

sd_contact_xfem, produced by operator DEFI_CONTACT [U4.44.01
]

3

and

C

\.FOND.TAILLE_R'
R

OJB

S

V

CHAM_ELEM

3D
2D

Cutting ⁽¹⁾

Preliminary phase of cutting

Each element (hexahedron, pentahedron) is virtually cut out in tetrahedrons

6 tetrahedrons to the maximum

Each element (quadrangle) is virtually cut out in triangles

2 triangles to the maximum

Cutting ⁽²⁾

Under-cutting

Each tetrahedron is in its turn cut out under-tetrahedrons

6 under-tetrahedrons to the maximum

Each triangle is in its turn cut out under-triangles

3 under-triangles to the maximum

Points of intersection

Points of intersection enters the edges of the tetrahedrons resulting from cutting ⁽¹⁾ and crack surfaces it

11 points of intersection to the maximum

Points of intersection enters the edges of the triangles resulting from cutting ⁽¹⁾ and the curve of the crack

3 points of intersection to the maximum

Points mediums

Points in the middle of the edges of the triangles resulting from cutting ⁽²⁾ [on condition that coinciding with no node of the complete initial element]

66 points to the maximum (59 points mediums to the maximum distinct from the central nodes and 7 central points to the maximum)

Points in the middle of the edges of the triangles resulting from cutting ⁽²⁾ [on condition that coinciding with no node of the initial element]

11 points mediums to the maximum (10 points mediums to the maximum distinct from the central nodes and 1 central point to the maximum)

Point central of

quadratic quadrangle

Storage of the points mediums of the faces quadrangle of the quadratic polyhedron.

7 central points to the maximum

Point of intersection of the two right diagonals of a quadratic quadrangle

1 central point to the maximum

Process of cutting⁽¹⁾ and under-cutting⁽²⁾

1 element X-FEM 3D
tetrahedron n°1

6 under-tetrahedrons with the max. one.⁽²⁾

tetrahedron n°2

6 under-tetrahedrons with the max. one.

...

1 element X-FEM 2D
triangle n°1

3 under-triangles with the max. one.

triangle n°2

3 under-triangles with the max. one.

(2)

TOPOFAC.HE
TOPONO.HFA
TOPONO.HFA

1 Contents of the objects of sd_contact_xfem

1.1 ACF (addresses IAD=JCMCF+ZCMCF* (IZONE-1))

1.2 .CAR

ZR (IAD+1) = diagram of integration (=1 if INTEGRATION = 'GAUSS'
(FPG12)

=4 if INTEGRATION = 'FPG4'

=6 if INTEGRATION = 'FPG6'

=7 if INTEGRATION = 'FPG7')

ZR (IAD+2) = COEF_REGU_CONT

ZR (IAD+3) = COEF_REGU_FROT

ZR (IAD+4) = coefficient of Coulomb for friction

ZR (IAD+5) = coefficient characterizing the presence of friction and its type

(=1 if FROTTEMENT=' SANS'

=3 if FROTTEMENT=' COULOMB')

ZR (IAD+6) = initial value of the threshold of friction (SEUIL_INIT)

ZR (IAD+23) = coefficient of setting at the level of contact pressures (COEF_ECHELLE)

ZR (IAD+24) = algorithm of restriction of the space of the multipliers of Lagrange

(=0 if ALGO_LAGR = 'NOT'

=1 if ALGO_LAGR = 'VERSION1'

=2 if ALGO_LAGR = 'VERSION2')

1.3 .ECPDON (addresses IAD=JECPD+ZECPD* (IZONE-1))

ZI (IAD+1) = indicating of axi-symmetry (=0 because MODL_AXIS=' NON')
ZI (IAD+2) = ITER_CONT_MAXI
ZI (IAD+3) = ITER_FROT_MAXI
ZI (IAD+4) = ITER_GEOM_MAXI (=1 because there is no geometrical loop)
ZI (IAD+5) = indicating of the initial contact (=0 if CONTACT_INIT=' NON'
=1 if CONTACT_INIT=' OUI')

1.4 .METHCO (addresses IAD=JMETH+ZMETH* (IZONE-1))

ZI (IAD+6) = number of the method used
(=11 if continuous method
=12 if method continues with contact slide)

1.5 .XFIMAI

Vector of K_8 of length equalizes with the number of zones of contact. For each zone of contact, is stored the main name of the crack (name of the sd_fiss_xfem).

1.6 .XNRELL

Vector of K_{24} of length with 3 times the number of cracks equalizes. For each crack, one stores the name of the objects for relations LBB of the sd_fiss_xfem (see [§4.5.2]):

- fiss_xfem/'\'.LISEQ',

1.7 RESOCO

1.8 relative

1.9 with

Structures of Data related to X-FEM

Summary

This document describes the structures of data (SD) related to method X-FEM. These objects are in several SD:

- 1) `sd_fiss_xfem`, produced by the operator `DEFI_FISS_XFEM` [U4.82.08] and used for a calculation of breaking process within the framework of method X-FEM [R7.02.12]. This structure of data is also used and produced by the operator `PROPA_FISS` [U4.82.11],
- 2) `sd_modele_xfem`, produced by the operator `MODI_MODELE_XFEM` [U4.41.11],
- 3) SD created for the resolution of the contact..

2 General information

During a calculation X-FEM, information concerning the crack not being contained in the grid, it is necessary to store them in a structure of data (SD) specific. Then, the enrichment of the elements leads to new finite elements, whose certain characteristics will be stored in the model. Then, if the contact is defined on the lips of the crack, a load of contact is then created, which contains information specific to X-FEM. At the time of the resolution of the problem of contact a specific SD is created then destroyed during the execution of `STAT_NON_LINE`. Thus, the objects relating to method X-FEM are distributed in 4 Different SD: `sd_fiss_xfem`, `model`, `load` and SD created at the time of the resolution of the problem of contact-rubbing. The SD defining the load of contact, produced by `DEFI_CONTACT`, is described completely in [D4.06.14]. The objects of this SD specific to formulation X-FEM are described in [D4.06.14, §8]. The three others SD are the object of three most of this document.

An object of the type `sd_fiss_xfem` described a crack contained in a grid 2D or 3D, without this grid not being inevitably in conformity with the geometry of the crack. The structure of data `sd_fiss_xfem` contains the description of the crack by the means of level sets, the local bases associated, of the data related to enrichment of the finite elements around the crack, topological information concerning the under-cutting of the finite elements nouveaux riches, and the data related to the contact between the lips of the crack. Moreover, this structure is enriched by the addition of data necessary to the propagation of the crack.

The enriched model makes it possible to globalize local information with each crack on the total grid. It contains also information on the cracks of the model. This document describes only the objects of the model specific to X-FEM (for the description of `sd_modele`, to see [D4.06.02]).

If the contact is defined on the lips, the SD of contact contains information specific to X-FEM, in particular concerning the algorithm of construction of the multipliers of Lagrange to satisfy condition LBB: lists of relations to be imposed,... This document describes only the objects of the contact specific to X-FEM and stored in Lhas `sd_modele` or in the SD created at the time of the resolution of the contact (for the description of the SD related to the contact friction, to see [D4.06.14]).

3 Tree structure

sd_fiss_xfem	(K8)	..: =record			
◆	\.INFO'		OJB	S	
	V	K16			
	\.MAILLAGE'		OJB	S	V
		K8			
◇	\.CHAMPS.LVS'		OJB	S	
	V	L			
◆	\.LTNO'		CHAM_NO		
	\.LNNO'		CHAM_NO		
	\.GRLTNO'		CHAM_NO		
	\.GRLNNO'		CHAM_NO		
	\.BASLOC'		CHAM_NO		
	\.STNO'		CHAM_NO		
	\.STNOR'		CHAM_NO		
◆	\.GROUP_MA_ENRI'		OJB	S	V
	I				
	\.GROUP_NO_ENRI'		OJB	S	V
	I				
	\.CARAFOND'		OJB	S	V
	R				
◇	% if the structure contains a bottom of crack				
	\.FONDFISS'		OJB	S	V
	R				
	\.BASEFOND'		OJB	S	V
	R				
	\.FONDMULT'		OJB	S	V
	I				
	\.LTNS'		OJB	S	V
	K24				
	\.LTNT'		OJB	S	V
	K16				
	\.FOND.TAILLE_R'		OJB	S	V
	R				
	\.NOFACPTFON '		OJB	S	V
	I				
◇	% so of the meshes are entirely cut by the crack				
	\.MAILFISS.HEAV'		OJB	S	V
	I				
◇	% so of the meshes contain the bottom of crack				
	\.MAILFISS.CTIP'		OJB	S	V
	I				
◇	% so of the meshes are cut by the crack and contain the bottom of crack				
	\.MAILFISS.HECT'		OJB	S	V
	I				
◇	% so of the contact is defined on certain cracks of the model				

	`.MAILFISS.CONT`	OJB	S	V	
	I				
◇	% if the crack is resulting from a propagation by the operator				
	`.PRO.RAYON_TORE`	OJB	S	V	R
	`.PRO.NOEUD_TORE`	OJB	S	V	L
	`.PRO.NOEUD_PROJ`	OJB	S	V	L
◇	% if a grid is associated with the crack by				
	`.GRI.MAILLAGE`	OJB	S	V	
K8					
	`.GRI.LNNO`			CHAM_NO	
	`.GRI.LTNO`			CHAM_NO	
	`.GRI.GRLNNO`			CHAM_NO	
	`.GRI.GRLTNO`			CHAM_NO	
◇	% if the grid contains a bottom of crack				
	`.FONDFISG`	OJB		S	V
	R				
◇	% so of the relations of equality between unknown factors of contact are imposed				
	`.LISEQ`	OJB		S	
	V	I			
◇	% so of the relations of equality between unknown factors of contact are imposed and if there are multi-Heaviside elements				
	`.LISEQ_LAGR`	OJB		S	
	V	I			
◇	% if there are groups of connected vital edges				
	`.CNCTE`	OJB		S	
	V	I			
◇	% if the keyword JUNCTION was used				
	`.JONFISS`	OJB		S	
	V	K8			
	`.JONCOEF`	OJB		S	
	V	I			

```

sd_modele_xfem      (K8)  .: =record

  ◆  \.MODELE_SAIN'                OJB                S
      V                K8

  ◆  \.TOPOSE.PIN'                CHAM_ELEM
      \.TOPOSE.PSEMI \          CHAM_ELEM
      \.TOPOSE.CNS'            CHAM_ELEM
      \.TOPOSE.HEA'            CHAM_ELEM
      \.TOPOSE.LON'            CHAM_ELEM
      \.TOPOSE.PAI'            CHAM_ELEM
      \.TOPOSE.POLympic Games \          CHAM_ELEM

  ◆  \.TOPONO.NHO'                CHAM_ELNO
      \.TOPONO.HSE'            CHAM_ELEM

  ◇   % if there are multi-Heaviside elements
      \.TOPONO.HFA'            CHAM_ELEM

  ◆  \.TOPOFAC.PI'                CHAM_ELEM
      \.TOPOFAC.AI'            CHAM_ELEM
      \.TOPOFAC.CF'            CHAM_ELEM
      \.TOPOFAC.LO'            CHAM_ELEM
      \.TOPOFAC.BA'            CHAM_ELEM
      \.TOPOFAC.OE'            CHAM_ELEM
      \.TOPOFAC.GE'            CHAM_ELEM
      \.TOPOFAC.GM'            CHAM_ELEM

  ◇   % if there are multi-Heaviside elements
      \.TOPOFAC.HE'            CHAM_ELEM

  ◆  \.LTNO'                      CHAM_ELNO
      \.LNNO'                  CHAM_ELNO
      \.BASLOC'                CHAM_ELNO
      \.STNO'                  CHAM_ELNO
      \.NOXFEM'                CHAM_ELNO

  ◇   % so of the elements store the information of several cracks
      \.FISSNO'                CHAM_ELNO
      \.HEAVNO'                CHAM_ELNO

  ◆  \.XFEM_CONT'                OJB                S
      V                I

  ◆  \.NFIS'                      OJB                S
      V                I
      \.FISS'                  OJB                S                V
          K8
      \.XMAFIS'                CHAM_ELEM
      \.PRE_COND'              OJB                S                V
          K8

sd_contact_xfem    (K16)  .: =record

  ◆  \.CARACF'                    OJB                S                V
      R

```


Code_Aster

Version
default

Titre : Structures de Données fiss_xfem
Responsable : GÉNIAUT Samuel

Date : 01/12/2017 Page : 9/41
Clé : D4.10.02 Révision :
85757f2782a2

\.ECPDON'	OJB	S	V
I			
\.METHCO'	OJB	S	V
I			
\.XFIMAI'	OJB	S	V
K8			
\.XNRELL'	OJB	S	V
K24			

◇ % cards of contact for the approach great slip

\.XFPO'	MAP
\.XFST'	MAP
\.XFPI'	MAP
\.XFAI'	MAP
\.XFCE'	MAP
\.XFHE'	MAP
\.XFPL'	MAP

4 Contents of the objects of sd_fiss_xfem

Notice preliminary on the distinction between 2D and 3D:

The presence of certain objects in the structure of data `sd_fiss_xfem` depends sometimes on the dimension of the grid on which the crack is represented. Thus, the distinction between 2D and 3D are referred corresponding to grid contained in the model given by the user under the keyword `MODEL` order `DEFI_FISS_XFEM`.

