND = 'GEOMETRIQUE'), alors NB_COUCHES est le nombre de couches d'éléments à enrichir (2 par défaut). L'utilisation d'un grand nombre de couches impacte seulement le coût de calcul en particulièrement en 3D.

## 3.8 Opérande JONCTION

```
# Si MAILLAGE_GRILLE = None,
◊  JONCTION  =  _F (
      ♦  FISSURE      =  (fiss1, fiss2, …,fissi),     [l_fiss_xfem]
      ♦   POINT       =  (Px , Py , Pz),              [l_R]
```

Cet opérande permet de définir une ou plusieurs jonctions entre une fissure et d'autres fissures déjà existantes. Ces autres fissures sont alors considérées comme des fissures mères: il faut les avoir déjà préalablement définies par des appels à DEFI_FISS_XFEM.
On renseigne une liste de fissures mères dans le mot clé FISSURE . Toutes ces fissures mères seront prises en compte lors de la définition de la fissure courante . Le mot clé POINT doit être donné dans le domaine où l'on souhaite que la fissure soit définie. Un exemple est fourni figure 3.7-a. On considère sur cette figure que la fissure 3 a été définie au départ pour couper verticalement le domaine. On introduit le mot clé JONCTION afin de la brancher sur les fissures 1 et 2. On peut obtenir les trois configurations de la figure selon le choix de la position donnée dans le mot clé POINT .





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

**Figure 3.8-1 : Exemple de définition de la fissure fiss3, pour les fissures mères *fiss1* et *fiss2*, en fonction du point $(px, py)$ choisi.**

Il faut bien faire attention lorsqu'on utilise l'opérande JONCTION: la fissure ne sera alors plus seulement représentée par ses level sets normale et tangente, mais elle dépendra aussi des fissures mères. Cela peut poser problème si l'on souhaite utiliser des opérateurs comme RAFF_XFEM , qui admettent uniquement la définition d'une fissure par des level-sets. De même la représentation graphique de l'iso-zéro des level sets n'est pas suffisante, il est préférable de passer par les opérations de post-traitement via l'opérateur POST_MAIL_XFEM .

Un exemple de configuration est présenté Figure 3.8-2. Sur cet exemple on identifie les fissures sur lesquelles une autre fissure vient se brancher. Ainsi pour la fissure $N=3$ , on a $K=2$ où K est l'ensemble des fissures sur lesquelles 3 se branche. De même pour la fissure $N=5$ , on déduit $K=[3,4]$ . La fissure 5 est donc définie à partir de l'opérande jonction et des deux fissures 3 et 4. La fissure 3 est elle-même définie auparavant à partir de l'opérande jonction agissant sur la fissure 2, elle-même définie à partir de l'opérande jonction agissant sur la fissure 1. L'arbre de hiérarchie de la Figure 3.8-2 nous sert donc à établir l'ordre de définition des différentes fissures (2 à partir de 1, 3 à partir de 2, 4 à partir de 1 et 5 à partir de 3 et 4).

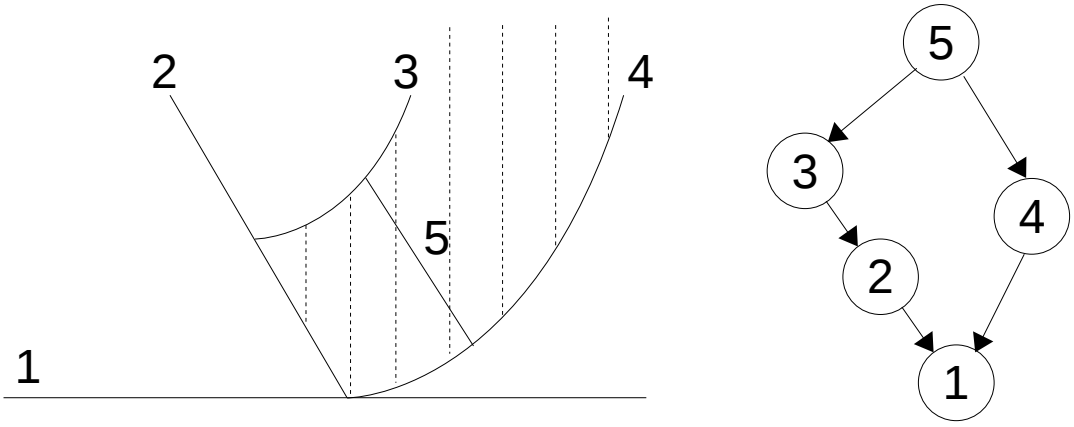



**Figure 3.8-2 : Réseau de fissures à gauche, arbre de hiérarchie à droite indiquant qui se branche et permettant d'établir les relations d'ordre pour les définitions de fissure et l'utilisation de l'opérande JONCTION.**

L'opérateur jonction permet d'établir une relation d'ordre pour une maille qui voit $N$ fissures. On distinguera ainsi sur la Figure 3.8-3 les deux arbres de hiérarchie de droite. Dans l'arbre le plus à droite, les fissures 1 et 2 sont suffisamment proches pour qu'une maille voyant la fissure 3, voit aussi les fissures 1 et 2 : on décide donc de brancher la fissure 3 sur les fissures 2 et 1, mais d'utiliser aussi l'opérande jonction pour indiquer que la fissure 2 est reliée à la fissure 1. Pour l'autre arbre, une maille voit au plus 2 fissures (aucune maille ne voit alors à la fois les fissures 1, 2 et 3) et dans ce cas la fissure 3 se branche simplement sur les fissures 2 et 1 qui ne sont plus reliées entre elles.



