

Opérateur PROPA_FISS

1 But

Le but de l'opérateur `PROPA_FISS` est de faire propager une fissure, cette fissure étant définie avec la méthode X-FEM. Deux méthodes sont disponibles :

- une méthode par intégration numérique des équations de propagation du fond de fissure (`METHODE_PROPA='SIMPLEXE'` ou `'UPWIND'`): cette méthode est disponible en 2D et 3D pour tous types d'éléments finis. La fissure initiale peut être définie sur le maillage du modèle ou sur un maillage auxiliaire. Cette dernière solution permet une réduction du temps de calcul et une meilleure représentation de la fissure.
- une méthode utilisant un maillage intermédiaire surfacique de la lèvres de la fissure (`METHODE_PROPA='MAILLAGE'`): cette méthode est disponible en 2D et en 3D pour tous types d'éléments finis. La fissure initiale doit elle aussi être définie à l'aide d'un maillage.

L'opérateur `PROPA_FISS` calcule ainsi la fissure résultante d'une propagation en fatigue d'après une loi de Paris locale, à partir de la fissure précédente, et du tableau des facteurs d'intensité des contraintes issus de `CALC_G` (ou de `POST_K1_K2_K3`, pour la méthode `MAILLAGE` uniquement).

Pour les fissures en demi-droite (en 2D) ou planes à fond rectiligne (en 3D), le maillage surfacique de la fissure initiale peut être créé par l'opérateur `PROPA_FISS` (`METHODE_PROPA = 'INITIALISATION'`).

L'opérateur produit :

- soit un concept de type `fiss_xfem` définissant la fissure propagée (cas `METHODE_PROPA='SIMPLEXE'` ou `'UPWIND'`);
- soit le maillage de la fissure initiale (cas `METHODE_PROPA='INITIALISATION'`);
- soit le maillage de la fissure propagée (cas `METHODE_PROPA='MAILLAGE'`).

L'aspect théorique de la propagation des level-sets pour la méthode X-FEM est abordé dans [R7.02.12].

2 Syntaxe

```
PROPA_FISS (

    ♦ METHODE_PROPA    = / 'SIMPLEXE'
                        / 'UPWIND'
                        / 'MAILLAGE'
                        / 'INITIALISATION'

#   Si METHODE_PROPA = 'SIMPLEXE' ou 'UPWIND'
    ♦ MODELE           = mo, [modele]
    ♦ TEST_MAIL        = / 'NON' [DEFAULT]
                        / 'OUI'
    # Si TEST_MAIL='OUI'
        ♦ ITERATIONS   = / iters, [I]
                        / 5, [DEFAULT]
        ♦ TOLERANCE    = / toler, [R]
                        / 5.0, [DEFAULT]
    # Fin si

    ♦ ZONE_MAJ         = / 'TORE' [DEFAULT]
                        / 'TOUT'

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

    ♦ FISSURE          = _F(
        ♦ FISS_ACTUELLE = fiss, [fiss_xfem]
        ♦ FISS_PROPAGEE = CO('FISS'), [TXM]
        ♦ GRILLE_AUX    = griaux, [modele]
        # Si TEST_MAIL='NON',
            ♦ TABLE     = sif, [table]
            ♦ NB_POINT_FOND = nbptfd, [I]
        # Fin si
        ),
    ♦ DA_MAX           = da, [R]
    ♦ RAYON            = ray, [R]
    # Si TEST_MAIL='NON',
    ♦ COMP_LINE        = F( ♦ COEFF_MULT_MINI = cmin, [R]
                          ♦ COEFF_MULT_MAXI = cmax, [R]
                          ),

#   Si METHODE_PROPA = 'MAILLAGE'
    ♦ MA_STRUC         = mast, [maillage]
    ♦ MA_XFEM1         = max1, [maillage]
    ♦ FISSURE1         = FISS1, [ fiss_xfem ]
    ♦ DTAN_ORIG        = (xo,yo,zo), [l_R]
    ♦ DTAN_EXTR        = (xe,ye,ze), [l_R]
    ♦ METHODE_POSTK    = /1
                        /2
                        /3
    ♦ TABLE           = sif, [table]
    ♦ ITERATION        = i, [ I ]
    ♦ HYPOTHESE        = / ' PLAN '
                        / ' NON_PLAN ' [DEFAULT]
    ♦ MA_XFEM2         = CO('max2'), [TXM]
    ♦ MA_TOT2          = CO('matot2'), [TXM]

#   Si METHODE_PROPA = 'INITIALISATION',
    ♦ MA_STRUC         = mast, [maillage]
```

```

    ♦ FORM_FISS = / 'DEMI_DROITE'
                  / 'DEMI_PLAN'
# Cas DEMI_DROITE (en 2D)
    ♦ PFON = (xf,yf,zf), [1_R]
    ♦ DTAN = (xt,yt,zt), [1_R]
# Cas DEMI_PLAN (en 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]

    ◇ MA_XFEM2 = CO('max2'), [TXM]
    ♦ MA_TOT2 = CO('matot2'), [TXM]
# Fin si

# Si (METHODE_PROPA='SIMPLEXE' ou 'UPWIND' et TEST_MAIL='NON')
# ou ( METHODE_PROPA ='MAILLAGE' ou 'INITIALISATION')
♦ LOI_PROPA = _F (
    ♦ LOI = /'PARIS',
    # Si LOI='PARIS'
        ♦ M = m, [R]
        ♦ N = n, [R]
        # Si METHODE_PROPA='MAILLAGE' ou 'INITIALISATION'
        ◇ / C = c, [R]
        / DA_MAX = dmax, [R]
        # Si METHODE_PROPA='SIMPLEXE' ou 'UPWIND'
        ♦ C = c, [R]
    # Fin si
    # Si METHODE_PROPA ='SIMPLEXE' ou 'UPWIND'
    ♦ MATER = ma , [mater]
    ),

# Impression d'informations
◇ INFO = / 0
          / 1 [DEFAULT]
          / 2
)

```

3 Opérandes communs à toutes les options

3.1 Opérande METHODE_PROPA

```
♦ METHODE_PROPA = / 'SIMPLEXE',  
                  / 'UPWIND',  
                  / 'MAILLAGE',  
                  / 'INITIALISATION',
```

Deux méthodes sont disponibles pour faire propager une fissure définie avec la méthode X-FEM :

• **une méthode par intégration numérique des équations de propagation du fond de fissure** (METHODE_PROPA='SIMPLEXE' et 'UPWIND') : pour l'instant la méthode 'UPWIND' est la plus fiable mais elle peut être seulement appliquée à des maillages très réguliers (voir l'opérande GRILLE_AUX et la documentation de référence [R7.02.12]).

La méthode 'SIMPLEXE' n'est pas encore très robuste mais elle peut être appliquée à des maillages quelconques. Pour plus de détails concernant les algorithmes relatifs à ces méthodes, on pourra se référer à la documentation de référence [R7.02.12].

• **une méthode utilisant un maillage intermédiaire surfacique de la lèvres de la fissure** (METHODE_PROPA='MAILLAGE') : cette méthode est disponible en 2D et en 3D pour tous types d'éléments finis.

