

Operator PROPA_FISS

1 Goal

The goal of the operator `PROPA_FISS` is to make propagate a crack, this crack being defined with method X-FEM. Three types of methods are available:

- a method based on an algorithm of *fast marching* (`METHODE_PROPA='SIMPLEXE', 'UPWIND'`): this method is available in 2D and 3D for all types of finite elements. The initial crack can be defined on the grid of the model or an auxiliary grid.
- a purely geometrical method by calculation of distances to the face of crack (`METHODE_PROPA='GEOMETRIQUE'`)
- a method using a surface intermediate grid of the lip of the crack (`METHODE_PROPA='MAILLAGE'`): this method is available in 2D and 3D for all types of finite elements. The initial crack must also be defined using a grid.

The operator `PROPA_FISS` thus calculate the crack resulting from a propagation in fatigue according to a local law of Paris, starting from the preceding crack, and of the table of the factors of intensity of the constraints resulting from `CALC_G` or of `POST_K1_K2_K3`.

For the cracks in half-line (in 2D) or plane at rectilinear bottom (in 3D), the surface grid of the initial crack can be created by the operator `PROPA_FISS` (`METHODE_PROPA = 'INITIALIZATION'`).

The produced operator:

- that is to say a concept of the type `fiss_xfem` defining the propagated crack (case `METHODE_PROPA='SIMPLEXE', 'UPWIND' or 'GEOMETRIQUE'`);
- that is to say it grid of the initial crack (case `METHODE_PROPA='INITIALISATION'`);
- that is to say Mayllage of the propagated crack (case `METHODE_PROPA='MAILLAGE'`).

The theoretical aspect of the propagation of the level-sets for method X-FEM is approached in [R7.02.12].

