

---

## Operator DEFI\_FOND\_FISS

---

### 1 Goal

---

To define the geometric standards relative to a crack, in particular the bottom of crack and the upper lips and lower of this crack, in a grid 2D or 3D.

The definition of the bottom of the crack can be done starting from entities (nodes or meshes) or starting from groups of entities. The nodes can be ordered in the direction of the increasing curvilinear X-coordinates. If it is not the case and if the bottom of crack is given by a list of meshes or groups of meshes, the operator will order the nodes with the help of the definition of a node origin.

Two initial configurations of the upper lips and lower are taken into account for the definition of the direction of propagation and the lips.

This operator creates a concept of the type `fond_fiss` who is usable by the operators `CALC_THETA` [U4.82.02], `CALC_G` [U4.82.03] and `POST_K1_K2_K3` [U4.82.05].

## 2 Syntax

```
FF [fond_fiss] = DEFI_FOND_FISS (

♦ GRID = my , [grid]
♦ INFORMATION = / 1,
  [DEFECT]
  / 2,

# For the definition of a bottom of crack
♦ / FOND_FISS = _F (

# A kind of bottom is defined

♦ TYPE_FOND = / 'OPENED', [DEFECT]
              / 'FIRM',
              / 'INF',
              / 'SUP',

# Either one defines one list of already ordered nodes

♦ / GROUP_NO = lgrno, [l_gr_noeud]
  / NODE = lno, [l_noeud]

# Is one defines a list of meshes whose nodes can be already ordered or not

♦ / GROUP_MA = lgrma, [l_gr_maille]
  / MESH = lma, [l_maille]

# In this last case it is then necessary to define an origin of the X-coordinates

♦ / NOEUD_ORIG = No , [l_noeud]
  / GROUP_NO_ORIG= grno, [l_gr_noeud]

# In the case or GROUP_MA or MESH is defined and if the bottom is not closed, one can then
define an end of the X-coordinates
♦ / NOEUD_EXTR = No , [l_noeud]
  / GROUP_NO_EXTR= grno, [l_gr_noeud]

# In the case or GROUP_MA or MESH is defined and if the bottom is closed, one can then define an
end of the X-coordinates
♦ / MAILLE_ORIG = my , [l_maille]
  / GROUP_MA_ORIG= grma, [l_gr_maille]

♦ / DTAN_ORIG = (Drug addict, Toy, Toz),
[l_R]
  DTAN_EXTR = (Tex, Tey, Tez), [l_R]
  / VECT_GRNO_ORIG = lgrno, [l_gr_noeud]
  VECT_GRNO_EXTR = lgrno, [l_gr_noeud]
),

# For the definition of the initial configuration of the lips
♦ / CONFIG_INIT = / 'COLLEE', [DEFECT]
  / 'DECOLLEE',

# Symmetry of the structure
♦ / SYME = / 'NOT', [DEFECT]
```

```

/ 'YES',
# Definition of the upper lips
◇ LEVRE_SUP = _F (
  ◆ / GROUP_MA = lgrma, [l_gr_maille]
    / MESH = lma, [l_maille]
  ),

# If the structure is not symmetrical, the lower lip should be defined

◆ LEVRE_INF = _F (
  ◆ / GROUP_MA = lgrma, [l_gr_maille]
    / MESH = lma, [l_maille]

# If the initial configuration is DECOLLEE : definition of the normal
◆ NORMAL = (Nx, Ny, Nz) [l_R]

◇ PREC_NORM = / 1.E-1, [DEFECT]
              / epsi, [R]

)
```

## 3 Operands

### 3.1 Operand GRID

- ♦ GRID = my

Name of the grid on which one will define the bottom of crack and the lips.

