

---

## Operator `DEFI_FISS_XFEM`

---

### 1 Goal

---

To define a crack or an interface 2D or 3D by the means of functions of levels.

For the cracks, one defines two functions of levels (level sets) allowing to characterize an unspecified crack (crack planes or not). For the interfaces, only one level-set is necessary. The level sets can be defined in four different ways:

- maybe starting from two groups of meshes (the group of meshes of the crack and the group of meshes of the bottom of crack);
- maybe starting from two analytical functions.
- maybe starting from a catalogue of preset forms.
- maybe by reading again fields with the nodes already existing.

For the case 2D, the meshes of the bottom of crack are meshes `POI`; they should not be given as a group of node but as a group of meshes.

For the case 3D, the bottom of crack is automatically directed. The ordered list of the points of the bottom is automatically displayed in the file of messages, at the time of the passage in the order `DEFI_FISS_XFEM`. The method of orientation and the specific case where it does not function, are described in documentation [R7.02.12].

This operator does not allow to define a crack coinciding with an edge of the structure.

The characteristics related to the contact on the lips of the crack are not given here but in a specific load of contact with the operator `DEFI_CONTACT`. One can also define forces of cohesion between the lips of the crack, by this same operator. The crack then is generally declared as being of type `COHESIVE`.

The theoretical aspect of method X-FEM is approached in [R7.02.12], and he is advised with the user who is not familiar with the use of the level sets of reading the few explanatory lines relating to the representation of surfaces by the method of the level sets.

Moreover, this operator is also used with cracks with a grid, in precondition to the calculation of the stress intensity factors in 3D by the operator `CALC_G` [U4.82.03] with the option `CALC_K_G`.

The operator produces a concept of the type `fiss_xfem`.

## 2 Syntax

```
[fiss_xfem] = DEFI_FISS_XFEM (  
  
  ◆ GRID = my, [grid]  
  
# Definition of a grid to be associated with the crack  
  ◇ / GRID_GRILLE= magri, [grid]  
  / FISS_GRILLE = fiss_orig, [fiss_xfem]  
  
# choice of a crack or an interface  
  ◇ TYPE_DISCONTINUITE = / 'CRACK', [DEFECT]  
  / 'COHESIVE'  
  / 'INTERFACE'  
  
# Definition of the crack or the interface  
  
  ◆ DEFI_FISS = _F (  
  
# Either one defines the group of meshes of a lip and the group of meshes of the bottom of crack:  
  ◆ / ◆ GROUP_MA_FISS = grma, [l_gr_maille]  
  ◇ GROUP_MA_FOND = grma, [l_gr_maille]  
  
# Either two analytical functions are defined; to characterize the surface of the crack, and the other  
to characterize the bottom of crack:  
  / ◆ FONC_LN = fonc, [function]  
  ◇ FONC_LT = fonc, [function]  
  
# For a crack of the COHESIVE type:  
  / ◆ GROUP_MA_BORD = group_ma, [l_gr_maille]  
  
# Either one uses the shape of crack or of predefined interface and one gives the geometrical  
characteristics:  
  
  / ◆ FORM_FISS = forms, [K8]  
  
  # If forms = 'ELLIPSE'  
  ◆ DEMI_GRAND_AXE = has, [R]  
  ◆ DEMI_PETIT_AXE = B, [R]  
  ◆ CENTER = (x0, y0, z0), [1_R]  
  ◆ VECT_X = (vxx, vxy, vxz), [1_R]  
  ◆ VECT_Y = (vyx, vyy, vyz), [1_R]  
  ◇ COTE_FISS = / 'IN', [DEFECT]  
  / 'OUT',  
  
  # If forms = 'RIGHT-ANGLED'  
  ◆ DEMI_GRAND_AXE = has, [R]  
  ◆ DEMI_PETIT_AXE = B, [R]  
  ◇ RAYON_CONGE = / 0 [DEFECT]  
  / R, [R]  
  ◆ CENTER = (x0, y0, z0), [1_R]  
  ◆ VECT_X = (vxx, vxy, vxz), [1_R]  
  ◆ VECT_Y = (vyx, vyy, vyz), [1_R]  
  ◇ COTE_FISS = / 'IN', [DEFECT]  
  / 'OUT',  
  
  # If forms = 'CYLINDER'
```

```

♦ DEMI_GRAND_AXE = has, [R]
♦ DEMI_PETIT_AXE = B, [R]
♦ CENTER = (x0, y0, z0), [l_R]
♦ VECT_X = (vxx, vxy, vxz), [l_R]
♦ VECT_Y = (vyx, vyy, vyz), [l_R]

# If forms = 'NOTCH'
♦ DEMI_LONGUEUR = has,
♦ RAYON_CONGE = R,
♦ CENTER = (x0, y0, z0),
♦ VECT_X = (vxx, vxy, vxz),
♦ VECT_Y = (vyx, vyy, vyz),

# If forms = 'DEMI_PLAN'
♦ PFON = (x0, y0, z0), [l_R]
♦ NORMAL = (vnx, vny, vnz), [l_R]
♦ DTAN = (vtx, vty, vtz), [l_R]

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

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

# If forms = 'RIGHT'
♦ NOT = (x0, y0, z0), [l_R]
♦ DTAN = (vtx, vty, vtz), [l_R]

),

```

# Either one reads again fields with the already existing nodes

```

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

```

# Definition of the zone of enrichment

```

◇ GROUP_MA_ENRI = grma, [l_gr_maille]
◇ CHAM_DISCONTINUITE = 'DEPL', [DEFECT]
◇ TYPE_ENRI_FOND = / 'TOPOLOGICAL' [DEFECT]
                  / 'GEOMETRICAL'