2 Syntax

```
PROPA_FISS (

    ♦ METHODE_PROPA = / 'SIMPLEX'
                    / 'UPWIND'
                    / 'GRID'
                    / 'INITIALIZATION'
                    / 'GEOMETRICAL'

# If METHODE_PROPA = 'GEOMETRICAL'
    ♦ OPERATION = / 'NOTHING' [DEFECT]
                / 'DETEC_COHESIF'
                / 'PROPA_COHESIF'

# If METHODE_PROPA = 'SIMPLEX' or 'UPWIND' or 'GEOMETRICAL'
    ♦ MODEL = Mo, [model]
    ♦ TEST_MAIL = / 'NOT' [DEFECT]
                  / 'YES'

# If TEST_MAIL=' OUI'
    ♦ ITERATIONS = / iters, [I]
                  / 5, [DEFECT]
    ♦ TOLERANCE = / toler, [R]
                  / 5.0, [DEFECT]

# End if

    ♦ ZONE_MAJ = / 'TORUS' [DEFECT]
                 / 'ALL'

# If ZONE_MAJ=' TORE'
    ♦ RAYON_TORE = raytor, [R]

    ♦ CRACK = _F (
        ♦ FISS_ACTUELLE = fiss, [fiss_xfem]
        ♦ FISS_PROPAGEE = CO ('FISS'), [TXM]

        # If TEST_MAIL=' NON',
            ♦ TABLE = sif, [table]
            ♦ NB_POINT_FOND = nbptfd, [I]
        # End if
    ),

# If OPERATION=' RIEN' or OPERATION=' PROPA_COHESIF'
    ♦ DA_MAX = da, [R]

# If OPERATION=' DETEC_COHESIF'
    ♦ RESULTAT = LMBO, [evol_noli]

# If OPERATION=' RIEN'
    ♦ RAY = ray, [R]

# If TEST_MAIL=' NON' and OPERATION=' RIEN',
    ♦ COMP_LINE = F ( ♦ COEF_MULT_MINI = cmin, [R]
                    ♦ COEF_MULT_MAXI = cmax, [R]
                    ),

# If METHODE_PROPA = 'GRID'
    ♦ MAIL_STRUC = mast, [grid]
```

```

♦ ITERATION      = iter,                [ I ]
♦ DA_MAX         = da,                  [R]

♦ CRACK          = _F (
  ♦ MAIL_ACTUEL  = max1                [ grid ]
  ♦ FISS_ACTUELLE= FISS1                [ fiss_xfem]
  ♦ MAIL_PROPAGE = CO ('max2'),        [TXM]
  ♦ GROUP_MA_FOND= / 'BOTTOM'         [DEFECT]
  / grmaff                               [TXM]
  ♦ GROUP_MA_FISS= / 'FISS'           [DEFECT]
  / grmafi,                               [TXM]
  ♦ TABLE       = sif,                [table]

♦ MAIL_TOTAL     = CO ('matot2'),      [TXM]

♦ COMP_LINE      = _F ( ♦ COEF_MULT_MINI = cmin,    [R]
                       ♦ COEF_MULT_MAXI = cmax,    [R]
                       ),
                       ),

# If METHODE_PROPA = ' INITIALISATION',
♦ MAIL_STRUC     = mast,                [grid]
♦ FORM_FISS      = / 'DEMI_DROITE'
  / 'DEMI_PLAN'
  / 'ELLIPSE'
♦ GROUP_MA_FOND= / 'BOTTOM'           [DEFECT]
  / grmaff                               [TXM]
♦ GROUP_MA_FISS= / 'FISS'             [DEFECT]
  / grmafi,                               [TXM]
♦ MAIL_FISS      = CO ('max2'),        [TXM]
♦ MAIL_TOTAL     = CO ('matot2'),      [TXM]

# Cas DEMI_DROITE (in 2D)
♦ PFON          = (xf, yf, zf),        [1_R]
♦ DTAN          = (xt, yt, zt),        [1_R]

# Cas DEMI_PLAN (in 3D)
♦ POINT_ORIG    = (xo, yo, zo),        [1_R]
♦ POINT_EXTR    = (Xe, ye, ze),        [1_R]
♦ DTAN          = (xt, yt, zt),        [1_R]
♦ NB_POINT_FOND= nbfond,              [I]

# Case ELLIPSE (in 3D)
♦ CENTER        = (xo, yo, zo),        [1_R]
♦ DEMI_GRAND_AXE = has ,                [R]
♦ DEMI_PETIT_AXE = B,                  [R]
♦ VECT_X        = (vxx, vxy, vxz),     [1_R]
♦ VECT_Y        = (vyx, vyy, vyz),     [1_R]
♦ ANGLE_ORIG    = ango,                [R]
♦ ANGLE_EXTR    = angel,               [R]
♦ NB_POINT_FOND = nbpf,               [I]

# End if

# If (METHODE_PROPA = ' SIMPLEX' or 'UPWIND' and TEST_MAIL=' NOT' )
# or ( METHODE_PROPA = ' MAILLAGE' )
♦ LOI_PROPA     = _F (
  ♦ LAW          = /'PARIS',
  # If LOI=' PARIS'
  ♦ M            = m,                  [R]
  ♦ C            = C,                  [R]

```

```
# End if

♦ MATER = my , [to subdue]
),
◇ CRIT_ANGL_BIFURCATION = / 'SITT_MAX' [DEFECT]
/ 'SITT_MAX_DEVER'
/ 'K1_MAX'
/ 'K2_NUL'
/ 'PLAN'
/ 'ANGLE_IMPO'
/ 'ANGLE_IMPO_GAMMA'
/ 'ANGLE_IMPO_BETA_GAMMA'

# Impression of information
◇ INFORMATION = / 0
/ 1 [DEFECT]
/ 2
)
```

3 Operands common runs with all the options

3.1 Operand METHODE_PROPA

◆ METHODE_PROPA = / 'SIMPLEX',
/ 'UPWIND',
/ 'GRID',
/ 'INITIALIZATION',
/ 'GEOMETRICAL',

Three méthodes are available to make propagate a crack defined with method X-FEM:

•Methods based on an algorithm of fast marching

(METHODE_PROPA=' UPWIND'): a discretization of the Eikonale equation is solved to reset l_{sn} and l_{st} . The use of the auxiliary grid is essential for the irregular grids. For more details concerning the algorithms relative to this method, one will be able to refer to the reference material [R7.02.12].

(METHODE_PROPA=' SIMPLEXE'): Robust method, which can be applied to unspecified grids. For more details concerning the algorithms relative to this method, one will be able to refer to the reference material [R7.02.12].

•**a method using a surface intermediate grid of the lip of the crack** (METHODE_PROPA=' MAILLAGE '): this method is available in 2D and 3D for all types of finite elements.