### 3.2 Description of the bottom of crack

The bottom of crack is defined by the whole of the ordered nodes of the bottom of crack. If none the keywords NOEUD\_ORIG or GROUP\_NO\_ORIG is not defined, the order of declaration of these nodes, via list of nodes or of meshes segments, the direction of course of the curvilinear X-coordinate of the bottom of crack will define. Load is thus left to the user compose an ordered list, within the meaning of the connectivity of the grid, by increasing curvilinear X-coordinate.

It is also possible to provide a list of meshes segments without worrying about the order. The data of a node origin, provided that it corresponds well at an end of the way defined by the meshes segments, then makes it possible to order the list of nodes.

In addition, EN 3D, for a knot slip of the bottom of crack, the direction of propagation is defined as being the average of the normals to the meshes segments of the bottom of crack on its left and its right-hand side. For the nodes ends, the normal is calculated starting from one only mesh, and can thus be less precise.

The code thus a correction of this normal envisages by taking of account the edges of the structure. However, keywords DTAN\_ORIG and DTAN\_EXTR, optional, allow the user to directly impose the directions of propagation at the origin and the end of the bottom.

In order to determine the direction of the vector of direction of propagation when the structure is symmetrical and when the meshes of the upper lip are not defined with LEVRE\_SUP (§3.3.3), the keyword DTAN\_ORIG is then obligatory.

Keywords DTAN\_ORIG and DTAN\_EXTR no direction has in the case of a closed bottom and is then prohibited. Indeed, any node has then a mesh segment on its left and its right-hand side; nothing distinguishes the node origin and the direction from propagation in this point a current node from the bottom from crack.

```
◇ / DTAN_ORIG      = (Drug addict, Toy, Toz),          [1_R]
   DTAN_EXTR     = (Tex, Tey, Tez),                [1_R]
  / VECT_GRNO_ORIG = lgrno ,                        [1_gr_noeud]
   VECT_GRNO_EXTR = lgrno ,                        [1_gr_noeud]
```

#### 3.2.1 Keyword factor FOND\_FISS

##### 3.2.1.1 Keyword TYPE\_FOND

There are three possibilities to define the bottom of crack:

- 1) If the bottom of crack is defined by a curve open in opposition to the bottom defined by closed curve. It is then a question of informing TYPE\_FOND = 'OPEN'. This value is the value by default.
- 2) If the bottom of crack is defined by a closed curve. It is then a question of informing TYPE\_FOND = 'FIRM'.
- 3) If the bottom of crack is defined by two entities coincidentes geometrically (in 2D and 3D). It is then a question of defining two basic types TYPE\_FOND = 'INF' and TYPE\_FOND = 'SUP'.

##### 3.2.1.2 Keyword GROUP\_NO

*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>)

```
/ ♦ / GROUP_NO = lgrno
```

List of groups of nodes limited to 1 obligatorily ordered compared to the bottom of crack. In the case of a closed bottom, the first node must be also the last node of the group.

### 3.2.1.3 Keyword **NODE**

```
/ NODE = lno
```

List of nodes obligatorily ordered compared to the bottom of crack. In the case of a closed bottom, the first node must be also the last node of the list of nodes.

### 3.2.1.4 Keyword **GROUP\_MA**

```
/ ♦ / GROUP_MA = lgrma
```

List of groups of meshes of the type **SEG2** or **SEG3**, ordinates or not compared to the bottom of crack. This keyword can be used only in 3D.

### 3.2.1.5 Keyword **MESH**

```
/ MESH = lma
```

List of meshes, ordinates or not, of type **SEG2** or **SEG3**. This keyword can be used only in 3D.

### 3.2.1.6 Keyword **NOEUD\_ORIG**

```
♦ / ♦ / NOEUD_ORIG = No
```

Single node defining the origin. To be an end of the way defining the bottom of crack, it must rest on one and only one nets **lgrma** or **lma**. This keyword can be defined only if **MESH** or **GROUP\_MA** is defined. This keyword can be used only in 3D.