```

```

# If TYPE_ENRI_FOND = 'GEOMETRICAL'
◇ RAYON_ENRI = Renri, [R]

```

```

# If RAYON_ENRI = Nun
◇ NB_COUCHES = / 4, [DEFECT]
              = / NR, [I]

```

# Definition of a junction on the crack

# If (GRID\_GRILLE = Nun and FISS\_GRILLE = Nun),

```

◇ JUNCTION = _F (
    ♦ CRACK = (fiss1, fiss2,..., fissi), [l_fiss_xfem]
    ♦ NOT = (Pox, Poy, Poz), [l_R]

```

## # Impression of information

```
◇ INFORMATION = / 1, [DEFECT]
                / 2,
                / 3,
                )
```

## 3 Operands

### 3.1 Operand GRID

◆ GRID = my

my : name of the grid on which one will define the crack. This grid is obligatorily a grid 2D or 3D.

**Attention, quadratic elements X-FEM miss robustness (especially in 3D) one advises to use linear elements X-FEM. It is thus preferable that my that is to say a linear grid.**

### 3.2 Definition of a grid to be associated with the crack

An auxiliary grid can be associated with the crack:

- maybe by giving a grid roasts (see §3.2.1),
- maybe by giving a crack already associated with a preceding grid (see §3.2.2),

In the case of use of the method upwind in PROPA\_FISS (Doc. U4.82.11), its use is almost obligatory. This keyword is used typically following a refinement by Lobster of the grid where the crack was defined.

#### 3.2.1 Operand MAILLAGE\_GRILLE

◇ GRID\_GRILLE = mygri [grid]

magri : name of the grid grid to be associated with the crack

This operand is used to associate the grid roasts magri with the crack. The two functions of level (level set) and their gradients are calculated at the same time on the grid my given by GRID (see §3.1) and on the grid magri according to the method chosen by DEFI\_FISS (see §3.4) and they are stored with the crack. Thus several cracks can be defined on the same grid magri.

Grid magri is created by the operator DEFI\_GRILLE [U4.24.02].

The auxiliary grid used must extend only on the zone envisaged from propagation from the crack in order to correctly be able to represent the bottom of the crack during all the simulation of the propagation. It can be partly outside the grid my (keyword GRID, to see §3.1) and no report exists between this grid and that of the grid magri. In the presence of several cracks and with a use of different grid for each one, the grids are independent and can be superimposed partially or completely. In this case one could also use only one roasts for all the cracks.

#### 3.2.2 Operand FISS\_GRILLE

If one defined a crack on a grid with an associated grid and where one wants to redefine it on another grid without changing grid, one can save computing time while avoiding recomputing the functions of level on the grid. Indeed, in this case the functions of level defined on the grid do not change.

That is always the case when one uses Lobster on the grid where the crack was defined because the grid is not affected by the Lobster modifications and the functions of level defined initially do not change. One can thus redefine the same crack on the grid refined with the same grid associated by calculating the functions with level only on the grid and by keeping them with identical on the grid.

◇ FISS\_GRILLE = fiss\_orig, [fiss\_xfem]

The crack is specified fiss\_orig who was definite front. The grid already present is associated with the new crack and the fields with the calculated nodes which define the functions of level and their gradients are recopied on the grid for the new crack. That is equivalent has to define the crack fiss\_orig on the new grid while always using the same grid.

## Note:

No checking can be made on coherence between the crack which will be defined by DEFI\_FISS (see § 3.4) and that which was defined on the grid of *fiss\_orig*.

## 3.3 Operand TYPE\_DISCONTINUITE

```
◇ TYPE_DISCONTINUITE = / 'CRACK', [DEFECT]
                       / 'COHESIVE'
                       / 'INTERFACE'
```

By default, one defines a crack. It is however possible to define an interface (to treat the case of 2 solids in contact, or under-thicknesses,...).

If TYPE\_DISCONTINUITE = 'CRACK', the structure of data produced is enriched by two named tables 'FOND\_FISS' and 'NB\_FOND\_FISS' recoverable by using the order RECU\_TABLE. The table 'NB\_FOND\_FISS' only one value contains corresponding to the number of funds of crack. The table thus contains only one column (named NUMBER) and only one line. The table 'FOND\_FISS' contains the list of the coordinates of the points of all the bottom of crack. This table 6 columns:

- the first column ( 'NUMÉRIQUE\_FOND' ) corresponds to the number of the bottom of crack,
- the following column ( 'NUM\_PT' ) corresponds to the number of the point of the bottom of crack, this number is local at the current bottom of crack,
- the following column ( 'ABSC\_CURV' ) corresponds to the value of the curvilinear X-coordinate along the current bottom of crack,
- 3 following columns ( 'COOR\_X' , 'COOR\_Y' and 'COOR\_Z' ) correspond to the coordinates of the point.

The type of discontinuity COHESIVE allows to have information on the face of propagation (of the fact data previously mentioned and the second function of level) in order to transmit them to algorithms of propagation or follow-up of fissured surface. On the other hand, contrary to the type CRACK , it does not contain related information with an enrichment at a peak, since this last does not take place to be in the presence of forces of cohesion. Indeed, it is important to note that for the type of discontinuity COHESIVE , the face of propagation does not coincide with the end of the Heaviside enrichment, which can cross all the structure. The initial face must thus be given independently, by the keyword GROUP\_MA\_BORD .

## 3.4 Keyword DEFI\_FISS

◆ DEFI\_FISS = \_F

The keyword factor DEFI\_FISS allows to define a crack or an interface in 4 different ways:

- maybe starting from two groups of meshes (see §3.4.1);
- maybe starting from two analytical functions (see §3.4.2);
- maybe starting from a catalogue of preset forms (see §3.4.3);
- maybe starting from existing fields (see §3.4.4).