The initial crack must also be defined using a grid, with specific rules of naming. For example in 3D, the bottom is called FOND_0 and contains the nodes NXA1 , NXB1 ,...; the lip is called FISS_0 and contains the surface meshes MXA1 , MXB1 ,...; the surface mesh MXA1 contains the nodes NXA0 , NXB0 , NXA1 and NXB1 .

•**a method using a purely geometrical approach** (METHODE_PROPA=' GEOMETRIQUE '): this method using a vectorial propagation of the bottom of crack and its local base as well as an estimate of the distances compared to this last in this local base is available in 2D and 3D for all types of finite elements without any restriction.

To facilitate the definition of this initial crack in the most usual cases (crack in half-line (in 2D) or planes at rectilinear bottom (in 3D)), the method should be used 'INITIALIZATION'.

Notice on the various methods:

The method of projection on an intermediate grid is as for it more approximate, but it makes it possible to have very satisfactory results in particular for the propagations in mode I pure.

The methods of fast marching are robust and give very satisfactory results for the propagations in mixed mode. The method simplex does not require an auxiliary grid.

The purely geometrical method is one method which do not require an auxiliary grid and offers good performances.

3.2 Operand OPERATION

```
# If METHODE_PROPA = 'GEOMETRICAL'  
  ◊ OPERATION = / 'NOTHING' [DEFECT]  
                / 'DETEC_COHESIF'  
                / 'PROPA_COHESIF'
```

This keyword defines the nature of the operation desired during the actualization of the crack:

If 'NOTHING' , one wishes a classical propagation of the level-sets normals and tangents according to some angle and a certain projection. It is the operation which it is necessary to select in the absence of cohesive forces.

The operation ' DETEC_COHESIF 'a detection of face by postprocessing of a result with cohesive forces carries out. Being given a possible surface of cracking, on which forces of cohesion are defined, it makes it possible to detect the new position of the face of propagation, by postprocessing of the result, which must be given under the keyword RESULT . It is thus about a tangent actualization of level-set only.

The operation ' PROPA_COHESIF 'extends the possible surface of cracking according to an angle given in argument in TABLE , and a uniform length specified by DA_MAX . It acts, once the direction of determined propagation, to extend the possible zone of cracking in order to carry out next mechanical calculation. This extension must be done over a length higher than the effective projection of the crack. This can be quantified by a simple dimensional analysis, and must be checked *a posteriori* during calculation.

The algorithms of propagation of crack with cohesive elements are described in the reference material [R7.02.19].

3.3 Keyword LOI_PROPA

```
# If ( METHODE_PROPA = 'SIMPLEXE' or 'GEOMETRICAL' 'UPWIND' or and  
RAFF_MAIL='NON')  
# or ( METHODE_PROPA = 'MAILLAGE' or 'INITIALIZATION')  
◊ LOI_PROPA = _F (  
  ◊ LAW = /'PARIS',  
    # If LOI='PARIS'  
      ◊ M = m, [R]  
      ◊ C = C, [R]  
    # End if  
  ◊ MATER = my, [ to subdue ]),
```

The keyword factor LOI_PROPA the law of propagation used defines.

For the moment, the only law available is a law of Paris, in the following form:

$$\frac{da}{dN} = C \cdot \Delta K^m$$

Coefficients C and m of this law must obligatorily be well informed.

The threshold of propagation of the crack is not considered.

Note:

*For the moment all the cracks of the model must propagate in same material. In fact only one material can be given and it is used for all the cracks of the model.
In more the law of Paris does not allow to consider the threshold of propagation of the crack.*

Caution:

The constants of the law of propagation must be given of way such as the unit rate of advance da/dN is equal to $L/cycles$, where L is the unit used in the model for the length.

3.4 Operand RESULT

```
# If OPERATION=' DETEC_COHESIF'  
  ♦ RESULTAT = LMBO,  
  [evol_noli]
```

It is a question of providing the computation result of STAT_NON_LINE whose field of internal variables of the cohesive law will make it possible to determine the position of the new face of propagation.

3.5 Operand DA_MAX

```
♦ DA_MAX = da, [R]
```

If there is only one fissures in the model M_0 , this operand defines the maximum advance of the crack.