4.1 Objects generals

4.1.1 .INFO

The concept `.INFORMATION` is a vector of `K16` of length 3 container of general information.

The 1st component of this vector informs about the type of discontinuity relative to `sd_fiss_xfem`: either a crack, or an interface. The 2nd component indicates the discontinuous field: either the field of displacement, or the stress field. The 3rd component specifies if the bottom is closed or opened.

`.INFORMATION (1) = 'CRACK' or 'INTERFACE'`

`.INFORMATION (2) = 'DEPL' or 'SIGM'`

`.INFORMATION (3) = 'OPEN' or 'FIRM'`

One reaches the values of this vector only by the routine `dismoif`.

Example:

```
CAL DISMOI ('F', 'TYPE_DISCONTINUITE', FISS, 'FISS_XFEM', IBID, TYPDIS, IRET)
```

```
CAL DISMOI ('F', 'CHAM_DISCONTINUITE', FISS, 'FISS_XFEM', IBID, TYPCHA, IRET)
```

```
CAL DISMOI ('F', 'TYPE_FOND', FISS, 'FISS_XFEM', IBID, TYPFON, IRET)
```

4.1.2 .GRID

The concept `.GRID` of `K8` length 1 container is a vector the name of the grid on which the crack (value of the keyword is built `GRID` as starter of `DEFI_FISS_XFEM`). In the event of propagation, this vector is duplicated old crack with the new crack.

One reaches this vector only by the routine `dismoif`.

Example:

```
CAL DISMOI ('F', 'NOM_MALLAGE', FISS, 'FISS_XFEM', IBID, MODEL, IRET)
```

4.2 Objects relating to the level sets

4.2.1 .CHAMPS.LVS

The concept `.CHAMPS.LVS` is a vector of Boolean length 1. If this vector is present (and its value is then always equal to `TRUE`), the fields level set of the current crack were given directly by the user (by `DEFI_FISS_XFEM` by using the keywords `CHAMP_NO_LSN` and `CHAMP_NO_LST`) and they were not calculated.

If the two fields were extracted from a crack propagated by `PROPA_FISS`, information on the localization of the field (`.PRO.RAYON_TORE` (§4.4.1) and `.PRO.NOEUD_TORE` (§4.4.2)) and on the use of an auxiliary grid for the propagated crack (`.GRI.MALLAGE` (§4.4.4), `.GRI.L*NO` (§4.4.5) and `.GRI.GRL*NO` (§4.4.6)) were lost and the SD of the current crack does not contain them. Indeed this information is stored on the level of the SD `fiss_xfem` and not on the level of the SD `champ_no`. That poses problem if the current crack is propagated by `PROPA_FISS` because one could get false results. Indeed whenever the localization of the field or the auxiliary grid were used, the level sets were put at days only in one small field around the bottom of the crack: in the first case that is obvious, and in the second case the level sets were projected on the physical grid with a field of localised projection independently of the use of the localization of a field of calculation in bottom of

crack on the grid. Apart from the field of localization/projection the value of the levels set is the same one as that calculated to define the crack: the level sets are thus not continuous in extreme cases of the field of localization/projection and the description of the crack is thus not correct on all the grid. If information on the field of localization and the auxiliary grid is not stored in the SD of the new crack, `PROPA_FISS` propagate the crack by considering that the level sets are correct on all the grid. When the bottom of the crack propagated leaves the field localization/projection used for the initial crack before propagation, false results are got and one has problems of convergence in the rebootstraping and the reorthogonalisation of the level set.

If the level sets given to define the new crack were not extracted from a crack propagated by `PROPA_FISS`, or although the localization of the field or the auxiliary grid were not used with `PROPA_FISS`, the fields level sets given correctly represent the crack on all the grid and the propagation of the new crack can be calculated without problem by `PROPA_FISS`.

If this vector is present in the SD of the crack, `PROPA_FISS` an alarm emits so that the user checks if it is not in the presence of a field extracted from a crack propagated with `PROPA_FISS` that it would use with localization of the field of calculation or with an auxiliary grid.

4.2.2 .LTNO and .LNNO

The concept `.LTNO` (resp. `.LNNO`) is a field with the nodes (`CHAM_NO`) scalar which contains for each node of the grid the actual value of the level set tangent (resp. normal) with the crack.

4.2.3 .GRLTNO and .GRLNNO

The concept `.GRLTNO` (resp. `.GRLNNO`) is a field with the nodes (`CHAM_NO`) with 3 real components. It contains for each node of the grid the values of the gradient of the level set tangent (resp. normal) in the 3 directions of space.

That is to say i $i^{\text{ème}}$ node of the grid,

$$V = .GRLTNO (I) ;$$

- V (1) Value of the following gradient x level set tangent, calculated with $i^{\text{ème}}$ node
- V (2) Value of the following gradient y level set tangent, calculated with $i^{\text{ème}}$ node
- V (3) Value of the following gradient z level set tangent, calculated with $i^{\text{ème}}$ node

In 2D, there are only 2 components according to x and y .

4.2.4 .BASLOC

The concept `.BASLOC` is a field with the nodes (`CHAM_NO`) with 9 real components (in 3D). It contains the origin and the vectors of the Local Base at the bottom of crack. For each node, the first three components are the coordinates of the project of the node on the bottom, which corresponds at the origin of the local base. The three following components are the coordinates of the 1^{er} vector (`GRLT`) base. The three last components are cordonnées of the 2^{ème} vector (`GRLN`) base. 3^{ème} vector of the base is not stored, because it is determined easily as being the vector product of the first 2 vectors.

That is to say i $i^{\text{ème}}$ node grid,

$$V = .BASLOC (I) ;$$

- V (1) Coordinate according to x project of node I on the bottom
- V (2) Coordinate according to y project of node I on the bottom
- V (3) Coordinate according to z project of node I on the bottom
- V (4) Coordinate according to x from the 1^{er} vector of the local base
- V (5) Coordinate according to y from the 1^{er} vector of the local base
- V (6) Coordinate according to z from the 1^{er} vector of the local base

- ∇ (7) Coordinate according to x from the 2^{ème} vector of the local base
- ∇ (8) Coordinate according to y from the 2^{ème} vector of the local base
- ∇ (9) Coordinate according to z from the 2^{ème} vector of the local base

In 2D, there are only 2 components according to x and y , are 6 components for `BASLOC`. Let us point out that in 2D, there is one bottom of crack, which is then a point (case of the emerging cracks 2D) and in this case all the nodes of the grid have the same project on the bottom of crack, that is to say there are two funds of crack, therefore two points (case of the cracks 2D not emerging) and in this case, part of the nodes of the grid is projected on the 1^{er} bottom of crack and another part of the nodes of the grid project on the 2^{ème} bottom of crack.

Note:

- For a discontinuous face of crack, projection used above must be carried out with precaution. It is advisable to distinguish two types of discontinuities:
 - that is to say the face continuous but is cut by a material discontinuity (hole in a part) what generates a discontinuous face in the matter, but prolongeable by continuity out of the matter,
 - that is to say the face is really discontinuous in the definition of the crack to propagate.

Before projection, it is necessary as a preliminary to test if the face is virtually continuous or if it is really discontinuous. To then detect the discontinuity of the face of crack, one is based on a simple criterion which consists in testing colinearity vectors of propagation (1^{er} vector of the local base) at the ends of the faces of crack between two consecutive faces:

$$\text{acos}(\vec{V}_{P_i} \cdot \vec{V}_{P_{i+1}}) < 10^\circ$$

If the angular threshold above is exceeded (10° angle), one treats the face finishing in P_i and the face starting in P_{i+1} like two faces separated during projection of each node on the total face of crack.

- In two dimensions, it is convenient "to turn over" the first vector of the local base (vector of propagation) to modify the sign of the mode of shearing plan (or mode 2) and to always obtain a sign positive. This modification is carried out locally only at ends of postprocessing and does not affect the structure of data described in this paragraph.

4.2.5 . FONDFISS

The vector `.FONDFISS` is a vector of real containing the coordinates of the points of the bottom of crack. The points are ordered according to the method described in the document [R7.02.12], so that a curvilinear X-coordinate can be defined.

If `NFON` is the number of points of the bottom of crack, then the length of the vector `.FONDFISS` is $4 \times \text{NFON}$. For each point of the bottom of crack, the first 3 components correspond to the 3 coordinates (in 3D) of the point, and the fourth component is its curvilinear X-coordinate.

When the bottom is closed, the last point is equal to the first. The last 4 terms of the vector `.FONDFISS` are then identical to the 4 first.

This structure is not modified in 2D. However one uses only the first 2 components, because neither the curvilinear X-coordinate nor the last geometrical component are relevant in 2D.

This object is created only if there exists at least a point of the bottom of crack.

4.2.6 . BASEFOND

The vector `.BASEFOND` is a vector of real containing the local base associated with each point of the bottom of crack.

If `NFON` is the number of points of the bottom of crack and if `NDIM` is the dimension of the model (2 in 2D and 3 in 3D), then the length of the vector `.BASEFOND` is $2 \times \text{NDIM} \times \text{NFON}$. In fact this

vector contains, successively for all the points of the bottom, the components (2 in 2D and 3 in 3D) of the normal vector to the plan of the crack and the direction of propagation (tangent vector to the plan of the crack).

In 3D, when the bottom is closed, the last point is equal to the first. The last 4 terms of the vector `.BASEFOND` are then identical to the 4 first.

4.2.7 . FONDMULT

In 3D, the bottom of crack is a line either closed (crack not emerging), or open (emerging crack). In most case, the bottom of crack is a continuous line, like that of the circular crack represented on Figure 4.2.7-1.

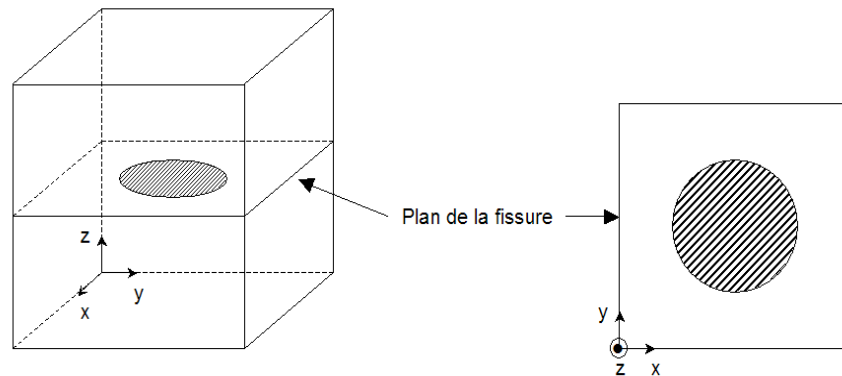


Figure 4.2.7-1 Case of a continuous bottom of crack.

However, it can happen that the bottom of crack is in fact made up of several discontinuous pieces. It is the case for example circular crack represented on Figure 4.2.7-2. In this case, one always speaks bottom of crack, like the whole of the pieces of the bottom. It is said that the bottom of crack is a multiple bottom. On the example of Figure 4.2.7-2, the bottom of crack is composed of the curved lines (BC) , (DE) , (FG) and (HA) .