### 3.4.1 Definition of the crack by groups of meshes

If there is a crack (or an interface) already with a grid, then one can define the crack by giving two groups meshes: GROUP\_MA\_FISS and GROUP\_MA\_FOND. This grid can be independent of the grid of the structure.

```
/ ◆ GROUP_MA_FISS = grma
with grma single group of meshes.
```

This group of meshes corresponds to only one of the lips of the crack (lower lip or higher). This group of meshes must be directed. In the case of lips which would not be stuck perfectly, the fact of privileging a side will have a light influence on the basis of local bottom of crack, all the more large influence as the

angle between the two lips is important. For the case 2D, the program sorts the meshes of GROUP\_MA\_FISS in order to have a group of contiguous meshes whatever the starting group. It is thus not necessary to give a group of meshes which are followed.

◇ GROUP\_MA\_FOND = grma

with grma single group of meshes.

This operand is used to define the bottom of crack. For the interfaces, GROUP\_MA\_FOND is not to inform.

In 2D, GROUP\_MA\_FOND is a group of meshes not.

## 3.4.2 Definition of the crack by analytical functions

The principal interest of this operator is the possibility of defining a crack without this one not being inevitably with a grid. In this case, one defines the crack using two functions of levels (level sets). The first level set (known as level set "normal") is that which makes it possible to characterize the surface of the crack. One thus informs FONC\_LN with a real function of  $X$ ,  $Y$  and  $Z$  defined as a preliminary by the operator FORMULA. The surface of the crack will be then the Iso-zero of this function. The second level set (known as level set "tangent") is that which makes it possible to characterize the bottom of crack. One thus informs FONC\_LT with a real function of  $X$ ,  $Y$  and  $Z$  defined as a preliminary by the operator FORMULA. The trace of the Iso-zero of FONC\_LT in the plan of cracking is the bottom of crack. The points of the crack are then characterized by  $\text{FONC\_LN} = 0$  and  $\text{FONC\_LT} < 0$ , whereas the bottom of crack is characterized by  $\text{FONC\_LN} = 0$  and  $\text{FONC\_LT} = 0$ .

For the interfaces, FONC\_LT is not to inform.

/ ◆ FONC\_LN = fonc,

with fonc a function or a formula defined before.

◇ FONC\_LT = fonc,

with fonc a function or a formula defined before.

### Note:

*For a crack, it is paramount that the level sets are the true functions distances, one cannot thus define an elliptic crack by:*

*LN=FORMULE (NOM\_PARA= ('X', 'Y', 'Z'), VALE=' Z-H')*

*LT=FORMULE (NOM\_PARA= ('X', 'Y', 'Z'), VALE=' (X/a) <sup>2</sup> + (Y/b) <sup>2</sup> - 1')*

*where has and B would be the equatorial radius and half-small axis of the ellipse. Indeed, the function LT is not the distance to the ellipse. The function distance to an ellipse is not expressed using usual functions. For introduced such a crack, the 3 should be used <sup>ème</sup> method: predefined forms.*

*For the interfaces, any function level set is possible, and it is not useful to introduce the true function outdistances.*

### Note:

*To date nothing would prohibit to have a level set which defines several closed funds of crack (for example several cracks of the type "Penny-shaped"). Under these conditions, a call to DEFI\_FISS\_XFEM could then define several cracks. This possibility is prohibited: each order DEFI\_FISS\_XFEM only one defines fissures. It is thus not possible to define several closed funds OR a bottom closed more one or more open funds.*

### Note:



Attention with the sign of  $FONC\_LT$  , always to remember that the crack is defined side  $FONC\_LT < 0$ .

### 3.4.3 Definition of the crack by a catalogue of form predefined

There exists a catalogue of predefined forms of crack/interface. The choice of the form depends on the dimension of the grid (2D or 3D). The possible choices are indexed in Table 3.4.3-1.

	Cracks	Interfaces
2D	SEGMENT, DEMI_DROITE	RIGHT-HAND SIDE , ELLIPSE , RECTANGLE , NOTCH
3D	ELLIPSE, RECTANGLE, CYLINDER, DEMI_PLAN	

**Table 3.4.3-1: Catalogue predefined forms**

For each form, should be given the specific geometrical characteristics.

/ ♦ FORM\_FISS = forms,

#### 3.4.3.1 FORM\_FISS = 'ELLIPSE'

Case of an elliptic plane crack in 3D or of an elliptic inclusion of form in 2D. It is necessary to inform the length of the equatorial radius by the keyword DEMI\_GRAND\_AXE and the length of the half-small axis by the keyword DEMI\_PETIT\_AXE. To characterize the plan of the ellipse, one must inform the center of the ellipse by the keyword CENTER and locates it plan by the keyword VECT\_X and VECT\_Y. The vector VECT\_X to the axis of the ellipse the length corresponds which the length is defined  $a$  data under the keyword DEMI\_GRAND\_AXE, and the vector VECT\_Y to the axis of the ellipse the length corresponds which the length is defined  $b$  data under the keyword DEMI\_PETIT\_AXE.

Note:

That one chooses  $0 < b \leq a$  or  $0 < a \leq b$  , VECT\_X remain always associated with DEMI\_GRAND\_AXE and VECT\_Y remain always associated with DEMI\_PETIT\_AXE .

In the case of a crack, one can moreover specify of which with dimensions is the crack (inside or outside the ellipse) by the keyword COTE\_FISS . Figure 3.4.3.1-1 represent an elliptic crack/inclusion, of equatorial radius  $a$  , of half-small axis  $b$  and of center  $C$  . The vector  $\vec{e}_1$  corresponds to the keyword VECT\_X and the vector  $\vec{e}_2$  corresponds to the keyword VECT\_Y .