### 3.2.1.7 Keyword **GROUP\_NO\_ORIG**

```
/ GROUP_NO_ORIG = grno
```

Single group of node, containing a single node. To be an end of the way defining the bottom of crack, it must rest on one and only one nets **lgrma** or **lma**. This keyword can be defined only if **MESH** or **GROUP\_MA** is defined. This keyword can be used only in 3D.

### 3.2.1.8 Keyword **NOEUD\_EXTR**

```
♦ / NOEUD_EXTR = No
```

Single node defining the end. This data is optional and is only used to check that the node end obtained by the operator is well that of which the user thinks. The code will stop in error if it is not the case. This keyword can be defined only if **NOEUD\_ORIG** or **GROUP\_NO\_ORIG** is defined. This keyword can be used only in 3D.

### 3.2.1.9 Keyword **GROUP\_NO\_EXTR**

```
/ GROUP_NO_EXTR = grno
```

Single group of node, containing a single node. This data is optional and is only used to check that the node end obtained by the operator is well that of

which the user thinks. The code will stop in error if it is not the case. This keyword can be defined only if `NOEUD_ORIG` or `GROUP_NO_ORIG` is defined. This keyword can be used only in 3D.

### 3.2.1.10 Operand `DTAN_ORIG`

◇ / `DTAN_ORIG` = vector (Drug addict, Toy, Toz)

Direction  $T_{or}$  at the origin of the bottom of crack directed in the direction of the propagation of the crack.

### 3.2.1.11 Operand `VECT_GRNO_ORIG`

/ `VECT_GRNO_ORIG` = lgrno with lgrno list of two groups of nodes containing each one only one node.

Direction  $T_{or}$  deduced from the data of two nodes.

### 3.2.1.12 Operand `DTAN_EXTR`

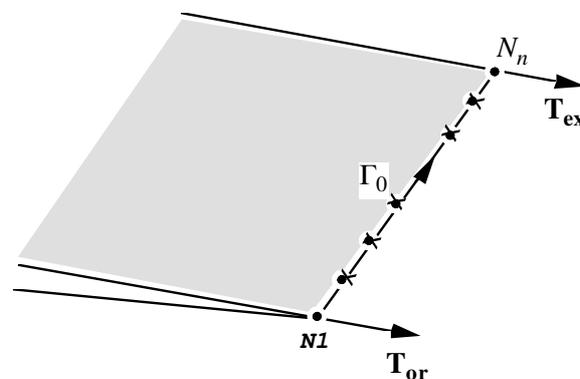
/ `DTAN_EXTR` = vector (Tex, Tey, Tez)

Direction  $T_{ex}$  at the end of the bottom of crack in the direction of the propagation of the crack.

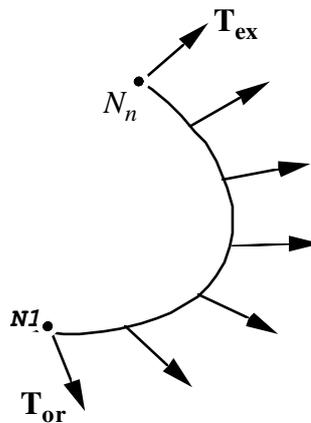
### 3.2.1.13 Operand `VECT_GRNO_EXTR`

/ `VECT_GRNO_EXTR` = lgrno with lgrno list of two groups of nodes containing each one only one node.

Direction  $T_{ex}$  deduced from the data of two nodes.



Example 1: case of a rectilinear plane crack:



Example 2: case of a crack planes curve:

**Note:**

$T_{or}$  and  $T_{ex}$  are normalized automatically.

### 3.3 Description of the lips

Two initial configurations are treated:

- 1) if the lips are stuck
- 2) if the lips are separated

In the case of stuck lips, two cases are distinguished:

- if the grid is complete then the algorithm identifies the meshes connected to the bottom of crack and the lips.
- if the grid is to be supplemented by symmetry compared to the average plan of the lips then the algorithm requires the definition of the upper lips.