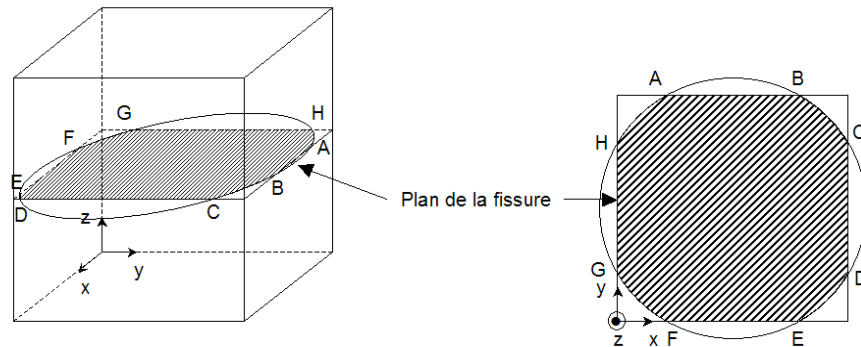


Figure 4.2.7-2 Case of a multiple bottom of crack.

The vector `.FONDMULT` is a vector of entières containing the indices of the starting and arrival points of the various pieces of the bottom of crack in `FONDFISS`. If `NFONFI` is the number of pieces of the bottom of crack, then the vector `.FONDMULT` is of dimension $2 \times NFONFI$.

For example:

`SI FONDMULT = (1; 8; 9; 12)`

Withat the time Piece n°1 of the bottom of crack: item 1 as in point 8 of `FONDFISS`

Morceau n°2 of the bottom of crack: item 9 as in point 12 of `FONDFISS`

In 2D, each point of the bottom of crack constitutes with him only a bottom, the vector `.FONDMULT` nevertheless is created. For example for 2 funds of cracks, it will be worth: `FONDMULT = (1; 1; 2; 2)`

The object is created provided there exists at least a point of the bottom of crack.

4.2.8 .CARAFOND

The vector `.CARAFOND` is a vector of realities of dimension 2. It contains the parameters users concerning the bottom of crack. The first component is the ray of enrichment of the elements and the second component is the number of layers of elements to be enriched.

As follows:

<code>CARAFOND(1) =</code>	<code>RAYON_ENRI</code>	if <code>RAYON_ENRI</code> is provided by the user,
	<code>0</code>	if not.
<code>CARAFOND(2) =</code>	<code>NB_COUCHES</code>	if <code>NB_COUCHES</code> is provided by the user,
	<code>0</code>	if not.

These data are those provided by the user in the order `DEFI_FISS_XFEM`. They are stored because they are useful again in the order `PROPA_FISS` to generate a propagated crack.

4.2.9 .LTNS

The vector `.LTNS` is a vector of character strings containing the two tables associated with the crack. These tables are accessible via the order `RECU_TABLE`.

The first named table `FOND_FISS` contains the coordinates of the points of funds of crack. In 2D, this one is consisted by the basic number of crack and of the coordinates x and y . Thus, the posting of the table will be:

```
NUME_FOND      COOR_X      COOR_Y
```

In 3D, this one described each bottom of crack. Thus, for each bottom of crack l , the table will provide the curvilinear X-coordinate and the coordinates of the points in bottom of crack l . The posting of the table will be:

```
NUME_FOND      NUME_PT      ABS_CURV      COOR_X      COOR_Y  
                COOR_Z
```

The second named table `NB_FOND_FISS` the number of funds of crack contains.

The construction of these tables rests on the use of the vectors `.FONDFISS` and `.FONDMULT`.

4.2.10. LTNT

The vector `.LTNT` is a vector of character strings containing the names of the two tables associated with the crack required to the order `RECU_TABLE`. It is about `FOND_FISS` and of `NB_FOND_FISS`.

4.2.11. FOND.TAILLE_R

This vector contains for each node of the bottom, an estimate of the maximum size according to the direction of propagation, of the meshes which are connected to them. These sizes are ordered according to the order of the nodes given in the vector `FONDFISS`.

One notes \vec{V}_{P_i} the vector of propagation of the local base to the node of the bottom NR_i and \vec{a}_{ijk} $K^{\text{ème}}$ edge of the $J^{\text{ème}}$ mesh to which the node NR belongs.

For each node of the bottom NR_i , the edges are projected \vec{a}_{ijk} on the vector of direction of propagation \vec{V}_{P_i} . Maximum size T_i meshes connected to NR_i is the maximum value of the absolute values of its projections. In other words, size T_i is equal to

$$T_i = \max_{\substack{1 \leq j \leq Nb_{\text{mailles},i} \\ 1 \leq k \leq Nb_{\text{arêtes},j}}} \left(\left| \vec{a}_{ijk} \cdot \vec{V}_{P_i} \right| \right),$$

where $Nb_{\text{mailles},i}$ is the number of meshes connected to the node NR_i and $Nb_{\text{arêtes},j}$ is the number of edges of the $J^{\text{ème}}$ mesh connected to the node NR_i .

4.2.12. NOFACPTFON

```
\.NOFACPTFON \ : S V I
```

This object contains the numbers of the nodes tops (total classification of the grid) of the faces of the elements parents containing the points of the bottom of crack.

That is to say:

```
V = .NOFACPTFON  
NFON = many points of the bottom of crack  
N = LONG (V) = 4xNFON
```

```
for I = 1, NFON :
```

$V(4 * (i-1) + 1)$	total number of the 1 ^{er} node top of the face containing the point of the bottom number I
$V(4 * (i-1) + 2)$	total number of the 2 ^{ème} node top of the face containing the point of the bottom number I
$V(4 * (i-1) + 3)$	total number of the 3 ^{ème} node top of the face containing the point of the bottom number I
$V(4 * (i-1) + 4)$	<ul style="list-style-type: none"> total number of the 4^{ème} node top of the face containing the point of the bottom number I if the face is quadrangular 0 if the face is triangular

Note:

- For each point of the bottom of crack, the face of which numbers of S nodes are stored in `.NOFACPTFON` is not necessarily single (for example degenerated case of a point of the bottom coinciding with an edge or a node): it is the last face found by the routine `intfac` in `xptfon` (phase of research of the points of the bottom of crack in the operator `DEFI_FISS_XFEM`)
- The object `.NOFACPTFON` is created only if the structure contains a bottom of crack.

4.3 Objects relating to enrichment

4.3.1 .GROUP_MA_ENRI

The vector `.GROUP_MA_ENRI` contains the list of the numbers of meshes of the zone to enrich.

4.3.2 .GROUP_NO_ENRI

The vector `.GROUP_NO_ENRI` contains the list of the numbers of nodes of the zone to enrich. Let us note that this vector can result from `.GROUP_MA_ENRI`.

Note:

These the last two vectors are stored because one wishes to re-use in `PROPA_FISS` related information with the keyword `GROUP_MA_ENRI` of `DEFI_FISS_XFEM`. And the only possibility of transmitting information is to store the numbers of the meshes and the nodes of the meshes of `GROUP_MA_ENRI`.

4.3.3 .STNO

The concept `.STNO` is a field with the nodes (`CHAM_NO`) with a whole component, corresponding to the statute of the node in question.

- If the node has its support entirely cut by the crack, its statute is worth 1.
- If the node is in a zone "close" (this concept is defined according to a criterion defined by the user) to the bottom of crack, its statute is worth 2.
- If the node satisfies the two preceding conditions, its statute is worth 3.
- In all the other cases, the statute is worth 0.

4.3.4 .STNOR

The concept `.STNOR` is exactly the same field as `.STNO` but with real components. It could thus be used for visualization with `Salome` for example.

4.3.5 .MAILFISS.HEAV

The vector `.MAILFISS.HEAV` is a vector of entières containing the list of the numbers of the meshes enriched by type Heaviside (mesh "ROUND").

4.3.6 .MAILFISS.CTIP

The vector `.MAILFISS.CTIP` is a vector of entières containing the list of the numbers of the meshes enriched by type Ace-Tip ("SQUARE" mesh).

4.3.7 .MAILFISS.HECT

The vector `.MAILFISS.HECT` is a vector of entières containing the list of the numbers of the meshes enriched by type Heaviside-Ace-Tip (mesh "RONDE-CARREE").

4.3.8 .MAILFISS.CONT

The vector `.MAILFISS.CONT` is a vector of entières containing the list of the numbers of the meshes on which contact is defined for the crack. They are the meshes:

- That is to say strictly intersected by the crack (standard 1 on the fig.4.3.8-1)
- That is to say presenting an edge (in 2D) or a face (in 3D) confused with the crack, and located on the side slave (standard 2 on the fig.4.3.8-1)
- Maybe, in multi-cracking, of the meshes presenting a Heaviside enrichment for the crack considered and which enter one of the first two categories for another crack (standard 3 on the fig.4.3.8-1).

Indeed, to limit the introduction of new elements during the implementation of the contact with multi-cracking, the number of degrees of freedom of contact is always equal to the number of degrees of freedom of Heaviside, since it is necessary to then putting the degrees of freedom of contact superfluous at 0. From where need for including the meshes of this third type in the group of the meshes "in contact" (but their degrees of freedom will be put at 0).

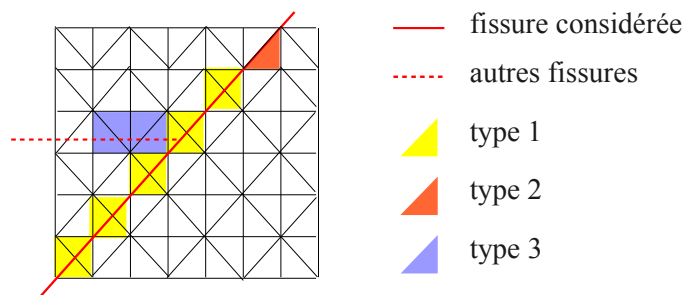


Figure 4.3.8-1 example of the group `.MAILFISS.CONT` of a crack.

4.3.9 .JONFISS

The vector `.JONFISS` the list of the cracks mothers contains on which the crack connects. It is in particular used in the operator `MODI_MODELE_XFEM` for cutting and multiple enrichment. This vector is created if the user called on the keyword `JUNCTION` in `DEFI_FISS_XFEM`.

4.3.10 .JONCOEF

The vector `.JONCOEF` is of the same length than `.JONFISS`. It contains the list of the coefficients with to associate with the level sets normals cracks of `.JONFISS` to find the field in which the connected

crack is defined. These coefficients are worth $+1$ or -1 and are calculated automatically. The connected crack is then defined in the zone where for all the cracks mothers i , products $\text{JONCOEF}(I) * \text{lsn}(\text{JONFISS}(I))$ are negative.

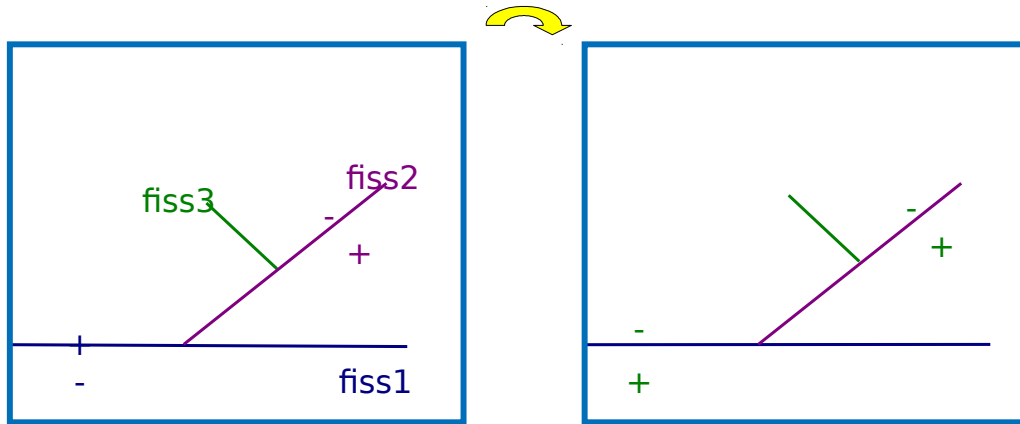


Figure 4.3.10-1 sign of the levels set of the cracks mothers on the left, sign obtained after multiplication by JONCOEF on the right, and definition of the connected crack on the side less.

One ensures thus that the connected crack is in the zone “less” compared to the cracks mothers. The figure 4.3.10-1 watch an example of use of `.JONCOEF` and of `.JONFISS` in order to define crack 3. On this figure crack 3 is connected on crack 2, it even connected on crack 1, `.JONFISS` crack 3 contains D onc cracks 1 and 2. The signs of the level sets of cracks 1 and 2 are given on the figure of left: one from of deduced the values from `JONCOEF` $(-1,1)$ as well as the répartiiti one of the sign of $\text{coef}(1) \times \text{lsn}(1)$ and $\text{coef}(2) \times \text{lsn}(2)$ on the figure of right-hand side . Crack 3 is defined in the zone where the 2 signs are less.