♦ DEMI\_GRAND\_AXE = has,  
♦ DEMI\_PETIT\_AXE = B,  
♦ CENTER = (Cx, Cy, Cz),  
♦ VECT\_X = (e1x, e1y, e1z),  
♦ VECT\_Y = (e2x, e2y, e2z),  
♦ COTE\_FISS = / 'IN',  
/ 'OUT',

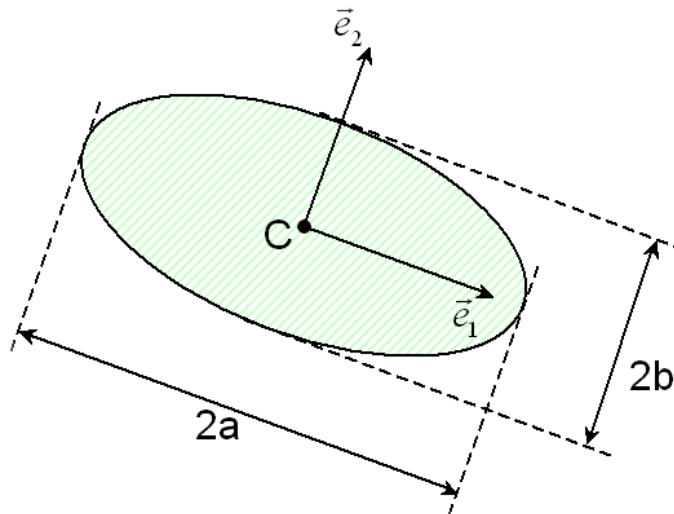


Figure 3.4.3.1-1: Diagram of an elliptic crack/inclusion

### 3.4.3.2 FORM\_FISS = 'RIGHT-ANGLED'

Case of a crack planes of rectangular form in 3D, or a rectangular inclusion of form in 2D, with possibility of corners rounded. It is a form similar to the ellipse, it is necessary to inform the same keyword: the value of the equatorial radius (or half-length) by the keyword `DEMI_GRAND_AXE` and the value half-small axis (or half-width) by the keyword `DEMI_PETIT_AXE`. To characterize the plan of the rectangle, one must inform the center of the rectangle by the keyword `CENTER` and locates it plan. The equatorial radius corresponds to the vector `VECT_X` and the half-small axis with the vector `VECT_Y`. Moreover, in the case of a crack, one can specify of which with dimensions the crack (inside or outside the rectangle) by the keyword is `COTE_FISS`. The only keyword in more compared to the form ellipse is that being used to give the value of the ray of the leave for the corners rounded (`RAYON_CONGE`). Figure 3.4.3.2-1 represent a crack/rectangular inclusion of form to corners rounded, equatorial radius  $a$ , of half-small axis  $b$  and of center  $C$ . The vector  $\vec{e}_1$  corresponds to the keyword `VECT_X` and the vector  $\vec{e}_2$  corresponds to the keyword `VECT_Y`. The ray of leave of the corners rounded is  $r$ . By default the ray of leave is null.

```

♦ DEMI_GRAND_AXE = has,
♦ DEMI_PETIT_AXE = B,
♦ RAYON_CONGE   = R,
♦ CENTER        = (Cx, Cy, Cz),
♦ VECT_X        = (e1x, e1y, e1z),
♦ VECT_Y        = (e2x, e2y, e2z),
♦ COTE_FISS     = / 'IN',
                / 'OUT',

```

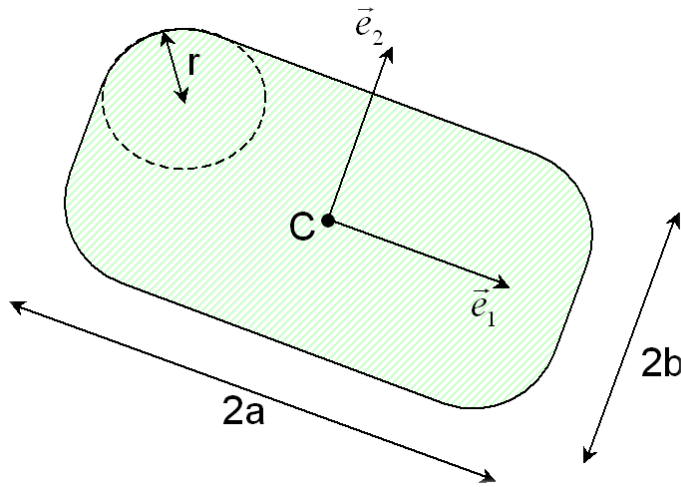


Figure 3.4.3.2-1: diagram of a rectangular crack/inclusion with corners rounded

### 3.4.3.3 FORM\_FISS = 'CYLINDER'

Cage of a cylindrical three-dimensional crack with elliptic section. To characterize the bottom (elliptic) crack, it is necessary to inform the length of the equatorial radius by the keyword `DEMI_GRAND_AXE` and the length of the half-small axis by the keyword `DEMI_PETIT_AXE`. One must also inform the center of the ellipse by the keyword `CENTER` as well as the reference mark of the plan ( `VECT_X`, `VECT_Y` ). Note: the equatorial radius corresponds to the vector `VECT_X` and the half-small axis with the vector `VECT_Y`. The generator of the cylinder and the direction of potential propagation of the crack are given by the vector product of `VECT_X` by `VECT_Y`. This cylinder is semi-infinite along this axis. Figure 3.4.3.3-1 illustrate a crack of the type rolls equatorial radius  $a$ , of half-small axis  $b$  and of center  $C$ . The vector  $\vec{e}_1$  corresponds to the keyword `VECT_X` and the vector  $\vec{e}_2$  corresponds to the keyword `VECT_Y`.