The direction of propagation of the crack and the normal with the plan of the crack are calculated for each node of the bottom of crack.

In the case of separated lips, it is necessary to give the normal to the plan of the crack using the operand `NORMAL` (2D and 3D for the plane cracks only).

For a posterior use in `POST_K1_K2_K3` (keyword `FOND_FISS`), the knowledge of the meshes of the lips is essential. Those are calculated in the case of stuck lips but not in the case of separated lips (one will use then `CALC_G`).

For the nonplane cracks 3D, the direction of propagation of the crack in any point of the bottom of crack is built in this operator and is used by the operators `CALC_THETA` [U4.82.02] and `CALC_G` [U4.82.03].

Currently, calculations of breaking process by `CALC_G`, `POST_K1_K2_K3` or other are not possible for defects 3D not plans and whose lips are separated.

#### 3.3.1 Keyword `CONFIG_INIT`

The initial configuration is that described by the grid. The lips are:

1. stuck if the angle between the 2 lips is lower or equal to  $5^\circ$ ;
2. taken off in the contrary case.

*Note:* The calculation of the factors of intensity of the constraints with the operator `POST_K1_K2_K3` [U4.82.05], or with the options `'CALC_K_G'` or `'CALC_K_MAX'` of the operator `CALC_G` [U4.82.03], can be carried out only if `CONFIG_INIT='COLLEE'`.

## 3.3.2 Keyword `SYME`

This keyword makes it possible to specify if modeling used takes account of a symmetry of the structure compared to the average plan of the lips of the crack (see Figure 3.1). If `SYME = 'YES'`, the result will be automatically multiplied by 2.

If `SYME = 'YES'` and `CONFIG_INIT='COLLEE'`, it is necessary then is to define the upper lip of the crack (`LEVRE_SUP`, §3.3.3) that is to say to give the direction of propagation to the point origin (`DTAN_ORIG`, §3.2.1.10), in order to know of which with dimensions of the bottom the crack is located.