In conclusion `.JONFISS` and `.JONCOEF` allow to define crack 3 with its level sets usual normal and tangent as well as the level set normals of the cracks mother S, to which one has elementary access of manner.

4.4 Objects relating to the propagation

4.4.1 `.PRO.RAYON_TORE`

Vector of realities length equalizes to 1.

If this vector exists in the SD, propagation of the crack was calculated by using the localization of the field of update of the level sets. The field coincides with a torus in 3D or a circle in 2D built around the bottom of the crack. The ray of this torus or this circle is stored in this object.

If this vector does not exist in the SD, all the model was used for the calculation of the propagation of the crack.

4.4.2 `.PRO.NOEUD_TORE`

Vector the Boolean ones length equal to the number of nodes of the grid on which the crack was defined or, if a grid is associated with the crack, to the number of nodes of the grid.

If this vector exists in the SD, propagation of the crack was calculated by using the localization of the field of update of the level sets. For each node of the grid (or grid), a value `TRUE` mean that the node is understood in the field used for calculation. For example, if the value of $i^{\text{ème}}$ component of the vector is `TRUE`, it $i^{\text{ème}}$ node of the grid (or the grid) is understood in the field; on the other hand, if the value of $i^{\text{ème}}$ component of the vector is `FALSE`, it $i^{\text{ème}}$ node of the grid (or the grid) is apart from the field.

If this vector does not exist in the SD, all the model (or all the grid) was used for the calculation of the propagation of the crack.

4.4.3 .PRO.NOEUD_PROJ

Vector the Boolean ones length equal to the number of nodes of the grid on which the crack was defined.

If this vector exists in the SD, propagation of the crack was calculated on the auxiliary grid associated with the crack and the level sets calculated was projected on the grid. For each node of the grid, a value `TRUE` mean that the node is understood in the field of projection. For example, if the value of $i^{\text{ème}}$ component of the vector is `TRUE`, it $i^{\text{ème}}$ node of the grid is understood in the field; on the other hand, if the value of $i^{\text{ème}}$ component of the vector is `FALSE`, it $i^{\text{ème}}$ node of the grid is apart from the field and the values of the level sets carried by this node were not put up to date.

If this vector does not exist in the SD, propagation of the crack was calculated directly on the grid.

4.4.4 .GRI.MAILLAGE

Vector of K8 length equalizes to 1.

Name of the grid of the grid associated with the crack. If this object does not exist, no grid is associated with the crack.

4.4.5 .GRI.LTNO and .GRI.LNNO

The concept `.GRI.LTNO` (resp. `.GRI.LNNO`) is a field with the nodes (`CHAM_NO`) scalar which contains for each node of the grid the actual value of the level set tangent (resp. normal) with the crack. If this object does not exist, no grid is associated with the crack.

4.4.6 .GRI.GRLTNO and .GRLNNO

The concept `.GRI.GRLTNO` (resp. `.GRI.GRLNNO`) is a field with the nodes (`CHAM_NO`) with 3 real components. It contains for each node of the grid the values of the gradient of the level set tangent (resp. normal) in the 3 directions of space.

That is to say i i -ème node of the grid,

```
V = .GRI.GRLTNO (I) ;
```

V (1)	Value of the following gradient x level set tangent, calculated with $i^{\text{ème}}$ node
V (2)	Value of the following gradient y level set tangent, calculated with $i^{\text{ème}}$ node
V (3)	Value of the gradient following Z of the level set tangent, calculated with $i^{\text{ème}}$ node

In 2D, there are only 2 components according to x and y .

If this object does not exist, no grid is associated with the crack.

4.4.7 .FONDFISG

The vector `.FONDFISG` is a vector of real containing the coordinates of the points of the bottom of crack built on the auxiliary grid. The points are ordered according to the same method as that described in the document [R7.02.12] for creation of the vector of the points of the bottom of crack on the real grid `.FONDFISS`, so that a curvilinear X-coordinate can be defined.

If `NFON` is the number of points of the bottom of crack over the grid, then the length of the vector `.FONDFISG` is $4 \times \text{NFON}$. For each point of the bottom of crack on the grid, the first 3 components correspond to the 3 coordinates (in 3D) of the point, and the fourth component is its curvilinear X-coordinate.

This object is created only in 3D within the framework of the propagation of crack using the methods Upwind or Simplex like tools of update of the level sets.

4.5 Objects relating to the contact

These objects are created in the `sd_fiss_xfem` by `DEFI_CONTACT`.

4.5.1 .BASCO

The concept `.BASCO` is a field with the nodes (`CHAM_NO`) with 12 real components (in 2D and 3D). It contains the origin and the vectors of the COvariante Base of the facets of contact. For each node, the first three components are the coordinates of the point of contact associated with this node, which corresponds at the origin of the base covariante. The three following components are the coordinates of the 1^{er} vector of the base. The three following components are cordonnées of the 2^{ème} vector of the base. The three last components are cordonnées of the 3^{ème} vector of the base.

That is to say i $i^{\text{ème}}$ node grid,

V = .BASCO (I) ;

- V (1) Coordinate according to x point of contact associated with the node i
- V (2) Coordinate according to y point of contact associated with the node i
- V (3) Coordinate according to z point of contact associated with the node i
- V (4) Coordinate according to x from the 1^{er} vector of the base at the point of contact associated with the node i
- V (5) Coordinate according to y from the 1^{er} vector of the base at the point of contact associated with the node i
- V (6) Coordinate according to z from the 1^{er} vector of the base at the point of contact associated with the node i
- V (7) Coordinate according to x from the 2^{ème} vector of the base at the point of contact associated with the node i
- V (8) Coordinate according to y from the 2^{ème} vector of the base at the point of contact associated with the node i
- V (9) Coordinate according to z from the 2^{ème} vector of the base at the point of contact associated with the node i
- V (10) Coordinate according to x from the 3^{ème} vector of the base at the point of contact associated with the node i
- V (11) Coordinate according to y from the 3^{ème} vector of the base at the point of contact associated with the node i
- V (12) Coordinate according to z from the 3^{ème} vector of the base at the point of contact associated with the node i

In 2D, components according to z are worthless.

4.5.2 .LISEQ (addresses JLIS1)

It is `READY` you of the relations of equality (`EQuality`) between the unknown factors of contact. Vector of entiereties length `NRELEQ*2` where `NRELEQ` is the number of relations of equality. For each relation of equality, one stores the numbers of the 2 nodes belonging to the equality: That is to say IE $i^{\text{ème}}$ relation of equality,

`ZI (JLIS1-1+2* (IE-1) +1)` is the number of the 1^{er} node which is part of the equality.

`ZI (JLIS1-1+2* (IE-1) +2)` is the number of the 2^{ème} node which is part of the equality.

4.5.3 .LISEQ_LAGR (addresses JLISLA)

It is `READY` you of the numbers of LAGRange associated with the relations with equality (`EQuality`) between the unknown factors of contact. This list is built in the case of the multi-Heaviside meshes. It

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 2021 EDF R&D - Licensed under the terms of the GNU FDL (<http://www.gnu.org/copyleft/fdl.html>)

makes it possible to find the number of ddl of Lagrange to be associated with the number of a node taken in a relation of equality.

ZI (JLISLA-1+2* (IE-1) +1) is the number of Lagrange to be associated with the 1^{er} node of the relation of equality.

ZI (JLISLA-1+2* (IE-1) +2) is the number of Lagrange to be associated with the 2^{ème} node of the relation of equality.

For example if the relation should be imposed LAGS_C=LAG3_C between the nodes 'N26' and 'N32' for l^{ème} relation of equality, one will aura:

ZI (JLIS1-1+2* (IE-1) +1) = 26

ZI (JLIS1-1+2* (IE-1) +2) = 32

ZI (JLISLA-1+2* (IE-1) +1) = 1

ZI (JLISLA-1+2* (IE-1) +2) = 3

4.5.4 .CNCTE (addresses JCNTES)

They is the groups of connected vital edges. For each local number of edge in a mesh, which belongs to a group, one increments NCTE and one stores:

ZI (JCNTES-1+4* (NCTE-1) +1) the number of the edge in the group,

ZI (JCNTES-1+4* (NCTE-1) +2) the number of group,

ZI (JCNTES-1+4* (NCTE-1) +3) the number of mesh,

ZI (JCNTES-1+4* (NCTE-1) +4) the local number of the edge in the mesh.

5 Contents of the objects of `sd_modele_xfem`

5.1 Objects generals

5.1.1 `.MODELE_SAIN`

The concept `.MODELE_SAIN` of K8 length 1 container is a vector the name of the model on which is built binds model enriched (value by the keyword `MODELE_IN` as starter of `MODI_MODELE_XFEM`).

5.2 Contents of the objects relating to the concatenation

5.2.1 `.LNNO`, `.LTNO`, `.BASLOC` and `.STNO`

Attention, these objects do not have the same structure as the objects of the same name contained in `sd_fiss_xfem` ([§4.2.2], [§4.2.4] and [§4.3.3]). It is of the elementary fields of type `ELNO` and not of `CHAM_NO`. The principal difference is that them `CHAM_NO` store information on the nodes of the grid (a value by node) whereas them `CHAM_ELNO` store information by node on the element. A node being connected to N elements, the value will be stored N time. This choice has as a finality the information storage of several cracks in an element, one then uses the concept of under point.

If an element is contained in `SD fiss (I) / '.MAILFISS.HEAV'`, it is said that he is seen by crack I : it is then necessary to store in the model the relative information with this crack. If the element sees 2 cracks or more, it is necessary to store in the model information of all the cracks. By defining the number of under points $NBSP$ `CHAM_ELNO` like the number of cracks seen by the element, one duplicates $NBSP$ time the memory capacity relating to the SD for the element which sees $NBSP$ cracks. Thus one can store information of all the cracks in this element.

The figure 5.2.1-1 watch an example of concatenation of 2 cracks in the model. One represents on this figure the SD `.LNNO` of the 2 cracks and that of the model. The element of the center sees 2 cracks, the field `MODEL` thus contains 2 components per node in this element.

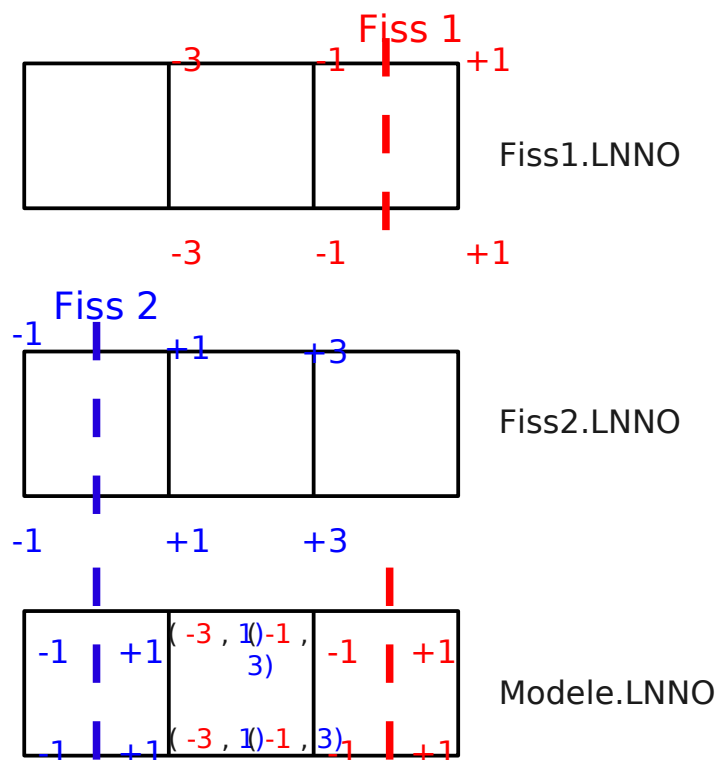


Figure 5.2.1-1 Example of storage of multiple SD in the model.

5.2.2 .NOXFEM

It is about one `CHAM_NO` with 2 components. The components are affected if the node has an enrichment X-FEM (i.e it exists a crack `fiss (I)` for which the field `fiss (I)/'.STNO'` is not null in this node). The first component corresponds to the number of mesh X-FEM containing the node. The second component is the local number of the node in this mesh X-FEM. This structure is useful for the imposition of limiting conditions of Dirichlet type.