If there are several cracks in the model M_0 , this operand defines the maximum advance of the crack which propagates more quickly. The number of cycles of tiredness is calculated by using the speed of the point of the bottom of the crack which propagates most quickly. This number of cycles is used for all the cracks of the model.

If OPERATION=' PROPA_COHESIF', this keyword defines the uniform length according to which one extends the new surface of possible cracking for next step of propagation.

3.6 Keyword factor COMP_LINE

```
# If TEST_MAIL=' NON',  
♦ COMP_LINE = _F ( ♦ COEF_MULT_MINI = cmin, [R]  
                  ♦ COEF_MULT_MAXI = cmax, [R]  
),
```

parameters COEF_MULT_MINI and COEF_MULT_MAXI operand COMP_LINE the conditions of minimal and maximum loading of the cycle of tiredness define. The two values are the constants by which one must multiply the loadings of the configuration of reference for to obtain the conditions of minimal and maximum loading of the cycle of tiredness. Thus if configuration of selected reference coincides with the condition of minimal or maximum loading, the value of COEF_MULT_MINI or COEF_MULT_MAXI, respectively, is equal to 1.

3.7 Keyword factor CRIT_ANGL_BIFURCATION

```
♦ CRIT_ANGL_BIFURCATION = / 'SITT_MAX '  
                        / 'SITT_MAX_DEVER' [DEFECT]
```

```
/ 'K1_MAX'  
/ 'K2_NUL'  
/ 'PLAN'  
/ 'ANGLE_IMPO'  
/ 'ANGLE_IMPO_GAMMA'  
/ 'ANGLE_IMPO_BETA_GAMMA'
```

This operand makes it possible to choose the criterion of calculation of the angle of junction:

- 'SITT_MAX' : criterion of the maximum circumferential constraint (Maximum Hoop Stress criterion, [R7.02.05 §2.5.2]) it is the by default criterion. It is calculated starting from the values of K1 and K2. It is available in 2D and 3D.
- 'SITT_MAX_DEVER' : criterion of the circumferential constraint maximum with taking into account of the KIII [R7.02.05 §2.5.2] and taken into account of the rotation of discharge [R7.02.13 §2.1]. It is calculated starting from the values of K1, K2 and K3 and available only in 3D.
- 'K1_MAX' , 'K2_NUL' : criteria of Amestoy, Bui and Dang-Van [R7.02.05 §2.5.1]. These 2 criteria are only available in 2D. The angle is given with a margin of 10 degrees. Attention, one notes that the criterion 'K2_NUL' do not function for an angle higher than 60°.
- 'PLAN' : defines a null angle corresponding to a plane propagation.
- 'ANGL_IMPO' : allows to choose an angle calculated beforehand. In this case, one takes as angle that given in the column BETA table as starter (table given under the keyword TABLE) and the rotation of discharge is not activated. It is necessary thus for that the table as starter of PROPA_FISS a column contains BETA .
- 'ANGLE_IMPO_GAMMA' : allows to choose a dumping angle calculated beforehand. In this case, one takes as dumping angle that given in the column GAMMA table as starter (table given under the keyword TABLE) and the angle of connection is calculated [R7.02.13§2.1]. It is necessary thus for that the table as starter of PROPA_FISS a column contains GAMMA . The dumping angle BETA is calculated by the operator POST RUPTURE, in the configuration CRITERION = SITT_MAX_DEVER [U4.82.04].
- 'ANGLE_IMPO_BETA_GAMMA' : allows to choose angles of beforehand calculated connection and discharge. In this case one takes as angle of connection and discharge respectively BETA and GAMMA given in the table as starter (under the keyword TABLE). It is necessary for that the table as starter contains a column BETA and a column GAMMA of the same size.

Notice :

The possibility of forcing the angle of connection BETA then to deduce the dumping angle from it GAMMA is not in conformity with the model. Indeed the calculation of GAMMA imposes the knowledge of the factors of intensity of the constraints which one cannot find starting from the only data of the angle of connection BETA.