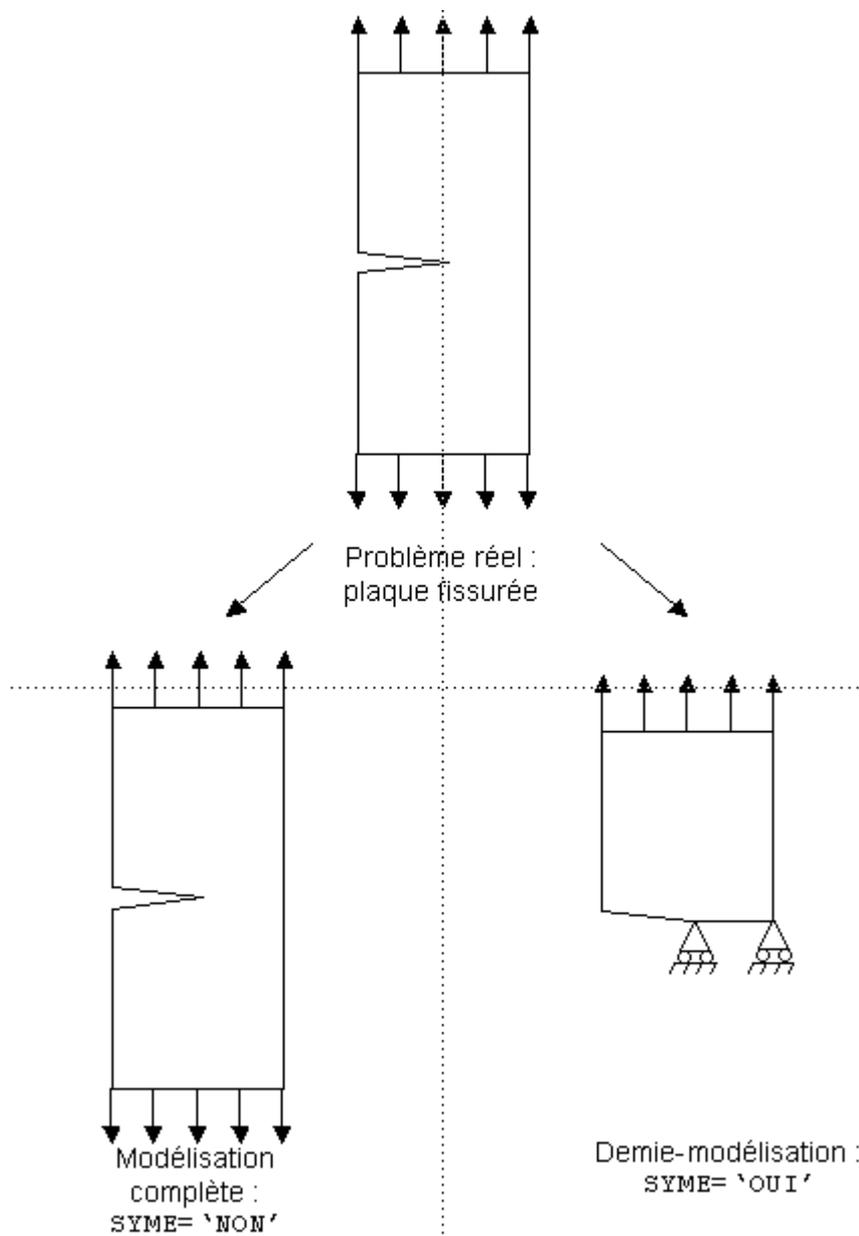


Figure 3.1 : Definition of symmetry.

## 3.3.3 Keyword LEVRE\_SUP

◇ LEVRE\_SUP =

The whole of the faces of the elements defines which are pressed on the upper lip of the crack. The whole of these faces is specified by the operands:

/ GROUP\_MA = lgrma list of groups of meshes.  
/ MESH = lma list of meshes.

The meshes are thus surface if the model is 3D and linear if the model is 2D.

## 3.3.4 Keyword LEVRE\_INF

◇ LEVRE\_INF =

The whole of the faces of the elements 3D defines which are pressed on the lower lip of the crack. If the crack is on a symmetry plane, this keyword should not be indicated.

The whole of these faces is specified by the operands:

/ GROUP\_MA = lgrma list of groups of meshes.  
/ MESH = lma list of meshes.

The meshes are thus surface if the model is 3D and linear if the model is 2D.

## 3.3.5 Operand NORMAL

This operator is to be defined only in the case of a crack in the separated lips. That concerns cases of open defect or notch.

The keyword `NORMAL` is used to specify the normal vector with the plan of these lips, therefore with the crack itself. This vector is used in any point of the bottom of crack to determine the direction of propagation and thus supposes that the crack is plane. The normal is then not the normal with the lips, but with the plan of propagation (symmetry plane).

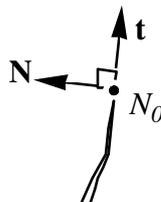
/ ◆ `NORMAL = (Nx, Ny, Nz)`

The keyword `NORMAL` allows to introduce the components  $N_x$ ,  $N_y$ ,  $N_z$  in the total reference mark of a normal  $N$  with the plan of the crack with the following convention of direction:

- In 3D,  $\mathbf{n} = \Gamma_0 \wedge N$ , where  $\mathbf{n}$  is the normal external with the crack in the plan of the lips,  $\Gamma_0$  is the bottom of crack directed (defined by the keyword `FOND_FISS`),
- in 2D, the normal  $N$  is defined such as the reference mark  $(N_0, \mathbf{t}, N)$  that is to say direct, with:

$N_0$  the node of the bottom of crack,

$\mathbf{t}$  direction of propagation of the crack.



In all the cases,  $N$  is normalized automatically. It is necessary to give the three components of the vector even in 2D.