La fissure initiale doit elle aussi être définie à l'aide d'un maillage, avec des règles de nommage spécifiques. Par exemple en 3D, le fond s'appelle FOND_0 et contient les nœuds NXA1, NXB1, ... ; la lèvres s'appelle FISS_0 et contient les mailles surfaciques MXA1, MXB1, ... ; la maille surfacique MXA1 contient les nœuds NXA0, NXB0, NXA1 et NXB1.

Pour faciliter la définition de cette fissure initiale dans les cas les plus usuels (fissure en demi-droite (en 2D) ou plane à fond rectiligne (en 3D)), il faut utiliser la méthode 'INITIALISATION'.

Remarque sur les différentes méthodes :

La méthode par intégration numérique des équations de propagation est théoriquement la plus précise; la méthode de projection sur un maillage intermédiaire est quant à elle plus approchée, mais elle permet d'avoir des résultats très satisfaisants notamment pour les propagations en mode I pur.

Ces deux méthodes sont encore en cours de validation.

3.2 Mot clé LOI_PROPA

```
# Si ( METHODE_PROPA='SIMPLEXE' ou 'UPWIND' et RAFF_MAIL='NON')  
# ou ( METHODE_PROPA='MAILLAGE' ou 'INITIALISATION')  
♦ LOI_PROPA = _F (  
  ♦ LOI = /'PARIS',  
    # Si LOI='PARIS'  
    ♦ M = m, [R]  
    ♦ N = n, [R]  
    # Si METHODE_PROPA='MAILLAGE' ou 'INITIALISATION'  
    ♦ / C = c, [R]  
    / DA_MAX = dmax, [R]  
    # Si METHODE_PROPA='SIMPLEXE' ou 'UPWIND'  
    ♦ C = c, [R]  
  # Fin si  
  # Si METHODE_PROPA='SIMPLEXE' ou 'UPWIND'  
♦ MATER = ma, [mater ]),
```

Le mot clé facteur `LOI_PROPA` définit la loi de propagation utilisée.

Pour l'instant, la seule loi disponible est une loi de Paris, dans la forme suivante:

$$\frac{da}{dN} = \frac{C}{(1-R)^n} \cdot \Delta K^m$$

Les coefficients m et n de cette loi doivent obligatoirement être renseignés. Le terme $(1-R)^n$ permet de considérer l'effet du rapport de charge sur la vitesse de propagation de la fissure. Bien que fortement déconseillé, on peut négliger cet effet en donnant $n=0$.

La seuil de propagation de la fissure n'est pas considérée.

Si on utilise `METHODE_PROPA = 'MAILLAGE' OU 'INITIALISATION'`, on doit renseigner un seul paramètre entre `C` et `DA_MAX`. L'incrément de propagation du fond de fissure entre deux itérations est:

- soit calculé à partir des coefficients `C` et m de la loi de Paris pour la méthode 'MAILLAGE' si `C` est fourni;
- soit imposé directement par l'utilisateur pour la méthode 'MAILLAGE' si `DA_MAX` est fourni. En 3D, `DA_MAX` correspond à l'incrément du nœud du fond qui se propage le plus vite.

Si on utilise `METHODE_PROPA = 'SIMPLEXE' OU 'UPWIND'`, on doit forcément renseigner le coefficient `C` de la loi de Paris et le matériau dans lequel les fissures du modèle `mo` propagent (voir opérande `MODELE` relatif à les méthodes *simplexe* et *upwind*). L'incrément de propagation du fond de fissure entre deux itérations est toujours donné par l'utilisateur (voir opérande `DA_MAX` relatif à les méthodes *simplexe* et *upwind*).

LIMITATION

Pour l'instant toutes les fissures du modèle doivent propager dans le même matériau. En fait un seul matériau peut être donné et il est utilisé pour toutes les fissures du modèle.

En plus la loi de Paris ne permet pas de considérer le seuil de propagation de la fissure.

ATTENTION!

Les constantes de la loi de propagation doivent être données en façon telle que l'unité de la vitesse d'avancement da/dN est égale à $L/cycles$, où L est l'unité utilisée dans le modèle pour la longueur.

3.3 Opérande INFO

/ 0 : impression sur le fichier 'MESSAGE '

- des étapes de calcul

/ 1 : impression sur le fichier 'MESSAGE '

- des mêmes informations qu'en `INFO=1`
- méthode 'UPWIND' et 'SIMPLEXE' : paramètres de convergence des boucles et étapes d'enrichissement de la SD fissure
- méthode 'MAILLAGE' et 'INITIALISATION' : impression du maillage surfacique de la fissure au format Aster

- / 2 : Impression sur le fichier 'MESSAGE'
- des même informations qu'en INFO=1
 - méthode 'UPWIND' et 'SIMPLEXE': des détails du calcul des champs et d'enrichissement de la SD fissure
 - méthode 'UPWIND': tableau de connexion des nœuds de la grille du calcul

4 Opérands spécifiques aux méthodes SIMPLEXE et UPWIND

4.1 Opérande MODELE

◆ MODELE = mo, [modele]

Nom du modèle sur lequel la fissure à propager est définie.

4.2 Opérande TEST MAIL

```
◇ TEST_MAIL = / 'NON' [DEFAULT]
              / 'OUT'
```

Le maillage utilisé pour la représentation des level sets (maillage du modèle `mo` ou de la grille auxiliaire `griaux` (voir opérande `GRILLE_AUX`)) doit être suffisamment raffiné pour bien représenter la fissure. On peut le vérifier en utilisant l'opérande `TEST_MAIL`.

Dans le cas `TEST_MAIL='OUI'`, la même vitesse de propagation est affectée à tous les points des fonds de fissures définies sur le modèle `mo`. Toutes les fissures avancent donc de la longueur donnée par l'opérande `DA_MAX`. Si le maillage est suffisamment raffiné, le fond de fissure après la propagation doit être homothétique au fond avant la propagation. Si cela n'est pas vérifié, le maillage doit être raffiné.

L'opérande calcule plusieurs propagations des fissures du modèle `mo` (voir opérandes `DA_MAX`, `ITERATIONS` et `TOLERANCE`) et pour chaque propagation il calcule la distance entre le nouveau fond après la propagation et le fond au début de la propagation (voir `FISS_ACTUELLE` dans l'opérande `FISSURE`). Si le maillage du modèle `mo` ou la grille auxiliaire `griaux` utilisés pour la représentation des level sets sont suffisamment raffinés, la distance calculée doit être égale à l'avance imposée `DA_MAX` multipliée par le numéro de l'itération courante. Une petite tolérance est utilisée pour cette vérification (voir opérande `TOLERANCE`). Si cela n'est pas vérifié, un message d'alerte est issue par `PROPA_FISS`.

La résolution du modèle fissuré et le calcul des facteurs d'intensité des contraintes ne sont pas nécessaires quand on utilise cet opérateur. La vérification est donc très rapide (voir cas test zzzz255a et zzzz255b).