5.2.3 .FISSNO

It is about one `CHAM_ELNO`. This SD is created when at least an element of the structure must store information of several cracks. For the introduction of several degrees of Heaviside freedom, one made the choice sequentially to increment by nodes the numbers of degrees of freedom. For example figure 5.2.3-1, two cracks are introduced. One affects for the first crack (in blue on the left) the ddl `H1`. For the second crack (in red in the center) one allots degrees of freedom `H1` with the nodes which do not have already degrees of freedom `H1` affected previously, and of the degrees of freedom `H2` for the nodes having degrees of freedom already `H1`. This choice is made with an aim of not weighing down the elementary catalogues. Indeed this strategy does not require of having element only `H2`.

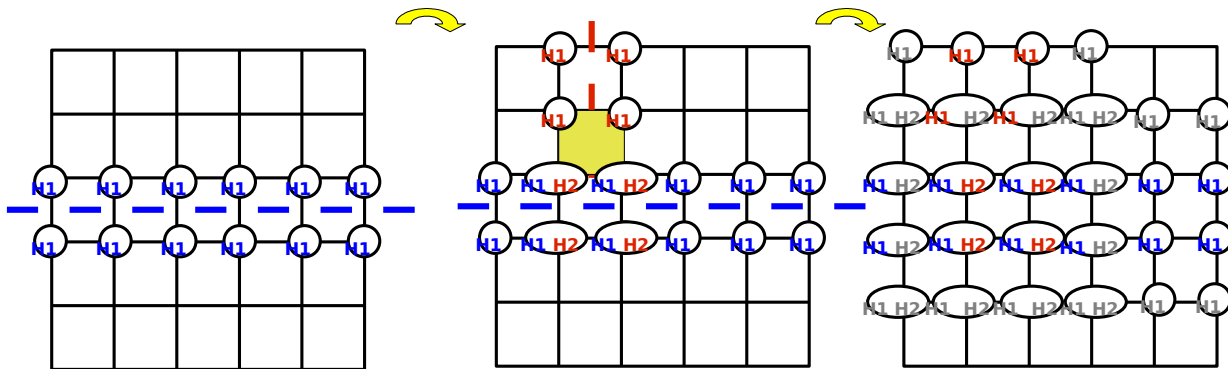


Figure 5.2.3-1 Degrees of Heavisides freedom, for two cracks which overlap

However attention should be paid, because in certain elements, the incrementing of the cracks does not ensure a direct link between the number of total degree of Heaviside freedom and the number of crack (In the example, $H1$ can correspond to crack 1 or crack 2 in the yellow element). During the assembly in TE, it is thus necessary to be able to associate the degrees of freedom $H_{IFH}[X, Y, Z]$ (with IFH the number of total degree of Heaviside freedom) of the node INO with the right number of crack. Moreover elements must be supplemented by degrees of Heaviside freedom which are then eliminated (degrees of freedom in gray on the figure of right-hand side). It is necessary to eliminate degrees of freedom $H_{IFH}[X, Y, Z]$ node INO to have the statute of the good crack in INO .

The SD MODEL `' .FISSNO'` allows to do that. It turns over the number of crack $IFISS$ considering locally with the node INO for the degrees of freedom $H_{IFH}[X, Y, Z]$. One has $IFISS = FISSNO(INO, IFH)$. One recovers then the good pointer for MODEL SD `' [.LNNO, .STNO .TOPOSE.HEA]`.

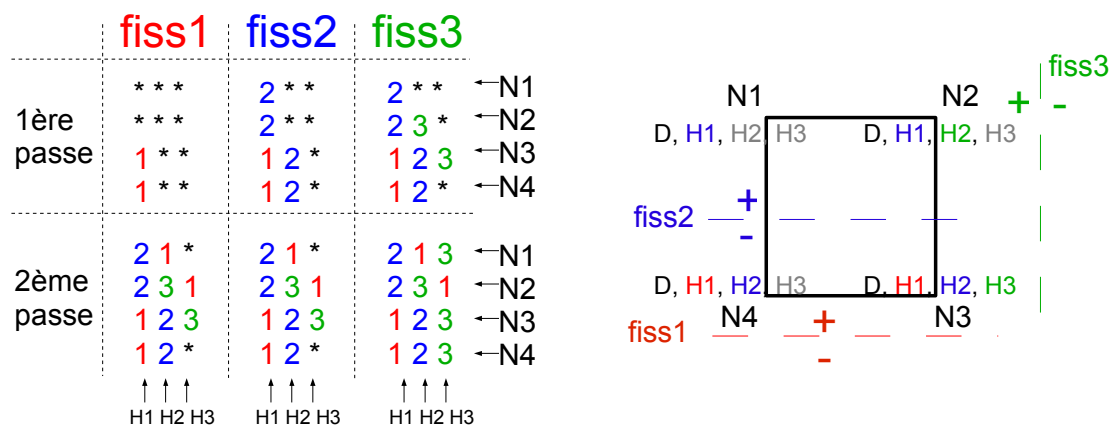


Figure 5.2.3-2 Example of construction of the SD Modèle.FISSNO

The figure 5.2.3-2 illustrate the construction of the table `.FISSNO`. The lines correspond to the numbers of node, and the column with the numbers of degree of Heaviside freedom. One fills out the table in a sequential way on the 2 ways. At the time of the first master key, one looks at the statute (`MODELE.STNO`) node INO for the crack $IFISS$. If the statute is not null, one fills the first box by the data IFH in their order of course with INO who was not already affected with $IFISS$. One

supplements the table so necessary at the time of the second master key while proceeding as for the first master key but by filling the box by the data *IFH* in *INO* if the statute is null.

To illustrate the use of this SD, one takes again the simpler example of the figure 5.2.1-1 on the figure 5.2.3-3 . One then built `Modèle.FISSNO` for the element of the medium. If one wants (during the assembly for example) to recover the Heaviside function associated with the node *N1* for the degrees of freedom $H_1[X, Y]$. One recovers the number of crack via $IFISS = FISSNO(INO=1, IFH=1)=2$. There is then the Heaviside function which is worth $HE(IFISS=2)=+1$. For the node *N4* , one will have $FISSNO(4,1)=1$ and thus $HE(IFISS=1)=-1$.

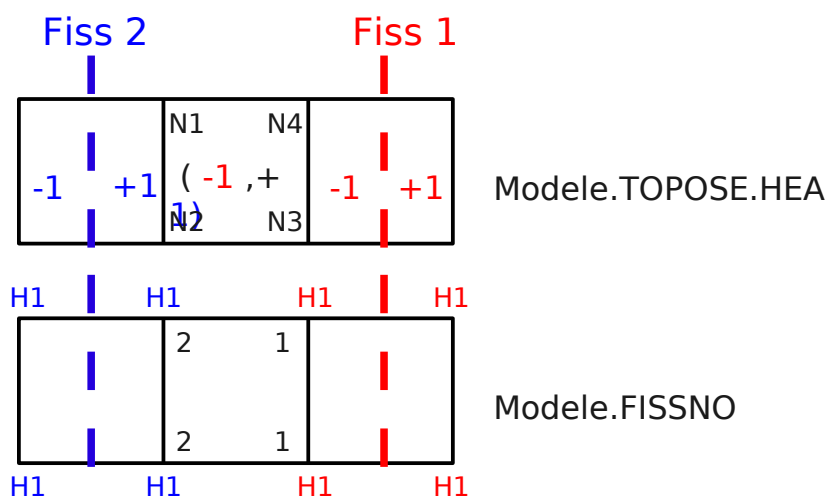


Figure 5.2.3-3 Example for the SD Modèle.FISSNO

More generally, the Heaviside function under element of integration *SE* , for a node *INO* and a degree of Heaviside freedom *IFH* is obtained in the following way:

$$HE(SE, INO, IFH) = HE(SE, IFISS(INO, IFH)) .$$

This structure of data also makes it possible to remove the degrees of Heaviside freedom in excess in *TE* . One recovers the statute (`MODELE.STNO`) node *INO* for the degree of Heaviside freedom *IFH* , whereas this structure is stored by crack:

$$STATUT(INO, IFH) = STATUT(INO, IFISS(INO, IFH)) .$$

This structure is also used for postprocessing X-FEM, during the interpolation of displacements on the lips of the crack.

5.2.4 .HEAVNO

It is the structure reverses `.FISSNO`. This SD is created when at least an element of the structure must store information of several cracks. It is built at the same time as `.FISSNO` and L turns overE number of degree of Heaviside freedom associated with the node *INO* and with the crack *IFISS* . By keeping the notations of the paragraph preceding one a:

$$IFH = HEAVNO(INO, IFISS)$$

In the case of the multi-Heaviside contact, the number of degree of freedom of Lagrange associated with a crack *IFISS* corresponds to the number of degree of Heaviside freedom *IFH* corresponding. Thus this structure of data makes it possible it to find the name of the degree of freedom of Lagrange (and its position in the elementary matrix/vector) starting from the number of crack. If one treats the contact on one facet of contact associated with the crack *IFISS* , the Lagrangian one which works with the node *INO* carry the number:

$$ILAG = HEAVNO(INO, IFISS)$$

5.3 Contents of the objects relating to the topology of the subelements

Notes with the developers:

- certain values *ma* are in hard dan FORTRAN (*te0514, xbsigm, xmel3d...*)
- one authorizes 4 cracks to the maximum for modelings multiheaviside
- the lengths of the structures of data are thus multiplied by 4 of mono-Heaviside with multi-Heaviside

Recall on the way in which elements X-FEM are cut out (R7.02.12 Document left 3.3 Pennies cutting):

	3D	2D
Cutting ⁽¹⁾ Preliminary phase of cutting	Each element (hexahedron, pentahedron) is virtually cut out in tetrahedrons 6 tetrahedrons to the maximum	Each element (quadrangle) is virtually cut out in triangles 2 triangles to the maximum
Cutting ⁽²⁾ Under-cutting	Each tetrahedron is in its turn cut out under-tetrahedrons 6 under-tetrahedrons to the maximum	Each triangle is in its turn cut out under-triangles 3 under-triangles to the maximum
Points of intersection	Points of intersection enters the edges of the tetrahedrons resulting from cutting ⁽¹⁾ and crack surfaces it 11 points of intersection to the maximum	Points of intersection enters the edges of the triangles resulting from cutting ⁽¹⁾ and the curve of the crack 3 points of intersection to the maximum
Points mediums	Points in the middle of the edges of the triangles resulting from cutting ⁽²⁾ [on condition that coinciding with no node of the complete initial element] 66 points to the maximum (59 points mediums to the maximum distinct from the central nodes and 7 central points to the maximum)	Points in the middle of the edges of the triangles resulting from cutting ⁽²⁾ [on condition that coinciding with no node of the initial element] 11 points mediums to the maximum (10 points mediums to the maximum distinct from the central nodes and 1 central point to the maximum)
Point central of quadratic quadrangle	Storage of the points mediums of the faces quadrangle of the quadratic polyhedron. 7 central points to the maximum	Point of intersection of the two right diagonals of a quadratic quadrangle 1 central point to the maximum
Process of cutting ⁽¹⁾ and under-	1 element X-FEM 3D	1 element X-FEM 2D

cutting ⁽²⁾	tetrahedron n°1 6 under-tetrahedrons with the max. one. tetrahedron n°2 6 under-tetrahedrons with the max. one. ...	triangle n°1 3 under-triangles with the max. one. triangle n°2 3 under-triangles with the max. one.
------------------------	---	--

(2)

Note: the number of under elements is not modified by the introduction of quadratic meshes.

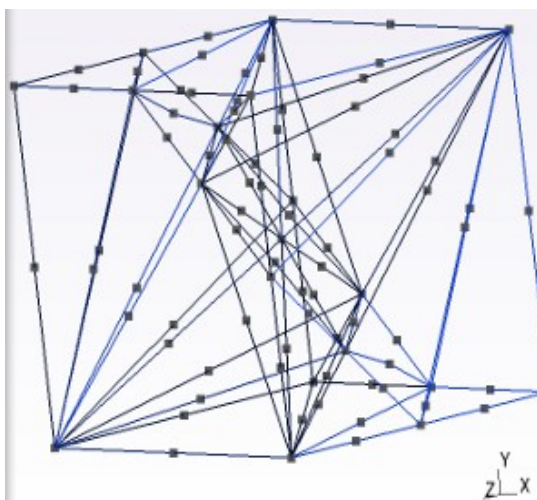


Figure 5.3-1 : configuration generating the maximum of point mediums/points of intersection

5.3.1 .TOPOSE.PIN

Inform about `PRUBBED` with oil of `INTERSECTION`.