3.8 Operand INFORMATION

- ```
/ 0 : impression on the file 'MESSAGE'
```
- stages of calculation
- ```
/ 1 : impression on the file 'MESSAGE'
```
- same information as in INFO=1
 - method 'GRID'and'INITIALIZATION': impression of the surface grid of the crack to the format Aster

- / 2 : Impression on the file 'MESSAGE'
- same information as in INFO=1
 - method 'UPWIND','SIMPLEX': details of the calculation of the fields and enrichment of the SD fissures
 - method 'UPWIND': plugboard of the nodes of the grid of calculation

4 Operands specific to the methods SIMPLEX, UPWIND and GEOMETRICAL

4.1 Operand MODEL

◇ MODEL = Mo, [model]

Name of the model on which the crack to be propagated is defined.

4.2 Operand TEST_MAIL

◇ TEST_MAIL = / 'NOT' [DEFECT]
/ 'YES'

Grid used for the representation of the level sets (grid of the model M_0) must be sufficiently refined for representing the crack well. One can check it by using the operand TEST_MAIL.

In the case TEST_MAIL=' OUI ', the same propagation velocity is assigned to all the points of the funds of cracks defined on the model M_0 . All the cracks thus advance length given by the operand DA_MAX. If the grid is sufficiently refined, the bottom of crack after the propagation must be homothetic at the bottom before the propagation. If that is not checked, the grid must be refined.

The operand calculates several propagations of the cracks of the model M_0 (see operands DA_MAX, ITERATIONS and TOLERANCE) and for each propagation it calculates the distance between the new bottom after the propagation and the bottom at the beginning of the propagation (see FISS_ACTUELLE in the operand CRACK). If grid of the model M_0 or the auxiliary grid `griaux` used for the representation of the level sets are sufficiently refined, the calculated distance must be equal in advance imposed DA_MAX multiplied by the number of the current iteration. A small tolerance is used for this checking (see operand TOLERANCE). If that is not checked, a message of alarm is exit by PROPA_FISS.

The resolution of the fissured model and the calculation of the factors of intensity of the constraints are not necessary when this operand is used. The checking is thus very fast (see case test zzzz255a and zzzz255b).

4.3 Operand ITERATIONS

```
# If TEST_MAIL=' OUI '  
◇ ITERATIONS = / iters, [I]  
/ 5, [DEFECT]
```

This operand is used to give the iteration count (propagations) calculated during the test of the grid (see operand TEST_MAIL). For each iteration, the crack is propagated and outdistances it between the bottom of crack propagated and the bottom at the beginning of the propagation (see FISS_ACTUELLE in the operand CRACK) is calculated and checked. It is strongly advised to use an iteration count equal or higher than 5 because the digital errors accumulate with each iteration.

4.4 Operand TOLERANCE

```
# If TEST_MAIL=' OUI '  
◇ TOLERANCE = / toler, [R]  
/ 5.0, [DEFECT]
```

This operand is used to give the tolerance, expressed like percentage, used in the test of the grid (see operand `TEST_MAIL`). For each iteration of the test (see operand `ITERATIONS`), the maximum and minimal distance enters the propagated bottom and the bottom at the beginning of the propagation (see `FISS_ACTUELLE` in the operand `CRACK`) are calculated. Theoretically these distances are identical and their value equal in advance is imposed `DA_MAX` multiplied by the number of the current iteration. One thus calculates the difference between the distances maximum and minimal and the theoretical distance and one checks that the two calculated differences are lower than a limiting value. This limiting value is calculated like percentage length of the smallest edge of the elements of the grid used for the representation of the level sets. The percentage used is given by the operand `TOLERANCE`.

4.5 Operand `ZONE_MAJ`

```
◇ ZONE_MAJ = / 'TORUS' [DEFECT]  
/ 'ALL'
```

This operand makes it possible to select the zone used for the update of the level sets. One can choose to calculate on all the nodes of the model `M0` while using `ZONE_MAJ=' TOUT '` or to limit the zone of calculation to the nodes of the model `M0` who are in a torus built around the bottom of the crack while using `ZONE_MAJ=' TORE '`.

This last choice makes it possible to reduce strongly the computing time even if the grid used for the representation of the level sets is not formed by a large number of meshes. In the case of propagation in mixed mode, the use of the restriction of the zone of update is strongly advised to improve the performance of calculation and to avoid the distortion of the level sets, which can cause a fatal error during the update.