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

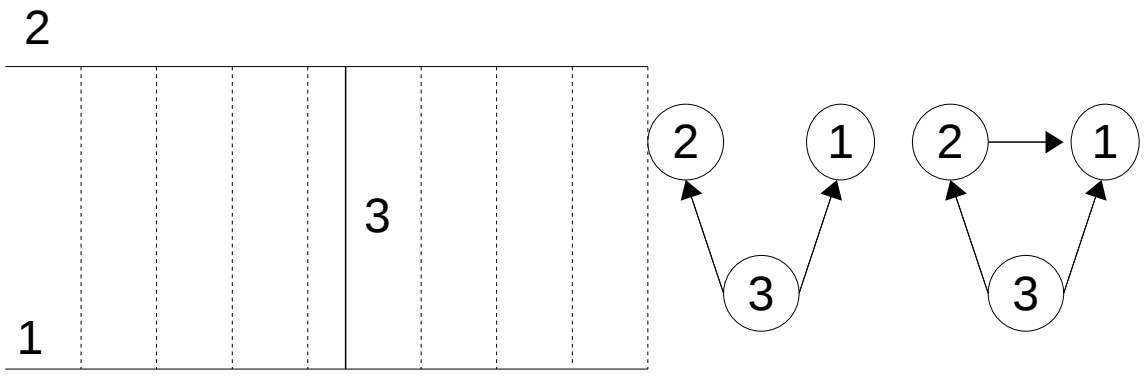



**Figure 3.8-3 : Réseau de fissures à gauche, arbres de hiérarchie à droite.**

L'arbre de hiérarchie le plus à droite indique que la fissure 2 ne peut pas être définie indépendamment de la fissure 1 car trop proche de cette dernière (une maille voyant la fissure 3 voit alors aussi les fissures 1 et 2).

## 3.9 Opérande INFO

/ 1 : impression sur le fichier 'MESSAGE'
1) du nombre de points du fond de fissure ;
2) du nombre de fonds de fissure ;
3) de la liste ordonnée des coordonnées des points du fond ainsi que leur abscisse curviligne, pour chaque fond de fissure.

/ 2 : mêmes impressions que 1 + impressions
1) du nombre de level sets réajustées ;
2) du nombre de mailles de la zone de la fissure ;
3) du nombre de mailles contenant le fond de fissure
4) du nombre de mailles de type Heaviside, Crack-tip et Heaviside Crack-tip.

/ 3 : même impression que 2 + impressions
5) des étapes du calcul ;
6) des champs level sets et de leurs gradients ;
7) du statut des nœuds ;
8) des numéros des mailles enrichies.



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

# 4 Exemples

## 4.1 Avec des formules analytiques

```
LN =  FORMULE   (  NOM_PARA=('X','Y','Z'),VALE=  'Z-12.5')
LT =  FORMULE   (  NOM_PARA=('X','Y','Z'),VALE=  'X-10'.)

FISS=  DEFI_FISS_XFEM   (  MAILLAGE  =  MAILLAIN,
                           DEFI_FISS =_F(   FONC_LT   =  LT,
                                            FONC_LN   =  LN,),
                           GROUP_MA_ENRI=  'VOL',
                           INFO=2,
                        )
```

## 4.2 Avec des groupes de mailles

```
FISS=  DEFI_FISS_XFEM   (  MAILLAGE  =  MAILLAIN,
                           DEFI_FISS =_F(   GROUP_MA_FISS    =  'LEVSUP',
                                            GROUP_MA_FOND    =  'FONFIS',),
                           GROUP_MA_ENRI    =  'VOL',
                           INFO=3,
                        )
```

## 4.3 Avec le catalogue de formes prédéfinies

Cas d'une fissure cylindrique :

```
FISS=DEFI_FISS_XFEM(MAILLAGE  =  MAILLAIN,
                    DEFI_FISS=_F(
                                 FORM_FISS = 'CYLINDRE',
                                 DEMI_GRAND_AXE = r,
                                 DEMI_PETIT_AXE = r,
                                 CENTRE         = (0., l_fiss ,0.),
                                 VECT_X         = (0., 0. ,1.),
                                 VECT_Y         = (1., 0. ,0.),
                                     ),
                    GROUP_MA_ENRI='VOL',
                    INFO=1,
                    );
```

Cas d'une fissure définie par un demi-plan :

```
 FISS=DEFI_FISS_XFEM(MAILLAGE =  MAILLAIN,
                    INFO=1,
                    DEFI_FISS=_F(
                                 FORM_FISS = 'DEMI_PLAN',
                                 PFON      =(0. , 5. ,15.),
                                 NORMALE   =(0. , 0. , 1.),
                                 DTAN      =(0. ,-1. , 0.)),
                    GROUP_MA_ENRI='VOL');
```



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