It is a constant field by element (`CHAM_ELEM`) with 33 real components.

For each element, this field comprises the coordinates of the points of intersection (with the direction previously definite). As the maximum number of such points is 11, and considering one works in dimension 3, the component count per element is dimensioned to 33. For each element, the components are ordered like this:

$\{x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_{11}, y_{11}, z_{11}\}$ where x_i, y_i, z_i are the coordinates of $i^{\text{ème}}$ point of intersection on the element.

It will be noted that if the element is not fissured, the components are worthless.

In 2D, one has to the maximum 3 points of intersection distincts of the nodes tops, with 2 components each one, therefore 6 components on the whole.

For the multiheaviside:

In 3D, the field requires $33 \times 4 = 132$ real components.

In 2D, the field requires $6 \times 4 = 24$ real components.

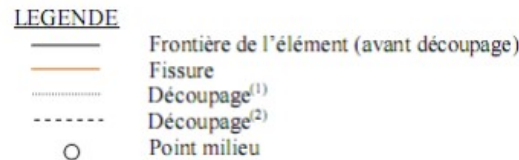
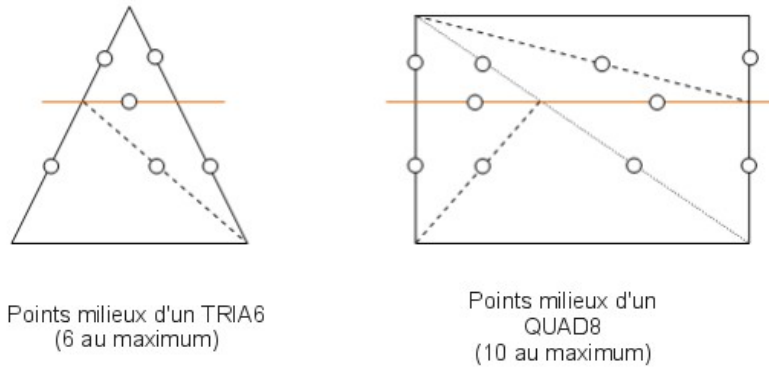
5.3.2 .TOPOSE.PMI

Inform about `PRUBBED` with oil Mediums in the presence of quadratic meshes.

5.3.2.1. 2D

It is a constant field by element (CHAM_ELEM) with 22 real components. For each element, this field comprises:

- coordinates of the points mediums



Note: O N recalls that the coordinates of the points mediums which coincide with a node of the element relative are not stored in this structure of data.

- coordinates of the central point introduced during cutting⁽¹⁾ of a QUAD8:

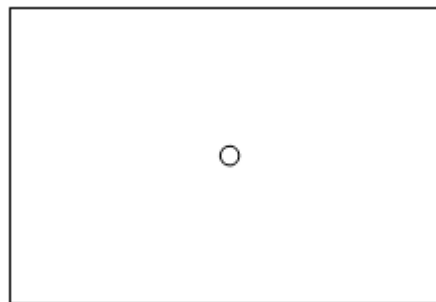


Figure 5.3.2.1-1: Point central of a QUAD8 (1 at the most)

That makes a total of 11 points to the maximum. Like one places oneself in 2D, one can have $11 \times 2 = 22$ components to the maximum. The field with 22 thus is dimensionned.

For each element, the components are ordered like this:

$$\{x_1, y_1, x_2, y_2, \dots, x_{11}, y_{11}\}$$

where $(x_i, y_i)_{1 \leq i \leq 10}$ are the coordinates of $i^{\text{ème}}$ not medium on the element and where (x_{11}, y_{11}) are the coordinates of the central node of a QUAD8.

5.3.2.2. 3D

It is a constant field by element (`CHAM_ELEM`) with 198 real components. For each element, this field comprises:

- coordinates of the points mediums (66 points to the maximum),
- coordinates of the central points introduced during the cutting of the quadratic element 3D (7 additional central points to the maximum, for example, during the cutting of one `HEXA20` via the complete element `HEXA27`).

For each element, the components are ordered like this:

$$\{x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_{65}, y_{65}, z_{65}\}$$

5.3.3 .TOPOSE.CNS

Inform about the connectivity of the subelements.

5.3.3.1. 3D

It is a constant field by element (`CHAM_ELEM`) with 320 whole components. This field is dimensioned compared to the under-cutting of the quadratic elements in `TETRA10`. For, a linear under-cutting, it is thus oversized, without notable consequence on calculation and the performances.

For each element, this field contains the connectivity of the subelements of the element. By convention, all the subelements are tetrahedrons with 10 nodes (`TETRA10`), and there can be to the maximum 32 per element X-FEM (for a detailed explanation of this result, to see the reference material on method X-FEM [R7.02.12]). There are to the maximum 6 tetrahedrons and 6 subelements per tetrahedron, but there is no configuration with 6 subelements for each tetrahedron, this is why the maximum is of 32. The components are arranged 4 by 4. The object `.TOPOSE.CNS` refers to 3 types of Nœuds: either of the already existing nodes of the grid, or of the points of intersection, or of the nodes mediums. Thus, for each element:

- if a top of a subelement is a node of element X-FEM in question, then this node is located by its local number in the element;
- if a top is a point of intersection, then this point is located by a number higher than 1000, where the figure of the units returns to its local number in the list of the points of intersection of the element.
- concerning the nodes mediums: if the node medium is a node of the element relative, one stores the local number of the node in the element relative; if it is about a new node medium generated by cutting, the point is then located by a number higher than 2000. If the point medium is also a central point of the element relative, it is then located by a number higher than 3000.

Example:

For the element `IE` :

`CNS (IE) =`

1001	1005	6	1002	2001	12	2005	3012	3021	2004	1003	6	...
------	------	---	------	------	----	------	------	------	------	------	---	-----

1^{er} under-tétra: Nœud n°1: 1001 -> returns to the 1^{er} not `PIN (IE)`

Nœud n°2: 1005 -> returns to the 5^{ème} not `PIN (IE)`

Nœud n°3: 6 -> returns to the 6^{ème} node of the element `IE`

Nœud n°4: 1002 -> returns to the 2^{ème} not `PIN (IE)`

Nœud n°5: 2001 -> returns to the 1^{ème} not `MILLET (IE)`

Nœud n°6: 12 -> returns to the 12^{ème} node of the element `IE`

Nœud n°7: 2005 -> returns to the 5^{ème} not `MILLET (IE)`

Nœud n°8: 3012 -> returns to the 12^{ème} not `MILLET (IE)`

Nœud n°9: 6 -> returns to the 6^{ème} node of the element `IE`

Nœud n°10: 3021 -> returns to the 21^{ème} not `MILLET (IE)`

2^{ème} under-tétra: top n°1: 1003 -> returns to the 3^{ème} not PIN (IE)
top n°2: 6 -> returns to the 6^{ème} node of the element IE
...

5.3.3.2. 2D

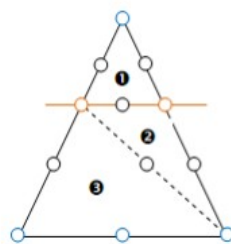
It is a constant field by element (CHAM_ELEM) with 36 whole components. This field is dimensioned compared to the under-cutting of the quadratic elements in TRIA6.

For each element, it contains the connectivity of under elements of the element.

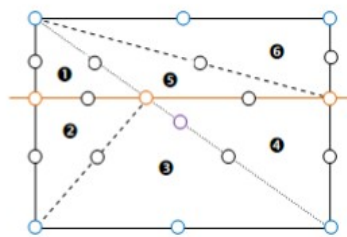
One counts:

- 6 under-triangles to the maximum (reached for a mesh quadrangle)
- 6 nodes to the maximum for each under-triangle (reached for a quadratic mesh)

What makes $6 \times 6 = 36$ components to the maximum. The field is thus dimensioned to 36 and one arranges these last 6 by 6.



Sous-découpage d'un TRIA6
3 sous-triangles au maximum
6 nœuds pour chacun
= 18 nœuds à répertorier



Sous-découpage d'un QUAD8
6 sous-triangles au maximum
6 nœuds pour chacun
= 36 nœuds à répertorier

LEGENDE

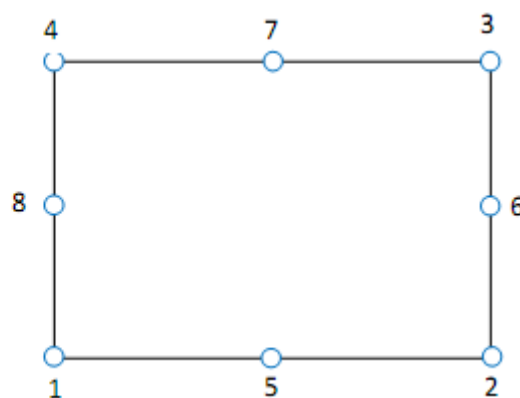
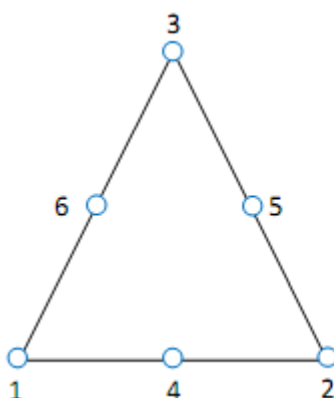
— Frontière de l'élément (avant découpage)
— Fissure
- - - Découpage⁽¹⁾
- - - Découpage⁽²⁾
❶ N° sous-triangle

○ Point d'intersection
○ Point milieu
○ Point central
○ Nœud de l'élément

The object .TOPOSE.CNS refers to 4 types of points:

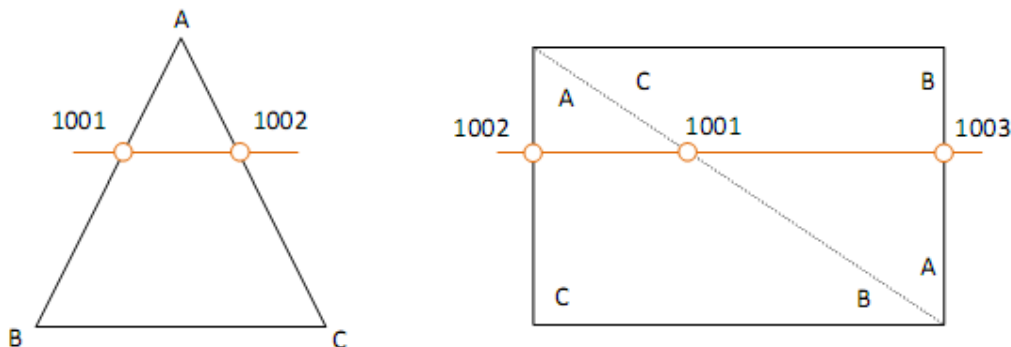
- Type 1: Node of the element (pertaining to the grid)

It is located by its local number in the element



○ Type 2: Point of intersection

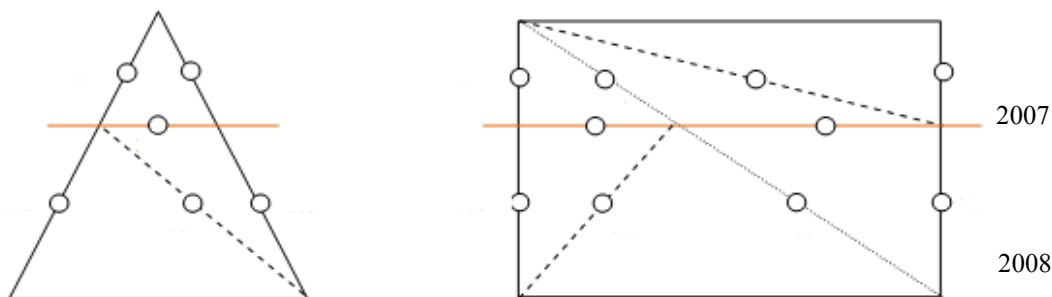
It is located by a number $1000+u$, where u return to its local number in the list of the points of intersection of the element.



1001 → 1^{er} not stored in PINTTO
1002 → 2^{eme} not stored in PINTTO...

○ Type 3: Not medium

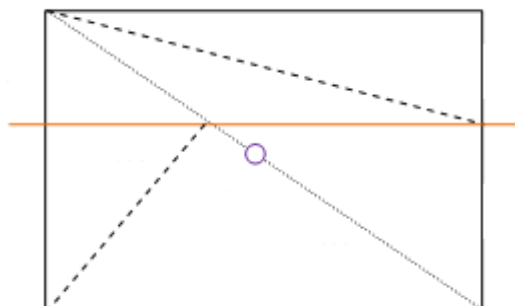
It is located by a number $2000+u$, where u return to its local number in the list of the points mediums of the element.