Of the ray of the torus of localization and its update with each propagation are calculated automatic way. However, if the size of the elements in the zone of propagation is not uniform, the calculated ray can be too small and an error message is resulting by the operator `PROPA_FISS`. In this case the user is obliged to give a value of the ray explicitly (see operand `RAYON_TORE`).

4.6 Operand `RAYON_TORE`

```
# If ZONE_MAJ=' TORE '  
◇ RAYON_TORE = raytor, [R]
```

This operand makes it possible to give the value of the ray of the torus used for the restriction of the zone of update of the level sets (see operand `ZONE_MAJ`). The ray of the torus is calculated automatically by the operator `PROPA_FISS` and the user is not obliged to use this operand. However the computed value can be too small in certain case, for example when the sizes minimal and maximum of the elements in the zone of propagation are very different between them. In all these cases, the user must forcing give the ray of the torus while following advise them contained in the error messages resulting by `PROPA_FISS`.

It should be noticed that the value of the ray cannot change much between two successive propagations (see R7.02.12 documentation for more details). Thus it is strongly advised to use the same value with each call of `PROPA_FISS` by giving the maximum value envisaged. The value of the ray must be always higher than a minimal value which can be estimated by calculating the addition of the values of the projection of the crack (see operand `DA_MAX`) and of the ray of convergence (see operand `RAY`). However this value is also dependent in keeping with elements in the torus and thus it is strongly advised not to use a value lower than that which

was calculated automatically by PROPA_FISS . An error is resulting by PROPA_FISS if the ray given is not good compared to the propagation to be simulated and in keeping with elements of the grid.

Caution: in the cases of cracks with several funds of crack (for example in the case of a crack of type segment, in 2d, which has 2 funds of crack); it only T is not necessaryore includes the other funds of the crack (even if these other funds are “apart from the grid”).

4.7 Keyword factor CRACK

This obligatory operand makes it possible to indicate the cracks of the model M_0 who propagate.

4.7.1 Syntax CRACK

```
♦ CRACK = _F (
  ♦ FISS_ACTUELLE = fiss, [fiss_xfem]
  ♦ FISS_PROPAGEE = CO ('FISS'), [TXM]

  # If RAFF_MAIL = 'NOT',
    ♦ TABLE = sif, [table]
    ♦ NB_POINT_FOND = nbptfd, [I]
  # End if
),
```

4.7.2 Operand FISS_ACTUELLE

```
♦ FISS_ACTUELLE = fiss, [fiss_xfem]
```

One gives the name of the current crack to propagate. This crack must be already affected with the model M_0 by using the operator MODI_MODELE_XFEM.

4.7.3 Operand FISS_PROPAGEE

```
♦ FISS_PROPAGEE = CO ('FISS'), [TXM]
```

This operand makes it possible to define the name of the propagated crack produces by the operator PROPA_FISS. The produced result is a concept of the type `fiss_xfem` defining the propagated crack.

4.7.4 Operand TABLE

```
# If TEST_MAIL='NON'
♦ TABLE = sif, [table]
```

This operand must be indicated with the table of the factors of intensity of the constraints in bottom of the crack given by the operand FISS_ACTUELLE. This table comes from the operators CALC_G or POST_K1_K2_K3.

In the basic case of multiple crack (see R7.02.12, paragraph “Fund of crack the multiple”), for which the bottom of the crack is formed by several pieces, operator CALC_G/POST_K1_K2_K3 must be called several times, once for each piece (operand NUME_FOND of CALC_G/POST_K1_K2_K3). One thus has several tables, a table for each piece, which must be combined in only one table which will be given to PROPA_FISS in the operand TABLE. For that, one can use the operator CALC_TABLE. By example, if the crack is formed by two pieces, the combination of the two tables can be realized in the following way:

```
SIF1 = CALC_G (THETA=_F (...
                                     NUME_FOND= 1 ,),
               ...)
SIF2 = CALC_G (THETA=_F (...
                                     NUME_FOND= 2 ,),
               ...)

SIF = CALC_TABLE (TABLE=SIF1,
                  ACTION= (_F (OPERATION=' COMB',
                                TABLE=SIF2,
                                NOM_PARA= ('NUMÉRIQUE_ORDRE',
                                           'NUMÉRIQUE_FOND')),),
                  );
```

Then one uses `SIF` in the operator `PROPA_FISS`. This table is formed by the union of the two tables resulting by `CALC_G`. An unspecified order can be used in the combination of the tables.