4.3 Opérande ITERATIONS

```
# Si TEST_MAIL='OUI'
◇ ITERATIONS      = / iters, [I]
                  / 5, [DEFAULT]
```

Cet opérande est utilisé pour donner le nombre de itérations (propagations) calculées pendant le test du maillage (voir opérande `TEST_MAIL`). Pour chaque itération, la fissure est propagée et la distance entre le fond de fissure propagé et le fond au début de la propagation (voir `FISS_ACTUELLE` dans l'opérande `FISSURE`) est calculée et vérifiée. Il est fortement conseillée d'utiliser un nombre d'itérations égal ou supérieur à 5 car les erreurs numériques s'accumulent à chaque itération.

4.4 Opérande TOLERANCE

```
# Si TEST_MAIL='OUI'
◇ TOLERANCE    =  / toler,          [R]
                  / 5.0,          [DEFAULT]
```

Cet opérateur est utilisé pour donner la tolérance, exprimée comme pourcentage, utilisée dans le test du maillage (voir opérateur TEST_MAIL). Pour chaque itération du test (voir opérateur ITERATIONS), la distance maximale et minimale entre le fond propagé et le fond au début de la propagation (voir FISS_ACTUELLE dans l'opérateur FISSURE) sont calculées. Théoriquement ces distances sont identiques et leur valeur est égale à l'avance imposé DA_MAX multipliée par le numéro de l'itération courante. On calcule donc la différence entre les distances maximale et minimale et la distance théorique et on vérifie que les deux différences calculées sont inférieures à une valeur limite. Cette valeur limite est calculée comme pourcentage de la longueur de la plus petite arête des éléments du maillage utilisée pour la représentation des level sets (voir opérateur GRILLE_AUX). La pourcentage utilisée est donnée par l'opérateur TOLERANCE.

4.5 Opérateur ZONE_MAJ

```
◇ ZONE_MAJ = / 'TORE' [DEFAULT]
/ 'TOUT'
```

Cet opérateur permet de sélectionner la zone utilisée pour la mise à jour des level sets. On peut choisir de faire le calcul sur tous les nœuds du modèle mo ou de la grille auxiliaire gri aux (voir opérateur GRILLE_AUX) en utilisant ZONE_MAJ='TOUT' ou de limiter la zone de calcul à les nœuds du modèle mo ou de la grille auxiliaire gri aux qui sont dans un tore construit autour du fond de la fissure en utilisant ZONE_MAJ='TORE'.

Ce dernier choix permet de réduire fortement le temps de calcul même si le maillage utilisé pour la représentation des level sets n'est pas formé par un grand nombre de mailles. Dans le cas de propagation en mode mixte, l'utilisation de la restriction de la zone de mise à jour est fortement conseillée pour améliorer la performance du calcul et pour éviter la distorsion des level sets, qui peut causer une erreur fatale pendant la mise à jour.

Le calcul du rayon du tore de localisation et sa mise à jour à chaque propagation sont fait de façon automatique. Toutefois, dans le cas où la taille des éléments dans la zone de propagation n'est pas uniforme, le rayon calculé peut être trop petit et un message d'erreur est issu par l'opérateur PROPA_FISS. Dans ce cas l'utilisateur est obligé de donner explicitement une valeur du rayon (voir opérateur RAYON_TORE).

La restriction de la zone de mise à jour peut générer des problèmes de convergence de la méthode simplexe. Dans ce cas l'utilisateur doit forcément utiliser un rayon du tore de localisation plus grand par rapport à celui qui a été calculé automatiquement (écrit dans le fichier .mess) (voir opérateur RAYON_TORE) ou utiliser tout le maillage pour le calcul (ZONE_MAJ='TOUT'). Par contre la méthode upwind est très stable et performante et n'est pas affectée par ce type de problème.

4.6 Opérateur RAYON_TORE

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

Cet opérateur permet de donner la valeur du rayon du tore utilisé pour la restriction de la zone de mise à jour des level sets (voir opérateur ZONE_MAJ). Le rayon du tore est calculé automatiquement par l'opérateur PROPA_FISS et l'utilisateur n'est pas obligé d'utiliser cet opérateur. Toutefois la valeur calculée peut être trop petite dans certain cas, par exemple quand les tailles minimale et maximale des éléments dans la zone de propagation sont très différentes entre elles. Dans tous ces cas, l'utilisateur doit forcément donner le rayon du tore en suivant les conseils contenus dans les messages d'erreur issus par PROPA_FISS.

Il faut remarquer que la valeur du rayon ne peut pas changer beaucoup entre deux propagations successives (voir la documentation R7.02.12 pour plus de détails). Donc il est fortement conseillé d'utiliser la même valeur à chaque appel de PROPA_FISS en donnant la valeur maximale prévue. La valeur du rayon doit être toujours supérieure à une valeur minimale qui peut être estimée en calculant l'addition des valeurs de l'avancée de la fissure (voir opérateur DA_MAX) et du rayon de convergence (voir opérateur RAYON). Toutefois cette valeur est liée aussi à la taille

des éléments dans le tore et donc il est fortement conseillé de ne pas utiliser une valeur inférieure à celle qui a été calculée automatiquement par PROPA_FISS. Une erreur est issue par PROPA_FISS si le rayon donné n'est pas bonne par rapport à la propagation à simuler et à la taille des éléments du maillage.

4.7 Mot clé facteur FISSURE

Cet opérateur obligatoire permet de indiquer les fissures du modèle `mo` qui propagent.

4.7.1 Syntaxe FISSURE

```
♦ FISSURE = _F(  
    ♦ FISS_ACTUELLE = fiss, [fiss_xfem]  
    ♦ FISS_PROPAGEE = CO('FISS'), [TXM]  
    ◇ GRILLE_AUX = griaux, [modele]  
    # Si RAFF_MAIL = 'NON',  
        ♦ TABLE = sif, [table]  
        ◇ NB_POINT_FOND = nbptfd, [I]  
    # Fin si  
)
```

4.7.2 Opérateur FISS_ACTUELLE

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

On donne le nom de la fissure actuelle à propager. Cette fissure doit être déjà affectée au modèle `mo` en utilisant l'opérateur MODI_MODELE_XFEM.

4.7.3 Opérateur FISS_PROPAGEE

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

Cette opérateur permet de définir le nom de la fissure propagée produit par l'opérateur PROPA_FISS. Le résultat produit est un concept de type `fiss_xfem` définissant la fissure propagée.

4.7.4 Opérateur GRILLE_AUX

```
◇ GRILLE_AUX = griaux, [modele]
```

Pour la représentation des level sets de chaque fissure, on peut utiliser le maillage du modèle `mo` ou une grille auxiliaire. Dans ce dernier cas, on donne la grille auxiliaire à utiliser pour la fissure spécifiée par FISS_ACTUELLE en utilisant l'opérateur GRILLE_AUX. Cette grille est prise coïncidente avec les nœuds du maillage utilisés pour le modèle `griax`. Au début de la propagation, la fissure à propager (opérateur FISS_ACTUELLE) doit être définie sur ce modèle en utilisant l'opérateur MODI_MODELE_XFEM. Pour chaque fissure, on doit utiliser un modèle `griax` différent et une seule fissure peut être définie sur chaque grille auxiliaire.