200 1 → 1^{er} not stored in PMILTO
200 2 → 2^{ème} not stored in PMILTO, etc.

○ Type 4: Not quadrangle medium

It is located by a number $3000+u$, where u return to its local number in the list of the points mediums of the element.



3011 → 11^{ème} not stored in PMILTO

5.3.3.3. Case multiheaviside

For modelings multiheaviside, one authorizes only one linear under-cutting. One then uses dimensions of the tables of connectivity of linear cutting (3D: TETRA4 instead of TETRA10/2D: TRIA3 instead of TRIA6).

Dimensions of the tables of connectivity monoheaviside are then multiplied by the maximum number of cracks:

In 3D, $128 \times 4 = 512$ whole components.

In 2D, $18 \times 4 = 72$ whole components.

5.3.4 . TOPOSE . HEA

Inform about the value of the function HEAVISIDE.

It is one CHAM_ELEM constant by subelement with 32 whole components corresponding to the value of the Heaviside function on the under-tetrahedrons (+1 or - 1), successively arranged.

There can be of the 0 if the full number of subelements is < 32 (number max of under-tetrahedrons).

In 2D, the subelements are with the maximum number of 6, the field TOPOSE.HEA is thus dimensioned to 6.

In the case or the element sees several cracks, this field is duplicated as many times as there are cracks. One then uses the number of under points NBSP defines in [the §5.2.1]. The field will be of $32 \times NBSP$ components in 3D and of $6 \times NBSP$ components in 2D.

5.3.5 . TOPOSE . LON

Inform about the value Length of fields used to define the subelements.

It is a constant field by element (CHAM_ELEM) with a whole component, in 2D as in 3D. This component corresponds to the number of subelements contained in element X-FEM.

5.3.6 . TOPOSE . PAI

Inform about the intersected edges. This structure of data passed to the option TOPOFA so that it is not recomputed to generate the structure of data .TOPOFAC.AI

Its structure is similar to .TOPOFAC.AI :

It is a constant field by element (CHAM_ELEM) to 15 or 55 real components containing of information on the intersected edges.

For each point of intersection of each element:

The 1^{ère} component is the local number of the corresponding edge (0 if it is a node top).

2^{ème} component is the local number of the node if it is a node top (0 if not).

3^{ème} component is the length of the edge.

4^{ème} component is the position of the point of intersection on the edge, with an arbitrary direction which depends on the mesh. This reality strictly lies between 0 and 1, and is worth 0 if the point of intersection is a node top.

5^{ème} component makes it possible to know if the intersected edge is vital or not. If it is vital, the component is worth 1, if it is not it, the component is worth 0.

The component count per point of intersection (5) is accessible in FORTRAN by a call to the function XXMMVD ('ZXAIN')

That is to say i ^{ème} element grid,

V = .TOPOFAC.AI (I) ;

V (1) Local number of the edge corresponding to the 1^{er} point of intersection
V (2) Local number of the node corresponding to the 1^{er} point of intersection
V (3) Length of the edge corresponding to the 1^{er} point of intersection
V (4) Position of the 1^{er} point of intersection on the edge
V (5) 1^{er} point of intersection: so on vital edge (1) if not (0)
V (6) Local number of the edge corresponding to the 2^{ème} point of intersection
...
V (55) 11^{ème} vital point of intersection (1) or not (0)

If the number of points of intersection is strictly lower than the maximum number of points of intersection (3 or 11 following cases), the vector is supplemented by 0.

In 2D, there are to the maximum 3 points with in the same way that in 3D 5 information (number of the edge, number of the node, length of the edge and position of the point of intersection, vital intersected edge or not). The field `TOPOFAC.AI` thus has 15 components per element. In 3D there are 55 components.

5.3.7 .TOPOSE .PJO

This SD is used only for the elements the cohesive multi-Heaviside HM-XFEM. It is about one `CHAM_ELEM` . who comprises a whole component by node of the element. It thus has to 20 components. This component is worth 0 if the node does not contain a junction of crack on its support and 1 if not. This information is then used to impose the continuity of the pressure of fluid in the branches of interfaces on the level of a junction of hydraulic cracks (confer [R7.02.18]).

5.4 Objects relating to the topology of the facets of contact

Fields `.TOPOFAC.PI`, `.TOPOFAC.AI`, `.TOPOFAC.CF`, `.TOPOFAC.LO` and `.TOPOFAC.BA`, are duplicated by element as many times as the number of cracks seen by the mesh. One uses for that the number of under-points which corresponds to n_{fiss} .

5.4.1 .TOPOFAC .LO

Inform about `LONGUOR` of the fields used for the contact.

It is a constant field by element (`CHAM_ELEM`) with 3 whole components, containing the number of points of intersection (`NINTER`), the number of triangular facets of contact (`NFACE`) as well as the number of points (nodes tops and nodes mediums) per triangular facet of contact (`NPTF`).

5.4.2 .TOPOFAC .PI

Inform about `PRUBBED` with oil ofIntersection.

It is a constant field by element (`CHAM_ELEM`) who comprises to 102 real components, which are the coordinates **within the space of reference relative** points of intersection of the edges of the element with the crack.

In 2D, one has up to 7 points of intersection (including `Nœuds mediums` and `Nœuds tops`) thus the field is dimensioned to 14. In 3D, one has some up to 34 thus the field is dimensioned to 102.

For the Heaviside elements, there are to the maximum 18 points of intersection in 3D, therefore the field is dimensioned to 54, and 8 points of intersection in 2D, therefore the field is dimensioned to 16.

That is to say i ^{ème} element grid,

V = .TOPOFAC.PI (I) ;

V (1)	Coordinate of reference according to x from the 1 ^{er} point of intersection
V (2)	Coordinate of reference according to y from the 1 ^{er} point of intersection
V (3)	Coordinate of reference according to z from the 1 ^{er} point of intersection
V (4)	Coordinate of reference according to x from the 2 ^{ème} point of intersection
V (5)	Coordinate of reference according to y from the 2 ^{ème} point of intersection
...	...
V (102)	Coordinate of reference according to z from the 34 ^{ème} point of intersection

If the number of points of intersection is strictly lower than the maximum number of points of intersection (7, 8, 18 or 34 following cases), the vector is supplemented by 0. The number of points of intersection is contained in the vector `.TOPOFAC.LO` (see §5.4.1).

5.4.3 .TOPOFAC.AI

Inform about `Withrêtes Intersectées`.

It is a constant field by element (`CHAM_ELEM`) to 35 or 170 real components containing of information on the intersected edges.

For each of the 7 or 34 points of intersection of each element:

The 1^{ère} component is the local number of the corresponding edge (0 if it is a node top).

2^{ème} component is the local number of the node if it is a node top (0 if not).

3^{ème} component is the length of the edge.

4^{ème} component is the position of the point of intersection on the edge, with an arbitrary direction which depends on the mesh. This reality strictly lies between 0 and 1, and is worth 0 if the point of intersection is a node top.

5^{ème} component makes it possible to know if the intersected edge is vital or not. If it is vital, the component is worth 1, if it is not it, the component is worth 0.

The component count per point of intersection (5) is accessible in FORTRAN by a call to the function `XXMMVD ('ZXAIN')`

That is to say i i ^{ème} element grid,

```
V = .TOPOFAC.AI (I);
```

V (1)	Local number of the edge corresponding to the 1 ^{er} point of intersection
V (2)	Local number of the node corresponding to the 1 ^{er} point of intersection
V (3)	Length of the edge corresponding to the 1 ^{er} point of intersection
V (4)	Position of the 1 ^{er} point of intersection on the edge
V (5)	1 ^{er} point of intersection: so on vital edge (1) if not (0)
V (6)	Local number of the edge corresponding to the 2 ^{ème} point of intersection
...	...
V (170)	34 ^{ème} vital point of intersection (1) or not (0)

If the number of points of intersection is strictly lower than the maximum number of points of intersection (7 or 34 according to the cases), the vector is supplemented by 0.

In 2D, there are to the maximum 7 points (4 4 point medium and points of intersection) with in the same way that in 3D 5 information (number of the edge, number of the node, length of the edge and position of the point of intersection, vital intersected edge or not). The field `TOPOFAC.AI` thus has 35 components per element. In 3D, one has up to 34 points of intersection, the field `TOPOFAC.AI` thus has 170 components per element.

5.4.4 .TOPOFAC.CF

Inform about `Connectivity of Facettes of contact`.

It is a constant field by element (`CHAM_ELEM`) to 9 or 90 whole components, containing the connectivity of the tops of the triangular facets. There are to the maximum 15 facets per element.

That is to say i $i^{\text{ème}}$ element grid,

```
V = .TOPOFAC.CF (I);

V (1)      Local number of the 1er top of the 1era triangular facet
V (2)      Local number of the 2ème top of the 1era triangular facet
V (3)      Local number of the 3ème top of the 1era triangular facet
V (4)      Local number of the 1er top of the 2ème triangular facet
V (5)      Local number of the 2ème top of the 2ème triangular facet
...
V (90)     Local number of the 3ème top of the 15ème triangular facet
```

If the number of triangular facets is strictly lower than 15, the vector is supplemented by 0.

In 2D linear, one has to 3 facets of contact, which are segments with two tops and possibly a node medium. The field `TOPOFAC.CF` thus has 9 components per element. In 3D, one has up to 15 facets which are triangles with 3 nodes tops and possibly 3 nodes mediums. The field `TOPOFAC.CF` thus has 90 components per element.

5.4.5 .TOPOFAC.BA

Inform about `BACovariante` of the facets of contact.

It is a field by element (`CHAM_ELEM`) to 28 or 306 real components which are the coordinates of the vectors of the base covariante (n, τ_1, τ_2) for each of the 7 or 34 points of intersection.

For each element (the exhibitor who refers to the number of the point of intersection):

$$\left[n_x^1, n_y^1, n_z^1, \tau_{1x}^1, \tau_{1y}^1, \tau_{1z}^1, \tau_{2x}^1, \tau_{2y}^1, \tau_{2z}^1, n_x^2, \dots, \tau_{2z}^7 \right]$$

That is to say i $i^{\text{ème}}$ element grid, and j the number of the point of intersection.

```
V = .TOPOFAC.BA (I);

V (9* (j-1) +1)  Coordinate according to x from the 1er vector of the base with jème point of
                  intersection
V (9* (j-1) +2)  Coordinate according to y from the 1er vector of the base with jème point of
                  intersection
V (9* (j-1) +3)  Coordinate according to z from the 1er vector of the base with jème point of
                  intersection
V (9* (j-1) +4)  Coordinate according to x from the 2ème vector of the base with jème point of
                  intersection
V (9* (j-1) +5)  Coordinate according to y from the 2ème vector of the base with jème point of
                  intersection
V (9* (j-1) +6)  Coordinate according to z from the 2ème vector of the base with jème point of
                  intersection
V (9* (j-1) +7)  Coordinate according to x from the 3ème vector of the base with jème point of
                  intersection
V (9* (j-1) +8)  Coordinate according to y from the 3ème vector of the base with jème point of
                  intersection
V (9* (j-1) +9)  Coordinate according to z from the 3ème vector of the base with jème point of
                  intersection
```

If the number of points of intersection is strictly lower than the maximum number of points of intersection (7 or 34 following cases), the vector is supplemented by 0.

In 2D, the base covariante consists of 2 vectors (n, τ_1) having each one 2 components and there is up to 7 points of intersection. The component count is thus 28. In 3D, the base covariante consists of 3 vectors (n, τ_1, τ_2) having each one 3 components and there is up to 34 points of intersection. The component count is 306.

5.4.6 TOPOFAC.HE

Inform about the value of the Heaviside functions on both sides of the facet of contact. This SD is useful only in the case of multi-Heaviside elements. It is not created if the model does not have such elements. That is to say n_{fiss} the number of cracks seen by the mesh. The number of under-points for this object is then n_{fiss}^2 (for each crack, one will want to know the values coming from the other cracks). The component count is of $2 \times n_{face}$ with n_{face} the preset maximum number of facets of contact is $n_{face} = 1$ in 2D and $n_{face} = 5$ in 3D.