- ◆ `DEMI_GRAND_AXE` = has,
- ◆ `DEMI_PETIT_AXE` = B,
- ◆ `CENTER` = (x0, y0, z0),
- ◆ `VECT_X` = (vxx, vxy, vxz),
- ◆ `VECT_Y` = (vyx, vyy, vyz),

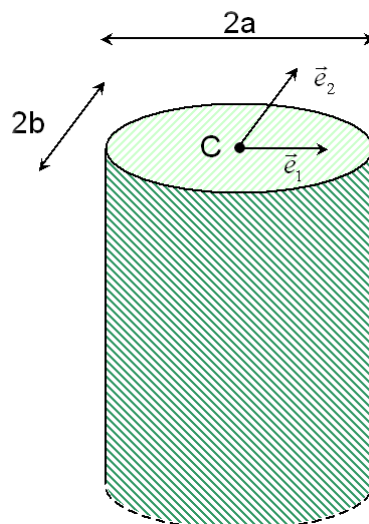


Figure 3.4.3.3-1: Diagram of a crack rolls

### 3.4.3.4 FORM\_FISS = 'NOTCH'

Case of an inclusion of the type notches in 2D. It is a typical case of inclusion of the right-angled type to corners rounded in where the half-height is equal to the ray of leave. To define the notch it is necessary to inform the value of the half-length by the keyword `DEMI_LONGUEUR`, the value of the ray of the leave by the keyword `RAYON_CONGE` and notch by the keyword `CENTER`. Moreover, the direction of the notch is given by the keywords `VECT_X` and `VECT_Y`. Figure 3.4.3.4-1 represent an inclusion of the type notches, of half-length  $a$ , of ray of leave  $r$  and of center  $C$ . The vector  $\vec{e}_1$  corresponds to the keyword `VECT_X` and the vector  $\vec{e}_2$  corresponds to the keyword `VECT_Y`.

- ◆ `DEMI_LONGUEUR` = has,
- ◆ `RAYON_CONGE` = R,
- ◆ `CENTER` = (Cx, Cy, Cz),
- ◆ `VECT_X` = (e1x, e1y, e1z),
- ◆ `VECT_Y` = (e2x, e2y, e2z),

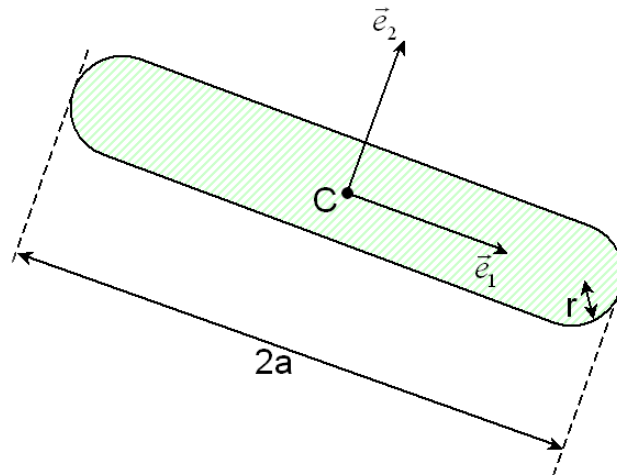


Figure 3.4.3.4-1: diagram of a rectangular crack/inclusion with corners rounded

### 3.4.3.5 FORM\_FISS = 'DEMI\_PLAN'

Case of a crack defined by a half-plane. `PFON` indicate a point of the bottom of crack. `NORMAL` the normal defines in the half-plane of the crack and `DTAN` provides a vector in the plan of crack, orthogonal at the bottom and directed in the direction of potential propagation. Figure 3.4.3.5-1 represent a crack of half-plane defined by  $P$  a point of the bottom, the vector  $\vec{e}_1$  corresponds to the keyword `DTAN` and the vector  $\vec{e}_2$  corresponds to the keyword `NORMAL`.

- ◆ `PFON` = (Px, Py, Pz),
- ◆ `DTAN` = (e1x, e1y, e1z),
- ◆ `NORMAL` = (e2x, e2y, e2z),

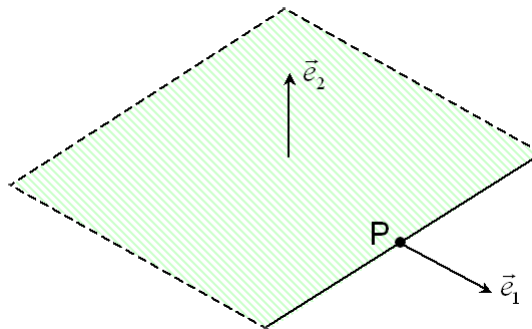


Figure 3.4.3.5-1: diagram of a crack half-plane

### 3.4.3.6 FORM\_FISS = 'SEGMENT'

ThatS of a crack 2D defines by a segment. PFON\_ORIG and PFON\_EXTR the ends of the segment indicate. Figure 3.4.3.6-1 present a crack 2D of type segment where P1 indicate the point of origin PFON\_ORIG and P2 the point end PFON\_EXTR .