4.7.5 Operand `NB_POINT_FOND`

```
# If TEST_MAIL=' NON'
◇ NB_POINT_FOND = nbptfd, [I]
```

If the operand `NB_POINT_FOND` was used in `CALC_G` for the calculation of the factors of intensity of the constraints, one must inevitably use this operand in `PROPA_FISS` too.

The operator `PROPA_FISS` cannot check if this operand were used or not in `CALC_G` or if the values given are correct. The user must thus pay attention to use this operand correctly!

The operand `NB_POINT_FOND` accept a list of values of entireties. For each piece (see operand `TABLE`) crack given by `FISS_ACTUELLE` one must give a value of `NB_POINT_FOND` equalize with the value used in `CALC_G`. If the operand were not used in `CALC_G` for a piece, one uses zero value. The order of the values of `NB_POINT_FOND` in the list is given by the operand `NUME_FOND` of `CALC_G`. Examples are given at the end of this document.

4.8 Operand `RAY`

```
◇ RAY= ray, [R]
```

This operand is used to define the ray of the torus around the bottom of crack in which the residues will be estimated for the stages of rebootstraping and reorthogonalisation of the level sets. This torus defines the zone where the level sets must be sufficiently close to functions of signed distances.

The value of this ray must always be larger than the higher ray of integration of the operator `CALC_G`. The operator `PROPA_FISS` cannot check if this condition is observed. The user must thus pay attention to use this operand correctly!

5 Operands specific to the method GRID

5.1 Operands MAIL_STRUC

- ◆ MAIL_STRUC = mast, [grid]

This operand makes it possible to define the grid of the healthy structure (*i.e.* without crack).

5.2 Operand ITERATION

- ◆ ITERATION = iter, [I]

This operand makes it possible to define the number of the iteration of the calculation of propagation. The first iteration (propagation of the initial crack) is iteration 1.

This operand facilitates the realization of calculations: it is thus possible to start again a calculation of propagation on a crack having already been propagated *nb_init* time.

5.3 Operand MAIL_TOTAL

- ◆ MAIL_TOTAL = CO ('matot2'), [TXM]

This operand makes it possible to define the name of the total grid, *i.e.* concatenation of the grid of the healthy structure (cf operand MAIL_STRUC) and of the grid of the propagated crack MAIL_PROPAGE.

It is this total grid *matot2* who must be used to define the model and the new crack in DEFI_FISS_XFEM.

5.4 Keyword factor CRACK

This obligatory operand makes it possible to indicate the cracks of the model *Mo* who propagate.

5.4.1 Syntax CRACK

- ◆ CRACK = _F (
 - ◆ MAIL_ACTUEL = max1 [grid]
 - ◆ FISS_ACTUELLE = FISS1 [fiss_xfem]
 - ◇ MAIL_PROPAGE = CO ('max2'), [TXM]
 - ◇ GROUP_MA_FOND = / 'BOTTOM' [DEFECT]
 - ◇ GROUP_MA_FISS = / 'FISS' [DEFECT]
 - ◆ TABLE = sif, [table]

5.4.2 Operand MAIL_ACTUEL

- ◆ MAIL_ACTUEL = max1, [grid]

This operand makes it possible to define the crack which one wants to propagate using his surface grid *max1*. Grid of the crack *max1* must comply with certain rules of naming, cf §5.

5.4.3 Operands FISS_ACTUELLE, TABLE

See the paragraphs §4.7.2 and §4.7.4.

5.4.4 Operand MAIL_PROPAGE

```
◇ GROUP_MA_FOND= / 'BOTTOM' [DEFECT]
                  / grmaff [TXM]
◇ GROUP_MA_FISS= / 'FISS' [DEFECT]
                  / grmafi, [TXM]
```

These operands make it possible to indicate how will be called the groups of meshes of the bottom and the lips (which will be successively FOND_0, FOND_1, etc). These keywords are essential to be able to use an unspecified grid for the initial crack; and to be able to treat the case of multi-cracking. A value by default exists.