## 3.4 Operand PREC\_NORM

This operand is useful only in the case 3D for a crack defined starting from its lips (keywords LEVRE\_SUP and LEVRE\_INF), before a calculation with POST\_K1\_K2\_K3 [U4.82.05].

The parameter PREC\_NORM the precision used in the research of the nodes of the lips defines which are on normal directions at the bottom of crack: for the interpolation of the jumps of displacement, one indeed use the nodes of which the distance  $D$  with the normal right-hand side at the bottom and passing by a node of this bottom checks:

$$d < \text{PREC\_NORM} \cdot l_f$$

where  $l_f$  is the minimal distance between two successive nodes of the bottom of crack. To increase the value of PREC\_NORM come down to increase the number of nodes potentially retained for the calculation of  $K$  in POST\_K1\_K2\_K3.

## 4 Phase of checks

With the execution:

- checking of the membership of the entities (nodes and meshes) to the grid,
- checking that the bottom of crack defined by the data of a list of mesh constitutes a related way well,
- checking that it NOEUD\_ORIG belongs well to one of the meshes and that it is well one of the two ends of the way defined by these meshes,
- in the case of a crack planes where the keyword NORMAL was used, one checks the orthogonality of this normal with the 2 tangent vectors given by the operands DTAN\_ORIG and DTAN\_EXTR,
- if the lips are defined by their group of meshes, it is checked that the surface meshes of the two lips are quite distinct and that the nodes of the bottom of crack belong well to at least a mesh of the lips.

In the structure of data produced by the operator are stored: the list of the nodes of the bottom of crack, the list of the meshes of the lips, if definite tangents at the ends, if definite the normal with the plan of the crack, and - if LEVRE\_SUP / LEVRE\_INF are used the list of the nodes of the lips belonging to the normals at the bottom of crack passing by each node of the bottom.

## 5 Examples

### 5.1 Whole crack defined by various types of entities in 3D

The bottom of crack is defined by a list of groups of nodes, the upper lip by a list of meshes, the lower lip by a list of groups of meshes.

```
Fiss1 = DEFI_FOND_FISS (GRID = my,  
    FOND_FISS = _F (GROUP_NO = ('GRN1', 'GRN2', 'GRN3'),  
        DTAN_ORIG = (1., 0., 0.),  
        DTAN_EXTR = (1., 0., 0.)),  
    LEVRE_SUP = _F (MESH = ('MA5', 'MA13', 'MA4',  
        'MA12')),  
    LEVRE_INF = _F (GROUP_MA = (' GRM1 ', ' GRM2 '),),  
)
```

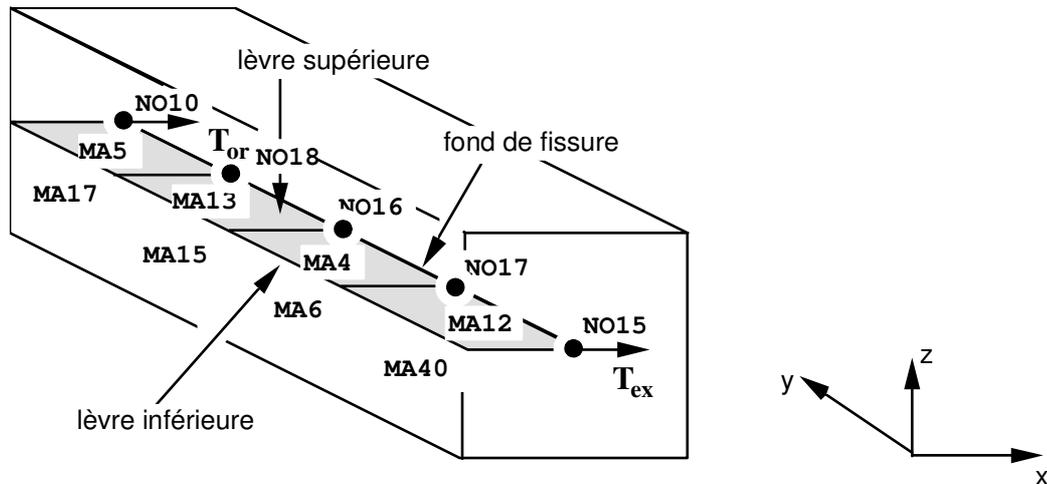
with the following groups defined in the grid my :

```
GRN1: {N010 N018}          GRN2: {N018 N016 N017}  
GRN3: {N017 N015}  
GRM1: {MA17 MA15 MA6}     GRM2: {MA40}
```