La grille auxiliaire s'étend seulement sur la zone prévue de propagation de la fissure. Cela permet de réduire le temps de calcul. La grille peut être en partie à l'extérieur du maillage du modèle `mo` et aucun rapport n'existe entre ce maillage et celui utilisé pour le modèle `griax`. S'il y a plusieurs fissures, les grilles utilisées sont indépendantes, c'est à dire que les grilles peuvent coïncider en partie ou pas et peuvent être définies en utilisant des maillages qui contiennent des types d'éléments différents.

Il faut remarquer qu'on doit toujours définir les fissures sur le modèle `mo` en utilisant l'opérateur MODI_MODELE_XFEM. En plus, aucun matériau, chargement et condition aux limites ne doivent être affectés au modèle utilisé pour définir la grille auxiliaire.

Si on utilise la méthode *upwind* (voir opérande METHODE_PROPA), le maillage utilisé pour la représentation des level sets doit être très régulier: en 2D on peut utiliser seulement des éléments QUAD4 avec une forme carrée ou rectangulaire et en 3D on peut utiliser seulement des éléments HEXA8 avec une forme cubique ou allongée. Dans tous les cas, les arêtes des éléments doivent être perpendiculaires entre elles. Cela représente une très forte limitation et donc l'utilisation d'une grille auxiliaire est presque obligatoire.

Si on utilise la méthode *simplexe* (voir opérande METHODE_PROPA), on peut utiliser un maillage quelconque.

L'utilisation d'une grille auxiliaire est fortement conseillée parce qu'elle permet de réduire considérablement les temps de calcul.

4.7.5 Opérande TABLE

```
# Si RAFF_MAIL='NON'  
◇ TABLE =   sif,                                     [table]
```

Cet opérande doit être renseigné avec le tableau des facteurs d'intensité des contraintes en fond de la fissure donné par l'opérande FISS_ACTUELLE. Ce tableau est issu par l'opérateur CALC_G. Dans le cas de fond de fissure multiple (voir R7.02.12, paragraphe «Fond de fissure multiple»), pour lequel le fond de la fissure est formé par plusieurs morceaux, l'opérateur CALC_G doit être appelé plusieurs fois, une fois pour chaque morceau (opérande NUME_FOND de CALC_G). On a donc plusieurs tableaux, un tableau pour chaque morceau, qui doivent être combinés dans un seul tableau qui sera donné à PROPA_FISS dans l'opérande TABLE. Pour cela, on peut utiliser l'opérateur CALC_TABLE. Par exemple, dans le cas où la fissure est formée par deux morceaux, la combinaison des deux tableaux peut être réalisée dans la façon suivante:

```
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=('NUME_ORDRE', 'NUME_FOND') , ) ,  
                  ) ,  
                  ) ;
```

et on utilise SIF pour l'opérateur PROPA_FISS. Ce tableau est formé par l'union des deux tableaux issus par CALC_G. Quelconque ordre peut être utilisé dans la combinaison des tableaux.

Rapport de charge R

Le nombre de numéro d'ordre (NUME_ORDRE) [U4.82.03] contenu dans le tableau *sif* peut être égal à 1 ou 2.

Si un seul numéro d'ordre est contenu, l'opérande associe les valeurs des facteurs d'intensité des contraintes à la condition du chargement maximal du cycle de fatigue et il assume que la condition de chargement minimal coïncide avec tous les chargements nuls. Il assume donc un rapport de charge R égal à zéro.

Par contre, si deux numéros d'ordre sont contenus dans chaque tableau, l'opérande assume que les deux conditions de chargement maximal et minimal du cycle de fatigue sont données. L'ordre des valeurs dans le tableau n'est pas important, c'est-à-dire qu'on peut donner comme premier numéro d'ordre la condition de chargement maximal et comme deuxième la condition de chargement minimal ou l'inverse. Les deux sont possibles.

| Si le comportement du modèle est linéaire, on peut utiliser l'opérande COMP_LINE.

4.7.6 Opérande NB_POINT_FOND

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

Si l'opérande NB_POINT_FOND a été utilisé dans CALC_G pour le calcul des facteurs d'intensité des contraintes, on doit forcément utiliser cet opérande dans PROPA_FISS aussi.

L'opérateur PROPA_FISS ne peut pas vérifier si cet opérande a été utilisé ou pas dans CALC_G ou si les valeurs données sont correctes. L'utilisateur doit donc faire attention à utiliser correctement cet opérande!

L'opérande NB_POINT_FOND accepte une liste de valeurs d'entiers. Pour chaque morceau (voir opérande TABLE) de la fissure donné par FISS_ACTUELLE on doit donner une valeur de NB_POINT_FOND égale à la valeur utilisée dans CALC_G . Si l'opérande n'a pas été utilisée dans CALC_G pour un morceau, on utilise la valeur zéro. L'ordre des valeurs de NB_POINT_FOND dans la liste est donné par l'opérande NUME_FOND de CALC_G . Des exemples sont donnés à la fin de ce document.

4.8 Opérande DA_MAX

```
◆ DA_MAX = da, [R]
```

Dans le cas où il y a une seule fissure dans le modèle mo, cet opérande définit l'avance maximale de la fissure.

Dans le cas où il y a plusieurs fissures dans le modèle mo, cet opérande définit l'avance maximale de la fissure qui propage plus vite. Le nombre de cycles de fatigue est calculé en utilisant la vitesse du point du fond de la fissure qui propage la plus vite. Ce nombre de cycles est utilisé pour toutes les fissures du modèle.

4.9 Opérande RAYON

```
◇ RAYON = ray, [R]
```

Cet opérande est utilisé pour définir le rayon du tore autour du fond de fissure dans lequel les résidus vont être estimés pour les étapes de réinitialisation et réorthogonalisation des level sets. Ce tore définit la zone où les level sets doivent être suffisamment proches de fonctions de distances signées.

La valeur de ce rayon doit être toujours supérieure au rayon supérieur d'intégration de l'opérateur CALC_G . L'opérateur PROPA_FISS ne peut pas vérifier si cette condition est respectée! L'utilisateur doit donc faire attention à utiliser correctement cet opérande!

4.10 Mot clé facteur COMP_LINE

```
# Si TEST_MAIL='NON',  
◇ COMP_LINE = _F( ◆ COEFF_MULT_MINI = cmin, [R]  
                  ◆ COEFF_MULT_MAXI = cmax, [R]  
                ),
```

Cet opérande peut être utilisé dans tous le cas où le comportement du modèle fissuré est linéaire. En fait dans ce cas la solution du modèle est toujours proportionnelle à la valeur du chargement. On peut donc réduire le temps du calcul en utilisant cette propriété.

D'abord on calcule les facteurs d'intensité des contraintes en utilisant **une configuration de référence** des chargements. Cette configuration peut être coïncidente avec la condition de chargement maximal ou minimal du cycle de fatigue ou pas. Un seul numéro d'ordre est contenu dans le tableau des facteurs d'intensité des contraintes. On donne ce tableau à l'opérande TABLE. Ensuite on peut donner les conditions de chargement minimal et maximal du cycle de fatigue en utilisant les paramètres COEFF_MULT_MINI et COEFF_MULT_MAXI de l'opérande COMP_LINE. Les deux valeurs sont les constantes par lesquelles on doit multiplier les chargements de la configuration de référence pour obtenir les conditions de chargement minimal et maximal du cycle

de fatigue. Donc si la configuration de référence choisie coïncide avec la condition de chargement minimal ou maximal, la valeur de `COEFF_MULT_MINI` ou `COEFF_MULT_MAXI`, respectivement, est égale à 1.