5.4.5 Operands GROUP_MA_FOND and GROUPE_MA_FISS

```
◇ MAIL_PROPAGE = CO ('max2'), [TXM]
```

This operand allows Dto éfinir the name of the surface grid corresponding to the propagated crack.

6 Operands specific to the method **INITIALIZATION**

6.1 Operands **MAIL_STRUC**

- ◆ `MAIL_STRUC = mast,` [grid]

This operand makes it possible to define the grid of the healthy structure (*i.e.* without crack).

6.2 Operands **FORM_FISS**

- ◆ `FORM_FISS = / 'DEMI_DROITE'`
`/ 'DEMI_PLAN'`
`/ 'ELLIPSE'`

This operand makes it possible to define the type of the crack: a half-line in 2D or a crack planes at rectilinear bottom in 3D, or an elliptic crack in 3D.

6.3 Case **DEMI_DROITE** : operands **PFON** and **DTAN_DTAN**

- ◆ `PFON = (xf, yf, zf),` [1_R]
- ◆ `DTAN = (xt, yt, zt),` [1_R]

These operands make it possible to define the initial crack in 2D by giving the coordinate of the point of the bottom of crack `PFON` and direction of propagation of this one `DTAN`.

6.4 Case **DEMI_PLAN** : operands **PFON**, **POINT_ORIG**, **POINT_EXTR** and **NB_POINT_FOND**

- ◆ `POINT_ORIG = (xo, yo, zo),` [1_R]
- ◆ `POINT_EXTR = (Xe, ye, ze),` [1_R]
- ◆ `DTAN = (xt, yt, zt),` [1_R]
- ◆ `NB_POINT_FOND = nbfond,` [I]

These operands make it possible to define the initial crack in 3D by giving the coordinates of the points origin and end of the bottom of crack and the direction of propagation of this one `DTAN`. The bottom of crack will be discretized with `nbfond` nodes.

6.5 Case **ELLIPSE** : operands **CENTER**, **DEMI_GRAND_AXE**, **DEMI_PETIT_AXE**, **VECT_X**, **VECT_Y**, **ANGLE_ORIG**, **ANGLE_EXTR** and **NB_POINT_FOND**

- ◆ `CENTER = (xo, yo, zo),` [1_R]
- ◆ `DEMI_GRAND_AXE = has,` [R]
- ◆ `DEMI_PETIT_AXE = B,` [R]
- ◆ `VECT_X = (vxx, vxy, vxz),` [1_R]
- ◆ `VECT_Y = (vyx, vyy, vyz),` [1_R]

These operands make it possible to define the initial elliptic crack in 3D by giving the coordinates of the center of the ellipse, the values of the half-small axis and equatorial radius, of the vectors of orientation of the plan of the ellipse. For more details concerning these keyword, to see U4.82.08.

- ◆ ANGLE_ORIG = ango, [R]
- ◆ ANGLE_EXTR = angel, [R]
- ◆ NB_POINT_FOND = nbpf, [I]

ANGLE_ORIG give the position (in degrees) first point in the reference mark (VECT_X , VECT_Y).

ANGLE_EXTR give the position (in degrees) last point in the reference mark (VECT_X , VECT_Y).

NB_POINT_FOND indicate the number of point to be created along the bottom of crack.

6.6 Operands GROUP_MA_FOND, GROUP_MA_FISS

See § 5.4.5

6.7 Operands MAIL_TOTAL

See § 5.3

6.8 Operands MAIL_FISS

Idem that MAIL_PROPAGE, to see § 5.4.4

7 Examples of propagation of crack with X-FEM

Several CAS-tests make it possible to validate the propagation of cracks with X-FEM.

In 2D:

- sslp318: propagation of a crack X-FEM not emerging requested in mode;
- sslp319: propagation of two cracks X-FEM emerging requested in mode;
- sslp318: Propagation of a crack X-FEM not emerging requested in mode;
- sslp320: Propagation of a crack X-FEM emerging requested in Mode I;
- sslp321: Propagation of a crack X-FEM in a flexbeam 3 points.

In 3D:

- sslv315: propagation in 3D of a tilted crack;
- sslv314: Propagation planes of a crack emerging with X-FEM;
- sslv316: Cracking with propagation imposed with X-FEM;
- sslv320: Propagation planes of a crack 3D dividing and amalgamating with X-FEM.