- ◆ PFON\_ORIG = (P1x, P1y, P1z),
- ◆ PFON\_EXTR = (P2x, P2y, P2z),

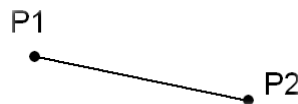


Figure 3.4.3.6-1: diagram of a crack 2d of type segment

### 3.4.3.7 FORM\_FISS = 'DEMI\_DROITE'

Case of a crack 2D defined by a half-line. PFON indicate the point of the bottom of crack. DTAN corresponds to a directing vector of the half-line directed in the direction of potential propagation. Figure 3.4.3.7-1 present a crack 2D of type half-line for which P is the point of the bottom (PFON) and of directing vector  $\vec{e}_1$  (DTAN).

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

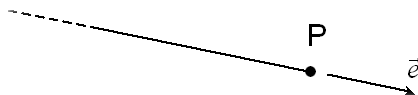


Figure 3.4.3.7-1: diagram of a crack 2D of type half-line

if forms = RIGHT

Case of a right interface in a definite structure 2D. NOT indicate a point of the interface. DTAN corresponds to a directing vector of the right-hand side. The orientation is not important.

- ◆ NOT = (x0, y0, z0),
- ◆ DTAN = (vtx, vty, vtz),

## 3.4.4 Definition of the crack by second reading of fields to the already existing nodes

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Copyright 2017 EDF R&D - Licensed under the terms of the GNU FDL (<http://www.gnu.org/copyleft/fdl.html>)

```

/   ◆ CHAM_NO_LSN      =  ch_lsn,           [cham_no]
     ◇ CHAM_NO_LST     =  ch_lst,           [cham_no]
  
```

Fields `ch_lsn` and `ch_lst` are fields with the already existing nodes. They must be created on the same grid as the grid `my` given as starter under the keyword `GRID` (cf §3.1).

These nodal fields can be obtained successively by the reading of a table (`LIRE_TABLE`), then by the creation of the field by extraction of the table (`CREA_CHAMP/OPERATION='EXTR'`).

They could also be generated in a stage of refinement of grid (`MACR_ADAP_MAIL/MAJ_CHAM`).

### 3.4.5 Definition of the initial face of propagation for a cohesive crack

```

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

For a study of propagation in the presence of cohesive forces, it is necessary to give the face of initial propagation, i.e. the line from which cracking starts. It is generally of a corner of the structure, or a bottom of notch, or more generally about a line located on an edge. It is given either in the form of a group of mesh 1D if it is with a grid, or in the form of a group of meshes 2D of edge, whose intersection with the level-set normal will provide the initial face of propagation.

### 3.4.6 Examples

- if one wishes to define a bar fissured right through in his plan in the height  $z = H$  (see [Figure 3.2-a], on which the crack is hatched):

```

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

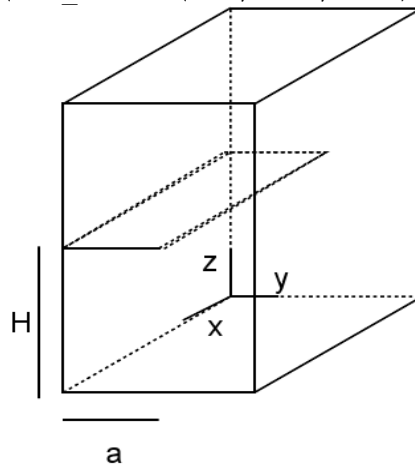


Figure 3.2-a: Fissured bar

- if one wishes to define an elliptic crack in an infinite solid mass in a plan in the height  $z = H$  (see [Figure 3.2-b], on which the crack is hatched):

```

DEFI_FISS = _F ( FORM_FISS      = 'ELLIPSE',
                 DEMI_GRAND_AXE  = has,
                 DEMI_PETIT_AXE  = B,
                 CENTER          = (0, 0, H),
                 VECT_X          = (1, 0.0),
                 VECT_Y          = (0, 1.0),
                 COTE_FISS       = 'IN', ),
  
```

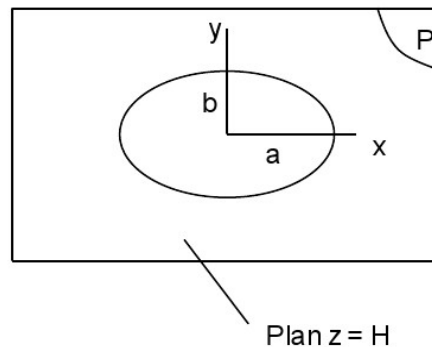


Figure 3.2-b: Elliptic crack

## 3.5 Operand GROUP\_MA\_ENRI

◇ GROUP\_MA\_ENRI = grma, [l\_gr\_maille]

This operand makes it possible to know the zone on which the procedure of enrichment will be carried out. The nodes nouveau riches must belong to this group of mesh.

If GROUP\_MA\_ENRI is not informed, then the procedure of enrichment will be carried out on all the grid my given as starter under the keyword GRID (cf §3.1)

A use of this keyword thus makes it possible to delimit the zone of enrichment, therefore crack. The following example presents an elliptic crack of form in a hollow roll. The crack is defined by the data of groups of meshes (keywords GROUP\_MA\_FISS and GROUP\_MA\_FOND). Two level sets are deduced from it, by considering a prolongation of the bottom of crack by continuity. If the zone of enrichment contains all the structure, that led then to a crack in the right part of the cylinder, which is not at all what one wishes. To solve this problem, it is enough to limit the zone of enrichment and to inform the keyword GROUP\_MA\_ENRI with the meshes of the left half of the cylinder for example.

## 3.6 Operand CHAM\_DISCONTINUITÉ

◇ CHAM\_DISCONTINUITÉ = 'DEPL', [DEFECT]

CE keyword makes it possible to choose the type of fields to be enriched. For the moment, only the field of displacement can be enriched.