The bottom of crack is made up here by the nodes N010, N018, N016, N017, N015 in an ordered way.

*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>)

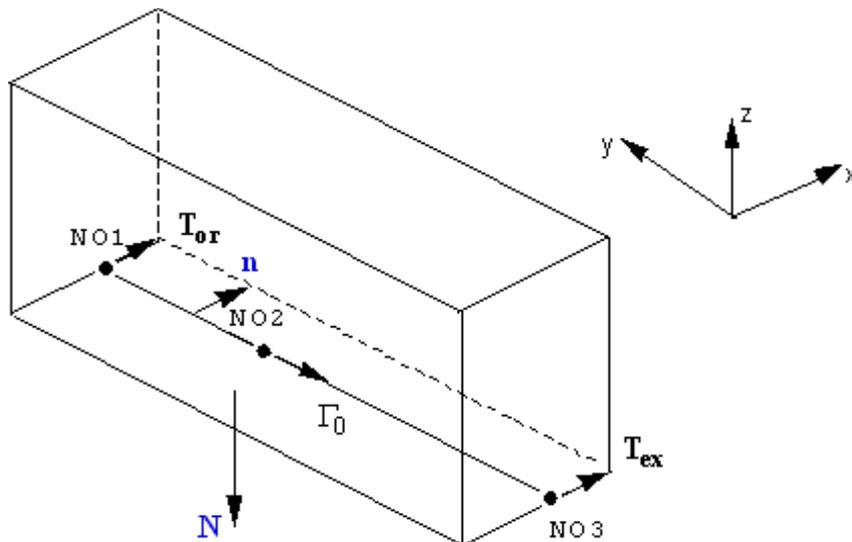


## 5.2 Use of the operand NORMAL (crack planes in 3D)

```
fiss = DEFI_FOND_FISS ( GRID          = my,
                        FOND_FISS =_F (NOEUD= ('NO1', 'NO2',
'NO3'),
                                      DTAN_ORIG = (1. , 0. , 0.),
                                      DTAN_EXTR = (1. ,0. , 0.)),
                        CONFIG_INIT=' DECOLLEE',
                        NORMAL          = (0. , 0. , - 1.),
                        )
```

The normal is defined  $N$  with the plan of the crack.

Direction  $\mathbf{n}$  normal at the bottom of crack in the plan of the lips of the crack is determined by  $\mathbf{n} = \Gamma_0 \wedge N$ .



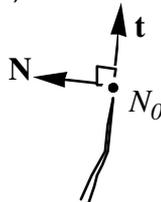
## 5.3 Crack in 2D

The bottom of crack is defined by the node  $N_{10}$  grid  $my$ . The groups of the meshes of the lips are noted respectively  $GMSUP$  and  $GMINF$ . The crack can be defined starting from the normal:

```
fiss = DEFI_FOND_FISS ( GRID = my,  
                        FOND_FISS = _F (NODE = 'N10'),  
                        CONFIG_INIT = 'DECOLLEE',  
                        NORMAL = (- 1. , 1. , 0.),  
                        )
```

maybe starting from the meshes of the lips:

```
fiss = DEFI_FOND_FISS ( GRID = my,  
                        FOND_FISS = _F (NODE = 'N10'),  
                        LEVRE_SUP = 'GMSUP',  
                        LEVRE_INF = 'GMINF',  
                        )
```