That is to say i $i^{ème}$ element grid, j the number of facet for the crack k

V = .TOPOFAC.HE (I);

V (2*nface* (nfiss* (k-1) +1-1) +2* (j-1) +1)	Value of the Heaviside function of crack 1 on the side slave of the facet
V (2*nface* (nfiss* (k-1) +1-1) +2* (j-1) +2)	Value of the Heaviside function of crack 1 on the main side of the facet
V (2*nface* (nfiss* (k-1) +2-1) +2* (j-1) +1)	Value of the Heaviside function of crack 2 on the side slave of the facet
V (2*nface* (nfiss* (k-1) +2-1) +2* (j-1) +2)	Value of the Heaviside function of crack 2 on the main side of the facet
...	
V (2*nface* (nfiss* (k-1) +n _{fiss} -1)+2* (j-1) +1)	Value of the Heaviside function of crack N _{fiss} on the side slave of the facet
V (2*nface* (nfiss* (k-1) +n _{fiss} -1)+2* (j-1) +2)	Value of the Heaviside function of crack N _{fiss} on the main side of the facet

This SD is useful to calculate the jump of displacement between the sides Master and slave in the case of the multi-Heaviside elements. Later on, this SD will be transformed into TOPONO.HFA to facilitate the identification of the 2 fields being next to the under-facets. This identification will then allow the evaluation of the functions characteristic of fields associated with the new approximation with the jump with displacement, like specifying in the reference material [R7.02.12].

5.4.7 .TOPOFAC.OE

It is about a field of which the data-processing structure is rigorously the same one as TOPOFAC.PI. As this last it contains the coordinates of the points of intersection in real space, but **in the initial configuration**. This field serves within the framework of the approach great slips with X-FEM.

5.4.8 .TOPOFAC.GE

It is about a field of which the data-processing structure is rigorously the same one as TOPOFAC.OE. As this last it contains the coordinates of the points of intersection in real space, but **brought up to date according to the kinematics of the facets slaves**. This field serves within the framework of the approach great slips with X-FEM.

5.4.9 .TOPOFAC.GM

It is about a field of which the data-processing structure is rigorously the same one as `TOPOFAC.OE`. As this last it contains the coordinates of the points of intersection in real space, but **brought up to date according to the kinematics of the facets Masters**. This field serves within the framework of the approach great slips with X-FEM.

5.5 Objects relating to the definition of the functions characteristic of fields

5.5.1 `TOPONO.HNO`

This field `ELNO` contains the “whole” codes representing the numbers identifiers of field by node and element. Each Nœud Heaviside nouveau riche can represent to 4 cracks to the maximum by element. One watch in an incremental way, that 4 cracks/junctions define to the maximum 5 fields per element.

To thus describe the environment of each node X-FEM (Heaviside), one thus allocates 5 sites per node and element, in the table `TOPONO.HNO(NNO*5)`, filled in the following way:

<code>V = .TOPONO.HNO (I) ;</code>	
<code>V (1)</code>	Whole code representing it 1 ^{er} field seen by the 1 ^{er} node of the element
<code>V (2)</code>	Whole code representing it 2 ^{ème} field seen by the 1 ^{er} node of the element
<code>V (3)</code>	Whole code representing it 3 ^{ème} field seen by the 1 ^{er} node of the element
<code>V (4)</code>	Whole code representing it 4 ^{ème} field seen by the 1 ^{er} node of the element
<code>V (5)</code>	Whole code representing it 5 ^{ème} field seen by the 1 ^{er} node of the element
<code>V (6)</code>	Whole code representing it 1 ^{er} field seen by the 2 ^{ème} node of the element
<code>V (7)</code>	Whole code representing it 2 ^{ème} field seen by the 2 ^{ème} node of the element
<code>V (8)</code>	Whole code representing it 3 ^{ème} field seen by the 2 ^{ème} node of the element
<code>V (9)</code>	Whole code representing it 4 ^{ème} field seen by the 2 ^{ème} node of the element
<code>V (10)</code>	Whole code representing it 5 ^{ème} field seen by the 2 ^{ème} node of the element
<code>...</code>	<code>...</code>
<code>V (5*N)</code>	Whole code representing it 5 ^{ème} field seen by the NR ^{ème} node of the element

In addition, concerning the organization of the storage of the 5 fields seen by each Nœud: the first 4 sites are reserved for the “complementary” fields of the field to which N belongsœud, the last site is reserved for the field to which the node belongs.

When the crack passes by the node, the selection of the field to which the node belongs, takes place automatically thanks to the function “signs”. By convention one a: $He(Isn=0)=sign(0)=1$. This convention makes it possible to prolong by continuity the coding of the fields on the edge of the elements and consequently with the nodes.

Let us recall that the coding of the fields is done by projection bases 4 of the functions of them signs Heaviside at the point **P** running (either a node, or a point of gauss):

$$code(\underline{P}) = \sum_{ifiss=1}^{nfiss} 4^{nfiss-ifiss} (He(\underline{P}, ifiss) + 2)$$

Using the `SD .LNNO` and `.TOPOSE.HEA`, for each node “Heaviside”, one then calculates the code of the field to which belongs the node and the code of the fields seen by the node (partitionnés in the subelements).

In addition, being given the local construction of the concaténé field `.LNNO`, the same field can have several codes of an element with another, because a node can see a number different from cracks of an element to another (even figure 5.2.3-3). That involves a notable challenge in the organization of field `TOPONO.HNO` so that coding keeps a nonlocal significance, in light, so that

information `TOPONO.HNO(jno,i)` always represent the intersection of the support of node $N^\circ jno$, with the field Ω_i whatever the element considered.

To circumvent this difficulty, we define polymorphic fields rather. In other words, the same field will be able to have one code different from a Heaviside element to another. As a result, the code `TOPONO.HNO(jno,i)` represent the intersection of the support of node $N^\circ jno$ with a field Ω_i , “seen element running” .

This “elementary” construction is then in adequacy with the whole of elementary calculations, including for the late meshes [R5.03.53].

5.5.2 TOPONO.HSE

This elementary field contains a “whole” code by subelement X-FEM. Its dimensioning leans then on the number of subelement resulting from under-cutting cf. `TOPOSE.LON` .

This SD represents the coding of the field of Heaviside sign by subelement `.TOPOSE.HEA` . It allows the calculation of the functions characteristic of fields the voluminal points of Gauss. Indeed, each point of Gauss XFEM is localised in a subelement. The subelement is itself a under-partition of a field. To assign a code by subelement then is enough to characterize all the partitions of fields on the current element.

As many whole sites then are allocated than there is subelement in the current element `TOPONO.HSE(NSE)` .

```
V = .TOPONO.HSE (I) ;
```

V (1)	Whole code representing the partition of field to which the 1 belongs ^{er} subelement
V (2)	Whole code representing the partition of field to which belongs the 2 ^{ème} subelement
V (3)	Whole code representing the partition of field to which belongs the 3 ^{ème} subelement
...	...
V (NR)	Whole code representing the partition of field to which the NR belongs ^{ème} subelement

5.5.3 TOPONO.HFA

This elementary field contains 2 “whole” codes per under-facet X-FEM. The first code represents the field slave. The second code represents the main field.

These two codes are calculated with the same formula of coding as preceding SD “.TOPONO”. On the other hand, coding here is pressed on the field of Heaviside sign `TOPOFAC.HE` . As this last field is present only in multi-Heaviside, the field `TOPONO.HFA` is created only in multi-Heaviside. To extend this field into mono-Heaviside does not present a major difficulty and would improve the legibility of the code. For the moment, in this case, the code `TOPONO.HFA` is calculated starting from the sign of the Heaviside function on both sides of the facet is ± 1 .

The dimensioning of the field `TOPONO.HFA` lean then on the number of under-facets resulting from under-cutting (cf. `TOPOFAC.LO`) and on the number of cracks seen by the element, since a under-facet is a partitioning of a crack within the current element.

One has then:

```
V = .TOPONO.HFA (I);
```

V (2*nface* (ifiss-1)	Code of the field on the side slave of the crack ifiss under-facet n° ifa
-----------------------	---

```
+2*ifa-1)  
V (2*nface* (ifiss-1)  
+2*ifa)
```

Code of the field on the main side of the crack ifiss
under-facet n° ifa

5.6 Other objects

5.6.1 .XFEM_CONT

Vector of entières length equalizes to 1. This value is worth:

- 0 if there is no contact;
- 1 if a relation of contact or a cohesive relation were defined between the lips of the crack, and that linear finite elements are used with a standard formulation for the contact (`CONTACT='STANDARD'` in `MODI_MODELE_XFEM`);
- 2 if a relation of contact or a cohesive relation were defined between the lips of the crack, and that linear elements are used with a formulation "mortar" for the contact (`CONTACT='MORTAR'` in `MODI_MODELE_XFEM`);
- 3 if a relation of contact or a cohesive relation were defined between the lips of the crack, and that quadratic elements are used with a standard formulation for the contact.

5.6.2 .NFIS

Vector of entières length equalizes to 1. It corresponds to the number of cracks X-FEM of the model.

5.6.3 .FISS

Vector of K_8 of length equalizes with the number of cracks X-FEM of the model. It corresponds in the name of the `sd_fiss_xfem` of each crack.

5.6.4 .XMAFIS

`CHAM_ELEM` of K_8 . For each mesh, it `CHAM_ELEM` corresponds in the name of the cracks seen by this mesh, in the order defined by the user. The number of under-points of this SD corresponds to the number of cracks seen by the mesh.

5.6.5 .PRE_COND

Character string length 8 [K8]. For each model XFEM, it informs about the activation of algebraic prepacking XFEM. This prepacking is essential with the quadratic elements.

6 Description of the objects related to the resolution of the problem of contact-rubbing

6.1 Objects related to the SD created at the time of the definition of the load of contact-rubbing (defico)

The complete and detailed description of the objects created at the time of the definition of the load of contact, by L'operator `DEFI_CONTACT`, is the object of the booklet [D4.06.14]. Afin D'to help the reader of this document to apprehend it booklet [D4.06.14], a list of the objects specific to X-FEM described in [D4.06.14, §8] is given hereafter:

- `MODELX`, this object stores a reference towards the model enriched in `sd_contac T`;
- `CARAXF`, this object is used to define the various parameters for each zone of contact, *i.e.* each crack Master;
- `XFIMAI`, this object stores the main name of the crack associated with each zone of contact;
- `XNRELL`, this object stores the object `.LISEQ` (cf § 4.5.2) associated with each crack Master;
- `MAESCX`, this object stores information S relating to the meshes slaves with an aim of carrying out pairing, in the case of the approach great slips.

6.2 Objects related to SD created at the time of the resolution of the problem of contact-rubbing, within the framework approach great slips (resoco)

They is the large cards of contact slip X-FEM which are built once the pairing made between the points of integrations of the elements slaves and Masters. For each pairing, a late element is created in the operator `STAT_NON_LINE`. One stores in these various fields the relative information with these useful late elements during the integration of the contributions of contact-friction.

6.2.1 'XFPO'

Map of `R` who stores for each late element of information on pairing (see the comments in the routine `XMCART` for more details).

6.2.2 'XFST'

Map of `I` who Stocke in the order of the nodes of the late element statutes of enrichment of the nodes of the elements slave and Master of the late element. It corresponds to a copy of `MODEL SD \.STNO'` for the elements slave and Master of the late element.

6.2.3 'XFPI'

Map of `R` who stores the local coordinates of the points of intersection of the element slave. It corresponds to a copy of `MODEL SD \.TOPOFAC.PI'` for the element slave of the late element.

6.2.4 'XFAI'

Map of `R` who stores information on the intersected edges of the element slave. It corresponds to a copy of `MODEL SD \.TOPOFAC.AI'` for the element slave of the late element.

6.2.5 'XFCF'

Map of `I` who stores information on the connectivity of the facets of the element slave. It corresponds to a copy of `MODEL SD \.TOPOFAC.CF'` for the element slave of the late element.

6.2.6 'XFHF'

Map of \mathbb{I} who stores the value of the Heaviside functions on both sides of discontinuity, if the element slave or Master is multi-Heaviside. It corresponds to a copy of `MODEL SD \.TOPOFAC.HE'` side slave of the element slave and main side of the main element for the late element.

6.2.7 'XFPL'

Map of \mathbb{I} who stores the place of Lagrange of contact to the nodes of the element slave, if the element slave or Master is multi-Heaviside. It corresponds to a copy of `MODEL SD \.HEAVNO'` for the crack associated with the zone with contact, for the element slave of the late element.