5 Opérandes spécifiques à la méthode MAILLAGE

5.1 Opérandes MA_STRUC

♦ `MA_STRUC = mast,` [maillage]

Cet opérande permet de définir le maillage de la structure saine (i.e. sans fissure).

5.2 Opérandes MA_XFEM1 et FISSURE1

♦ `MA_XFEM1 = max1,` [maillage]
♦ `FISSURE1 = FISS1,` [fiss_xfem]

Cet opérande permet de définir la fissure que l'on veut propager à l'aide de son maillage surfacique `MA_XFEM1`, et du concept `FISS1` produit par l'opérateur `DEFI_FISS_XFEM`.

Le maillage de la fissure `max1` doit respecter certaines règles de nommage, cf. §4.

5.3 Opérande ITERATION

♦ `ITERATION = iter,` [I]

Cet opérande permet de définir le numéro de l'itération du calcul de propagation. La première itération (propagation de la fissure initiale) est l'itération 1.

Cet opérande facilite la réalisation des calculs : il est ainsi possible de relancer un calcul de propagation sur une fissure ayant déjà été propagée `nb_init` fois.

5.4 Opérandes DTAN_ORIG et DTAN_EXTR

◇ `DTAN_ORIG = (xo,yo,zo),` [l_R]
◇ `DTAN_EXTR = (xe,ye,ze),` [l_R]

Ces opérandes permettent de préciser, en 3D, la direction de propagation des nœuds origine et extrémité de la fissure. Si ces opérandes ne sont pas spécifiés, la direction de propagation est orthogonale au fond de fissure.

5.5 Opérandes TABLE et METHODE_POSTK

♦ `TABLE = sif,` [table]
◇ `METHODE_POSTK = /1`
 /2
 /3

Le mot-clé facteur `TABLE` doit être renseigné avec le tableau des facteurs d'intensité des contraintes en fond de fissure, calculés par les opérateurs `CALC_G` ou `POST_K1_K2_K3`.

Si la table a été calculée avec `POST_K1_K2_K3`, il faut obligatoirement indiquer avec quelle méthode d'interpolation des sauts de déplacements doivent être calculés les facteurs d'intensité des contraintes (`METHODE_POSTK = 1, 2 ou 3`).

Si le mot clé `METHODE_POSTK` est absent, on suppose que la table a été calculée avec `CALC_G`.

5.6 Opérande HYPOTHESE

```
◇ HYPOTHESE = / ' PLAN '
              / ' NON_PLAN ' [DEFAULT]
```

On peut forcer une propagation plane de la fissure en indiquant `HYPOTHESE='PLAN'`. Par défaut, la propagation est non plane, i.e. la fissure peut bifurquer si elle est sollicitée en mode mixte.

5.7 Opérandes `MA_XFEM2` et `MA_TOT2`

```
◇ MA_XFEM2 = CO('max2'), [TXM]
◆ MA_TOT2  = CO('matot2'), [TXM]
```

Ces opérandes permettent de définir :

- le nom du maillage surfacique correspondant à la fissure propagée `MA_XFEM2`
- le nom du maillage total, i.e. la concaténation du maillage de la structure saine (cf. opérande `MA_STRUC`) et du maillage de la fissure propagée `MA_XFEM2`.

C'est ce maillage total `MA_TOT2` qui doit être utilisé pour définir le modèle et la nouvelle fissure dans `DEFI_FISS_XFEM`.

6 Opérands spécifiques à la méthode INITIALISATION

6.1 Opérands MA_STRUC

- ♦ `MA_STRUC = mast,` [maillage]
Cet opérateur permet de définir le maillage de la structure saine (i.e. sans fissure).

6.2 Opérands FORM_FISS

- ♦ `FORM_FISS = / 'DEMI_DROITE'`
`/ 'DEMI_PLAN'`

Cet opérateur permet de définir le type de la fissure : une demi-droite en 2D ou une fissure plane à fond rectiligne en 3D.

6.3 Cas DEMI_DROITE : opérateurs PFON et DTAN_DTAN

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

Ces opérateurs permettent de définir la fissure initiale en 2D en donnant la coordonnée du point du fond de fissure `PFON` et la direction de propagation de celle-ci `DTAN`.

6.4 Cas DEMI_PLAN : opérateurs PFON, POINT_ORIG, POINT_EXTR et 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]

Ces opérateurs permettent de définir la fissure initiale en 3D en donnant les coordonnées des points origine et extrémité du fond de fissure et la direction de propagation de celle-ci `DTAN`. Le fond de fissure sera discrétisé avec `nbfond` noeuds.

6.5 Opérateurs MA_XFEM2 et MA_TOT2

- ♦ `MA_XFEM2 = CO('max2'),` [TXM]
- ♦ `MA_TOT2 = CO('matot2'),` [TXM]

Ces opérateurs permettent de définir :

- le nom du maillage surfacique correspondant à la fissure initiale `MA_XFEM2`
- le nom du maillage total, i.e. la concaténation du maillage de la structure saine (cf. opérateur `MA_STRUC`) et du maillage de la fissure `MA_XFEM2`.

C'est ce maillage total `MA_TOT2` qui doit être utilisé pour définir le modèle et la fissure dans `DEFI_FISS_XFEM`.

7 Exemples de propagation de fissure avec X-FEM

Plusieurs cas-tests permettent de valider la propagation de fissures avec X-FEM dans une plaque:

- sslp314 : propagation en 2D d'une fissure inclinée;
- sslv314 : propagation en 3D d'une fissure droite (mode I pur);
- sslv315 : propagation en 3D d'une fissure inclinée;
- sslv316 : fissuration à propagation imposée avec x-fem;
- sslp318 : propagation d'une fissure x-fem non débouchante sollicitée en mode I
- sslp319 : propagation de deux fissures x-fem débouchantes sollicitée en mode I
- ssnv185 : propagation de fissure débouchante dans une plaque 3D
- zzzz255 : validation de l'option TEST_MAIL dans PROPA_FISS

On donne ci-dessous des exemples d'utilisation des différentes méthodes disponibles dans PROPA_FISS.