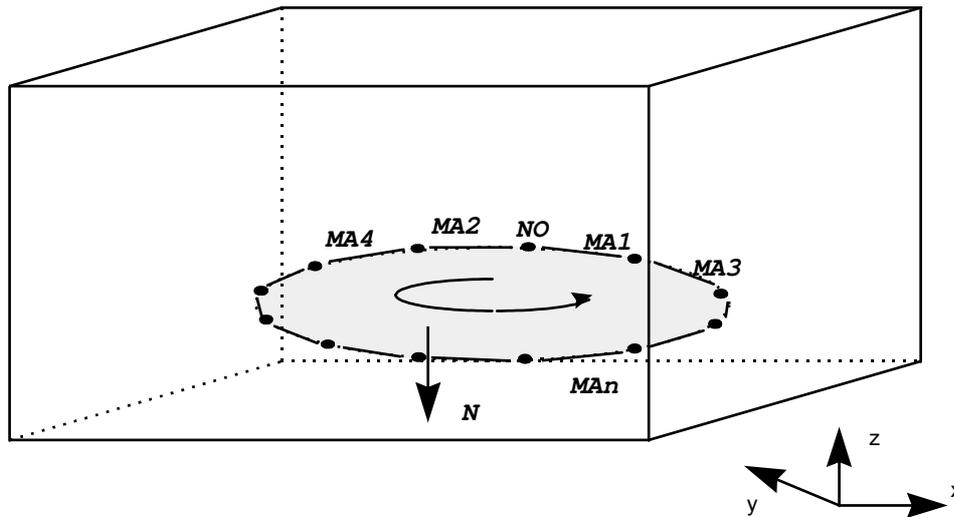


## 5.4 Fund of crack closed with scheduling of the nodes

```
fiss = DEFI_FOND_FISS (GRID = my,  
                        FOND_FISS=_F (TYPE_FOND=' FERME',  
                                       MAILLE= ('MA3', 'MA1', 'MA2'...),  
                                       GROUP_NO_ORIG = 'NO',  
                                       MAILLE_ORIG = 'MA2' ),  
                        CONFIG_INIT = 'DECOLLEE',  
                        NORMAL = (0. , 0. , - 1.),  
                        )
```

The normal is defined  $\mathbf{N}$  with the plan of the crack. Direction  $\mathbf{n}$  normal at the bottom of crack in the plan of the lips of the crack is determined by  $\mathbf{n} = \Gamma_0 \wedge \mathbf{N}$ .

The order of declaration of the meshes in the list does not have any importance. The operator checks that the node  $NO$  belongs well to the mesh  $MA2$  and that the whole of the meshes segments provided form well a closed related curve. The order of the nodes in the produced concept will be that given by the arrow of the drawing below, on the basis of  $NO$ .



## 5.5 Fund of crack defines by two groups of nodes coincidents

Bottom of crack of the grid `my` is defined by 2 groups of nodes:

- 1) `'FONDINF'` : group of nodes belonging to the lower lip.
- 2) `'FONDSUP'` : group of nodes belonging to the upper lip whose nodes coincide geometrically with those of the group `'FONDINF'`

```
fiss = DEFI_FOND_FISS (GRID = my,
    FOND_FISS=_F (TYPE_FOND=' INF',
        GROUP_NO=' FONDINF',
        DTAN_ORIG= (0. , 1. , 0.),
        DTAN_EXTR= (1. , 0. , 0.)),
    FOND_FISS=_F (TYPE_FOND=' SUP',
        GROUP_NO=' FONDSUP',
        DTAN_ORIG= (0. , 1. , 0.),
        DTAN_EXTR= (1. , 0. , 0.)),
    LEVRE_INF=_F (GROUP_MA=' LEVINF'),
    LEVRE_SUP=_F (GROUP_MA=' LEVSUP'),
)
```