## 3.7 Operands TYPE\_ENRI\_FOND, RAYON\_ENRI, NB\_COUCHES

◇ TYPE\_ENRI\_FOND = / 'TOPOLOGICAL' [DEFECT]  
/ 'GEOMETRICAL'

CE keyword makes it possible to choose the type of enrichment in bottom of crack. Topological enrichment is activated (only one sleeps) if TYPE\_ENRI\_FOND = 'TOPOLOGIQUE' (by default). Geometrical enrichment is activated (several layers) if TYPE\_ENRI\_FOND = 'GEOMETRIQUE' (for more information concerning the effects of enrichment on quality of the results, to see [R7.02.12] and [U2.05.02]). Geometrical enrichment is controlled by the keywords RAYON\_ENRI and NB\_COUCHES. This kind of enrichment is not advised at all with quadratic grids (under penalty of null pivot).

◇ RAYON\_ENRI = Renri, [R]

This keyword can be indicated only if `TYPE_ENRI_FOND = 'GEOMETRICAL'` . If this operand is indicated, it makes it possible to specify a geometrical criterion `Renri` such as all the nodes whose distance to the bottom of crack is lower than this criterion will be enriched by the singular functions. This enrichment is called "geometrical". Studies showed that such an enrichment largely improved the precision of calculations, in particular when the grid is refined in bottom of crack. It is advised to choose `Renri` approximately equal to 1/10 the length of the crack. The disadvantage is that a too large ray of enrichment can deteriorate the conditioning of the system and generate difficulties of convergence.

To avoid the nuisances caused by a too large ray of enrichment, one proposes to define the zone to be enriched by stipulating the number of layers of elements to be enriched around the bottom by crack instead of ray by enrichment.

```
# If RAYON_ENRI = Nun
◇ NB_COUCHES = / 4, [DEFECT]
/ NR, [I]
```

If `RAYON_ENRI` is not well informed (and only `TYPE_ENRI_FOND = 'GEOMETRIQUE'`), then `NB_COUCHES` is the number of layers of elements to be enriched (4 by defaults). Generally, 7 layers is the limit not to exceed to avoid the worthless pivots at the time of the resolution.

## 3.8 Operand JUNCTION

```
# If MAILLAGE_GRILLE = Nun,
◇ JUNCTION = _F (
    ◆ CRACK = (fiss1, fiss2, ..., fissi), [l_fiss_xfem]
    ◆ NOT = (Px, Py, Pz), [l_R]
```

This operand makes it possible to define one or more junctions between a crack and other already existing cracks. These other cracks are then regarded as cracks mothers: they should have been already beforehand defined by calls to `DEFI_FISS_XFEM`.

One informs a list of cracks mothers in the keyword `CRACK` . All these cracks mothers will be taken into account at the time of the definition of the current crack . The keyword `NOT` must be given in field where it is wished that the crack be defined. An example is provided figure 3.7-a. One considers on this figure that crack 3 was defined at the beginning to cut the field vertically. The keyword is introduced `JUNCTION` in order to connect it on cracks 1 and 2. One can obtain the three configurations figure according to the choice of the position given in the keyword `NOT` .

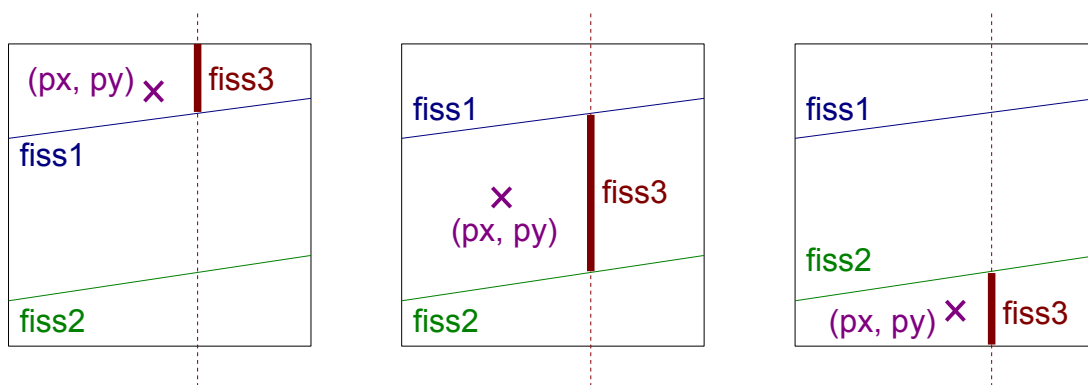


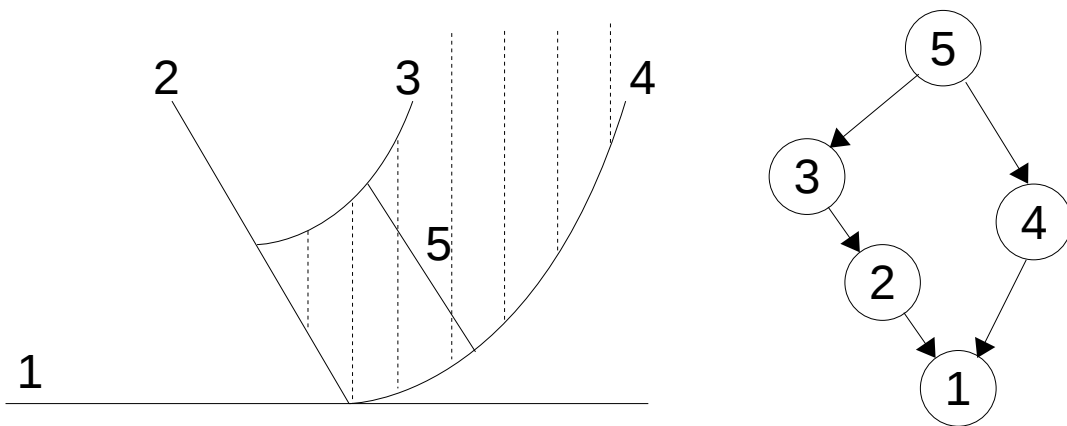
Figure 3.8-1 : Example of definition of the crack `fiss3`, for the cracks mothers `fiss1` and `fiss2` , according to the point  $(px, py)$  chosen.