7.1 Cas METHODE_PROPA='MAILLAGE'

L'exemple ci-dessous correspond à la propagation en 3D d'une fissure plane à fond rectiligne.

```
MAST = LIRE_MAILLAGE();
PROPA_FISS( METHODE_PROPA= 'INITIALISATION',
            MA_STRUC      = MAST,
            MA_XFEM2      = MAX[0],
            MA_TOT2       = MA[0],
            FORM_FISS     = 'DEMI_PLAN',
            POINT_ORIG    = (0., 2., 9.),
            POINT_EXTR    = (1., 2., 9.),
            DTAN          = (0., 1., 0.),
            NB_POINT_FOND= 5,
            );

for i in range(nbcalc) :

    MODELE[i]=AFFE_MODELE( MAILLAGE=MA[i],
                          AFPE= (_F( GROUP_MA=('VOL',),
                                      PHENOMENE='MECANIQUE',
                                      MODELISATION='3D',),
                                _F(GROUP_MA=('SURFINF','SURFSUP',),
                                    PHENOMENE='MECANIQUE',
                                    MODELISATION='3D',),
                                );

    FF[i]=DEFI_FISS_XFEM( DEFI_FISS=_F( GROUP_MA_FISS='FISS_'+str(i),
                                         GROUP_MA_FOND='FOND_'+str(i)),
                        GROUP_MA_ENRI='VOL',
                        ORIE_FOND=_F(...),
                        MODELE=MODELE[i],
                        );

    MODELX[i]=MODI_MODELE_XFEM(MODELE_IN=MODELE[i],FISSURE=FF[i],INFO=1,);

    RESU[i]=STAT_NON_LINE( MODELE=MODELX[i],
                          ...);

    SIF[i]=CALC_G( RESULTAT=RESU[i],
                  OPTION='CALC_K_G',
                  THETA=_F( FISSURE=FF[i],
                           R_INF=RI,R_SUP=RS, ),
```



```
);  
  
MAX[i+1] = CO('MAX_%d'%(i+1))  
MA[i+1] = CO('MA_%d'%(i+1))  
PROPA_FISS( TABLE=SIF[i],  
             METHODE_PROPA='MAILLAGE',  
             LOI_PROPA=_F(LOI='PARIS',  
                           M=1.,  
                           DA_MAX=0.3, ),  
             MA_STRUC=MAILLAG1,  
             MA_XFEM1=MAX[i],  
             MA_XFEM2=MAX[i+1],  
             MA_TOT2=MA[i+1],  
             ITERATION=i+1,  
             FISSURE1=FF[i],  
             HYPOTHESE='PLAN',  
             INFO=2, );
```

7.2 Cas METHODE_PROPA='SIMPLEXE' et 'UPWIND'

Dans ce paragraphe on donne des exemples d'utilisation de l'opérateur PROPA_FISS en utilisant les méthodes *simplexe* ou *upwind*. On donne d'abord un cas très simple (cas 1) pour comprendre l'utilisation de base et ensuite on le complique en ajoutant d'autres fissures ou des morceaux de fissure (voir R7.02.12, paragraphe «Fond de fissure multiple») et en utilisant des grilles auxiliaires. Les changements du fichier .comm entre deux cas successifs sont toujours indiqués.

7.2.1 Cas 1: une seule fissure en un seul morceau, aucune grille auxiliaire

```
MODELE[0] = AFFE_MODELE( MAILLAGE=maillage,
                        AFFE= (_F( GROUP_MA=('volume',),
                                PHENOMENE='MECANIQUE',
                                MODELISATION='3D',),),);

FISS[0]      = DEFI_FISS_XFEM( MODELE=MODELE[0],
                              ...,);

for i in range(1,n) :

    MODELE[i] = MODI_MODELE_XFEM( MODELE_IN=MODELE[0],
                                FISSURE=FISS[i-1],);

    CHXFEM[i] = AFFE_CHAR_MECA( MODELE=MODELE[i],
                                LIAISON_XFEM='OUI',);

    RESU[i]    = STAT_NON_LINE( MODELE=MODELE[i],
                                EXCIT=(_F(CHARGE=CHXFEM[i],),
                                ...,),
                                ...,);

    MAXFEM[i] = POST_MAIL_XFEM( MODELE=MODELE[i],);

    DEPL[i]    = POST_CHAM_XFEM( MAILLAGE=MAXFEM[i],
                                MODELE=MODELE[i],
                                RESULTAT=RESU[i],
                                NOM_CHAM='DEPL',);

    FIC[i] = CALC_G ( MODELE=MODELE[i],
                      RESULTAT=RESU[i],
                      OPTION='CALC_K_G',
                      THETA=_F( FISSURE=FISS[i-1],
                                R_INF=RI[i],
                                R_SUP=RS[i],
                                ),
                      ...,);

    FISS[i] = CO('FISS_d'%(i))
    PROPA_FISS(MODELE=MODELE[i],
               FISSURE=_F(FISS_ACTUELLE=FISS[i-1],
                           FISS_PROPAGEE=FISS[i],
                           TABLE=FIC[i],),
               METHODE_PROPA='UPWIND',
               DA_MAX=...,
               RAYON=RS[i],
               LOI_PROPA=_F(LOI='PARIS',
                             C=...,
                             M=...,
                             N=...,
                             MATER=...),
               INFO=1,);
```

7.2.2 Cas 2: deux fissures en un morceau chacune, aucune grille auxiliaire

```
MODELE[0] = AFPE_MODELE( MAILLAGE=maillage,
                        AFPE= (_F( GROUP_MA=('volume',),
                                PHENOMENE='MECANIQUE',
                                MODELISATION='3D',),),),);

# On va définir les deux fissures dans le modèle MODELE[0]
FISS1[0] = DEF_FISS_XFEM( MODELE=MODELE[0],
                        ...,);
FISS2[0] = DEF_FISS_XFEM( MODELE=MODELE[0],
                        ...,);

for i in range(1,n) :

    # On affecte les deux fissures au modèle MODELE[0]. L'ordre des
    # fissures dans la liste (FISSURE) n'est pas significatif.
    MODELE[i] = MODI_MODELE_XFEM( MODELE_IN=MODELE[0],
                                FISSURE=(FISS1[i-1],FISS2[i-1]));

    CHXFEM[i] = AFPE_CHAR_MECA( MODELE=MODELE[i],
                                LIAISON_XFEM='OUI',);

    RESU[i] = STAT_NON_LINE( MODELE=MODELE[i],
                            EXCIT=(_F(CHARGE=CHXFEM[i],),
                                ...,),
                            ...,);

    MAXFEM[i] = POST_MAIL_XFEM( MODELE=MODELE[i],);

    DEPL[i] = POST_CHAM_XFEM( MAILLAGE=MAXFEM[i],
                            MODELE=MODELE[i],
                            RESULTAT=RESU[i],
                            NOM_CHAM='DEPL',);

    # On calcule les facteurs d'intensité des contraintes
    # pour la première fissure FISS1[i-1]
    FIC1[i] = CALC_G ( MODELE=MODELE[i],
                    RESULTAT=RESU[i],
                    OPTION='CALC_K_G',
                    THETA=_F( FISSURE=FISS1[i-1],
                            R_INF=RI[i],
                            R_SUP=RS[i],
                            ),
                    ...,);

    # On calcule les facteurs d'intensité des contraintes
    # pour la deuxième fissure FISS2[i-1]
    FIC2[i] = CALC_G ( MODELE=MODELE[i],
                    RESULTAT=RESU[i],
                    OPTION='CALC_K_G',
                    THETA=_F( FISSURE=FISS2[i-1],
                            R_INF=RI[i],
                            R_SUP=RS[i],
                            ),
                    ...,);

    # On propage les deux fissures
    FISS1[i] = CO('FISS1_%d'%(i))
    FISS2[i] = CO('FISS2_%d'%(i))
    PROPA_FISS(MODELE=MODELE[i],
```

```
FISSURE=( _F(FISS_ACTUELLE=FISS1[i-1],
             FISS_PROPAGEE=FISS1[i],
             TABLE=FIC1[i],),
          _F(FISS_ACTUELLE=FISS2[i-1],
             FISS_PROPAGEE=FISS2[i],
             TABLE=FIC2[i],),),
METHODE_PROPA='UPWIND',
DA_MAX=...,
RAYON=RS[i],
LOI_PROPA=_F(LOI='PARIS',
             C=...,
             M=...,
             N=...,
             MATER=...),
INFO=1,);
```

7.2.3 Cas 3: une seule fissure en deux morceaux, aucune grille auxiliaire

```
MODELE[0] = AFFE_MODELE( MAILLAGE=maillage,
                        AFFE= (_F( GROUP_MA=('volume',),
                                PHENOMENE='MECANIQUE',
                                MODELISATION='3D',),),);

# On va définir la fissure dans le modèle MODELE[0]. Le fond
# de la fissure est constitué de deux morceaux. Cela est déterminé
# automatiquement par l'opérateur DEFI_FISS_XFEM (voir le fichier
# .mess pour avoir les détails!)
FISS[0] = DEFI_FISS_XFEM( MODELE=MODELE[0],
                        ...,);

for i in range(1,n) :

    MODELE[i] = MODI_MODELE_XFEM( MODELE_IN=MODELE[0],
                                FISSURE=FISS[i-1]);

    CHXFEM[i] = AFFE_CHAR_MECA( MODELE=MODELE[i],
                                LIAISON_XFEM='OUI',);

    RESU[i] = STAT_NON_LINE( MODELE=MODELE[i],
                            EXCIT=( _F(CHARGE=CHXFEM[i],),
                                    ...,),
                            ...,);

    MAXFEM[i] = POST_MAIL_XFEM( MODELE=MODELE[i],);

    DEPL[i] = POST_CHAM_XFEM( MAILLAGE=MAXFEM[i],
                             MODELE=MODELE[i],
                             RESULTAT=RESU[i],
                             NOM_CHAM='DEPL',);

# On calcule les facteurs d'intensité des contraintes
# pour le premier morceau de la fissure
FICa[i] = CALC_G ( MODELE=MODELE[i],
                  RESULTAT=RESU[i],
                  OPTION='CALC_K_G',
                  THETA=_F( FISSURE=FISS[i-1],
                          NUME_FOND=1,
                          R_INF=RI[i],
                          R_SUP=RS[i],
                          ),
                  ...,);
```

```
# On calcule les facteurs d'intensité des contraintes
# pour le deuxième morceau de la fissure
FICb[i] = CALC_G ( MODELE=MODELE[i],
                   RESULTAT=RESU[i],
                   OPTION='CALC_K_G',
                   THETA=_F( FISSURE=FISS[i-1],
                           NUME_FOND=2,
                           R_INF=RI[i],
                           R_SUP=RS[i],
                           ),
                   ..., );

# On fait l'union des deux tableaux des deux morceaux
SIF[i]=CALC_TABLE( TABLE= FICa[i] ,
                   ACTION=( _F(OPERATION='COMB',
                               TABLE = FICb[i] ,
                               NOM_PARA=('NUME_ORDRE',
                                         'NUME_FOND', ),
                               ),
                   ),
                   );

# On propage la fissure
FISS[i] = CO('FISS_%d'%(i))
PROPA_FISS(MODELE=MODELE[i],
           FISSURE=_F(FISS_ACTUELLE=FISS[i-1],
                     FISS_PROPAGEE=FISS[i],
                     TABLE=SIF[i], ),
           METHODE_PROPA='UPWIND',
           DA_MAX=...,
           RAYON=RS[i],
           LOI_PROPA=_F(LOI='PARIS',
                       C=...,
                       M=...,
                       N=...,
                       MATER=...),
           INFO=1, );
```

7.2.4 Cas 4: deux fissures en deux morceaux pour la première et en un seul morceau pour la deuxième, aucune grille auxiliaire

```
MODELE[0] = AFFE_MODELE( MAILLAGE=maillage,
                        AFFE= (_F( GROUP_MA=('volume', ),
                                PHENOMENE='MECANIQUE',
                                MODELISATION='3D', ), ), );

# On va définir les deux fissures dans le modèle MODELE[0].
# Le fond de la première fissure est constitué par trois
# morceaux. Cela est déterminé automatiquement par l'opérateur
# DEFI_FISS_XFEM (voir le fichier .mess pour avoir les détails!)
FISS1[0] = DEFI_FISS_XFEM( MODELE=MODELE[0],
                          ..., );

# On va définir la deuxième fissure (un seul morceau)
FISS2[0] = DEFI_FISS_XFEM( MODELE=MODELE[0],
                          ..., );

for i in range(1,n) :

    # On affecte les deux fissures au modèle MODELE[0]. L'ordre des
    # fissures dans la liste (FISSURE) n'est pas significatif.
    MODELE[i] = MODI_MODELE_XFEM( MODELE_IN=MODELE[0],
                                FISSURE=(FISS1[i-1], FISS2[i-1]));
```

```
CHXFEM[i] = AFFE_CHAR_MECA( MODELE=MODELE[i],  
                             LIAISON_XFEM='OUI', );  
  
RESU[i] = STAT_NON_LINE( MODELE=MODELE[i],  
                           EXCIT=( _F( CHARGE=CHXFEM[i], ),  
                           ..., ),  
                           ..., );  
  
MAXFEM[i] = POST_MAIL_XFEM( MODELE=MODELE[i], );  
  
DEPL[i] = POST_CHAM_XFEM( MAILLAGE=MAXFEM[i],  
                           MODELE=MODELE[i],  
                           RESULTAT=RESU[i],  
                           NOM_CHAM='DEPL', );  
  
# On calcule les facteurs d'intensité des contraintes  
# pour le premier morceau de la première fissure  
FIC1a[i] = CALC_G ( MODELE=MODELE[i],  
                    RESULTAT=RESU[i],  
                    OPTION='CALC_K_G',  
                    THETA=_F( FISSURE=FISS1[i-1],  
                               NUME_FOND=1,  
                               R_INF=RI[i],  
                               R_SUP=RS[i],  
                               ),  
                    ..., );  
  
# On calcule les facteurs d'intensité des contraintes  
# pour le deuxième morceau de la première fissure  
FIC1b[i] = CALC_G ( MODELE=MODELE[i],  
                    RESULTAT=RESU[i],  
                    OPTION='CALC_K_G',  
                    THETA=_F( FISSURE=FISS1[i-1],  
                               NUME_FOND=2,  
                               R_INF=RI[i],  
                               R_SUP=RS[i],  
                               ),  
                    ..., );  
  
# On fait l'union des deux tableaux des deux morceaux  
# de la première fissure  
SIF1[i]=CALC_TABLE( TABLE= FIC1a[i] ,  
                    ACTION=( _F( OPERATION='COMB',  
                                   TABLE = FIC1b[i] ,  
                                   NOM_PARA=('NUME_ORDRE',  
                                              'NUME_FOND', ),  
                                   ),  
                    );  
  
# On calcule les facteurs d'intensité des contraintes  
# pour la deuxième fissure  
SIF2[i] = CALC_G ( MODELE=MODELE[i],  
                    RESULTAT=RESU[i],  
                    OPTION='CALC_K_G',  
                    THETA=_F( FISSURE=FISS2[i-1],  
                               R_INF=RI[i],  
                               R_SUP=RS[i],  
                               ),  
                    ..., );  
  
# On propage les deux fissures
```

```
FISS1[i] = CO('FISS1_%d'%(i))
FISS2[i] = CO('FISS2_%d'%(i))
PROPA_FISS(MODELE=MODELE[i],
           FISSURE=(_F(FISS_ACTUELLE=FISS1[i-1],
                       FISS_PROPAGEE=FISS1[i],
                       TABLE=SIF1[i]),
                     _F(FISS_ACTUELLE=FISS2[i-1],
                       FISS_PROPAGEE=FISS2[i],
                       TABLE=SIF2[i]),)),
           METHODE_PROPA='UPWIND',
           DA_MAX=...,
           RAYON=RS[i],
           LOI_PROPA=_F(LOI='PARIS',
                        C=...,
                        M=...,
                        N=...,
                        MATER=...),
           INFO=1,);
```