It should well be paid attention when the operand is used `JUNCTION`: the crack then will not be only any more represented by its level sets normal and tangent, but it will also depend on the cracks



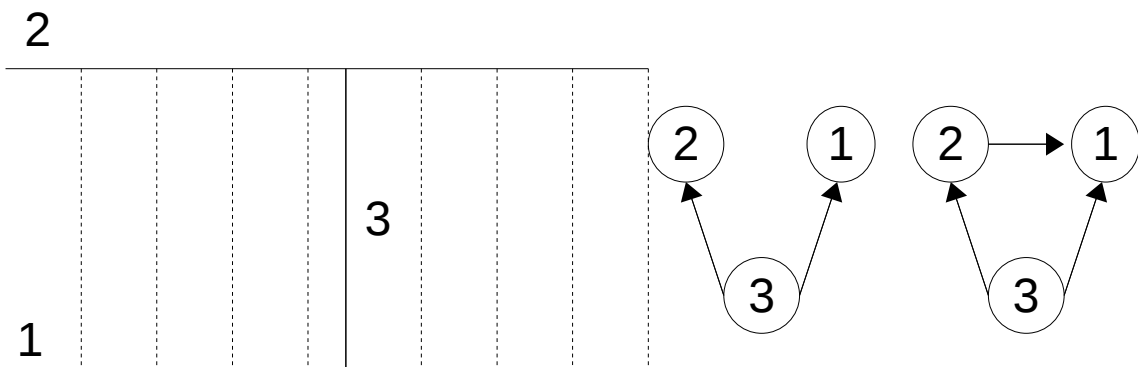
mothers. That can pose problem if one wishes to use operators like `RAFF_XFEM`, which take into account that the concept of level sets like mode of definition of the crack. In the same way the chart of the Iso-zero of the level sets is not sufficient, it is preferable to pass by the operations of postprocessing via the operator `POST_MAIL_XFEM`.

An example of configuration is presented Figure 3.8-2. On this example one identifies the cracks on which another crack comes to connect. Thus for the crack  $N=3$ , one has  $K=2$  where  $K$  is the whole of the cracks on which 3 connects. In the same way for the crack  $N=5$ , one deduces  $K=[3,4]$ . Crack 5 is thus defined starting from the operand junction and of two cracks 3 and 4. Crack 3 is itself before definite starting from the operand junction acting on crack 2, itself definite starting from the operand junction acting on crack 1. The tree of hierarchy of Figure 3.8-2 we is thus used to establish the order of definition of the various cracks (2 from 1.3 from 2.4 from 1 and 5 from 3 and 4).



**Figure 3.8-2 : Network of cracks on the left, tree of hierarchy on the right indicating which connects and allowing to establish the relations of order for the definitions of crack and the use of the operand JUNCTION.**

The operator junction allows to establish a relation of order for a mesh which sees  $N$  cracks. One will distinguish thus on Figure 3.8-3 two trees of hierarchy of right-hand side. In the tree on the right, cracks 1 and 2 are sufficiently close so that a mesh seeing crack 3, sees also cracks 1 and 2: one thus decides to connect crack 3 on cracks 2 and 1, but to as use the operand junction to indicate as crack 2 is connected to crack 1. For the other tree, a mesh sees with more the 2 cracks (no mesh sees then at the same time cracks 1.2 and 3) and in this case crack 3 connects simply on the cracks 2 and 1 which are not connected any more between them.



**Figure 3.8-3 : Network of cracks on the left, trees of hierarchy on the right.**

The tree of hierarchy on the right indicates that crack 2 cannot be defined independently of crack 1 because too near to the latter (a mesh seeing crack 3 then sees also cracks 1 and 2).

## 3.9 Operand INFORMATION

- / 1 : impression on the file 'MESSAGE'
  - 1) amongst points of the bottom of crack;
  - 2) amongst funds of crack;
  - 3) ordered list of the coordinates of the points of the bottom like their curvilinear X-coordinate, for each bottom of crack.
- / 2 : same impressions as 1 + impressions
  - 1) amongst level sets readjusted;
  - 2) amongst meshes of the zone of the crack;
  - 3) amongst meshes containing the bottom of crack
  - 4) amongst meshes of the Heaviside type, Ace-tip and Heaviside Ace-tip.
- / 3 : even impression that 2 + impressions
  - 5) stages of calculation;
  - 6) fields level sets and their gradients;
  - 7) statute of the nodes;
  - 8) numbers of the enriched meshes.

## 4 Examples

### 4.1 With analytical formulas

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

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

### 4.2 With groups of meshes

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

### 4.3 With the catalogue of preset forms

Case of a cylindrical crack:

```
FISS=DEFI_FISS_XFEM (GRID = MAILLAIN,
                    DEFI_FISS=_F (
                        FORM_FISS = 'CYLINDER',
                        DEMI_GRAND_AXE = R,
                        DEMI_PETIT_AXE = R,
                        CENTER = (0. , l_fiss, 0.),
                        VECT_X = (0. , 0. , 1.),
                        VECT_Y = (1. , 0. , 0.),
                    ),
                    GROUP_MA_ENRI=' VOL',
                    INFO=1,
                    );
```

Case of a crack defined by a half-plane:

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