7.2.5 Cas 5: deux fissures en deux morceaux pour la première et en un seul morceau pour la deuxième, une grille auxiliaire utilisée pour chaque fissure

```
MODELE[0] = AFFE_MODELE( MAILLAGE=maillage,
                        AFFE= (_F( GROUP_MA=('volume',),
                                PHENOMENE='MECANIQUE',
                                MODELISATION='3D',),),);

# On va définir les deux fissures dans le modèle MODELE[0].
# Le fond de la première fissure est constitué des deux
# morceaux. Cela est déterminé automatiquement par l'opérateur
# DEFI_FISS_XFEM (voir le fichier MESS pour avoir les détails!)
FISS1[0] = DEFI_FISS_XFEM( MODELE=MODELE[0],
                          ...,);

# On va définir la deuxième fissure (un seul morceau)
FISS2[0] = DEFI_FISS_XFEM( MODELE=MODELE[0],
                          ...,);

# -----
# GRILLE1 POUR LA PREMIERE FISSURE
# On va définir la grille utilisée pour la première fissure
GRILLE1 = AFFE_MODELE( MAILLAGE=grd1,
                      AFFE= (_F( GROUP_MA=('volume',),
                                PHENOMENE='MECANIQUE',
                                MODELISATION='3D',),),);

# On va définir la même fissure FISS1 sur la grille
FISGR1 = DEFI_FISS_XFEM( MODELE=GRILLE1,
                        ...,);

# On affecte la fissure à la grille
FGR1 = MODI_MODELE_XFEM( MODELE_IN=GRILLE1,
                        FISSURE=FISGR1);

# -----
# GRILLE2 POUR LA DEUXIEME FISSURE
# On va définir la grille utilisée pour la deuxième fissure
GRILLE2 = AFFE_MODELE( MAILLAGE=grd2,
                      AFFE= (_F( GROUP_MA=('volume',),
                                PHENOMENE='MECANIQUE',
                                MODELISATION='3D',),),);
```

```
# On va définir la même fissure FISS2 sur la grille
FISGR2 = DEFI_FISS_XFEM( MODELE=GRILLE2,
                        ..., );

# On affecte la fissure à la grille
FGR2 = MODI_MODELE_XFEM( MODELE_IN=GRILLE2,
                        FISSURE=FISS2 );

# -----
# PROPAGATION

for i in range(1,n) :

    # On affecte les deux fissures au modèle MODELE[0]. L'ordre des
    # fissures dans la liste (FISSURE) n'est pas significatif.
    MODELE[i] = MODI_MODELE_XFEM( MODELE_IN=MODELE[0],
                                FISSURE=(FISS1[i-1], FISS2[i-1]));

    CHXFEM[i] = AFFE_CHAR_MECA( MODELE=MODELE[i],
                                LIAISON_XFEM='OUI', );

    RESU[i] = STAT_NON_LINE( MODELE=MODELE[i],
                             EXCIT=( _F(CHARGE=CHXFEM[i], ),
                                     ..., ),
                             ..., );

    MAXFEM[i] = POST_MAIL_XFEM( MODELE=MODELE[i], );

    DEPL[i] = POST_CHAM_XFEM( MAILLAGE=MAXFEM[i],
                             MODELE=MODELE[i],
                             RESULTAT=RESU[i],
                             NOM_CHAM='DEPL', );

    # On calcule les facteurs d'intensité des contraintes
    # pour le premier morceau de la première fissure
    FIC1a[i] = CALC_G ( MODELE=MODELE[i],
                        RESULTAT=RESU[i],
                        OPTION='CALC_K_G',
                        THETA=_F( FISSURE=FISS1[i-1],
                                NUME_FOND=1,
                                R_INF=RI[i],
                                R_SUP=RS[i],
                                ),
                        ..., );

    # On calcule les facteurs d'intensité des contraintes
    # pour le deuxième morceau de la première fissure
    FIC1b[i] = CALC_G ( MODELE=MODELE[i],
                        RESULTAT=RESU[i],
                        OPTION='CALC_K_G',
                        THETA=_F( FISSURE=FISS1[i-1],
                                NUME_FOND=2,
                                R_INF=RI[i],
                                R_SUP=RS[i],
                                ),
                        ..., );

    # On fait l'union des deux tableaux des deux morceaux
    # de la première fissure
    SIF1[i]=CALC_TABLE( TABLE= FIC1a[i] ,
                        ACTION=( _F(OPERATION='COMB',
                                    TABLE = FIC1b[i] ,
```



```
NOM_PARA= ('NUME_ORDRE',  
           'NUME_FOND',),  
),  
);  
  
# On calcule les facteurs d'intensité des contraintes  
# pour la deuxième fissure  
SIF2[i] = CALC_G ( MODELE=MODELE[i],  
                  RESULTAT=RESU[i],  
                  OPTION='CALC_K_G',  
                  THETA=_F( FISSURE=FISS2[i-1],  
                           R_INF=RI[i],  
                           R_SUP=RS[i],  
                           ),  
                  ... );  
  
# On propage les deux fissures  
FISS1[i] = CO('FISS1_%d'%(i))  
FISS2[i] = CO('FISS2_%d'%(i))  
PROPA_FISS(MODELE=MODELE[i],  
           FISSURE=( _F(FISS_ACTUELLE=FISS1[i-1],  
                       FISS_PROPAGEE=FISS1[i],  
                       TABLE=SIF1[i],  
                       GRILLE_AUX=FGR1, ),  
                   _F(FISS_ACTUELLE=FISS2[i-1],  
                       FISS_PROPAGEE=FISS2[i],  
                       TABLE=SIF2[i],  
                       GRILLE_AUX=FGR2, ) ),  
           METHODE_PROPA='UPWIND',  
           DA_MAX=...,  
           RAYON=RS[i],  
           LOI_PROPA=_F(LOI='PARIS',  
                        C=...,  
                        M=...,  
                        N=...,  
                        MATER=...),  
           INFO=1, );
```

7.2.6 Cas 6: deux fissures en deux morceaux pour la première et en un seul morceau pour la deuxième, une grille auxiliaire utilisée pour chaque fissure, NB_POINT_FOND utilisé

```
MODELE[0] = AFFE_MODELE( MAILLAGE=maillage,  
                        AFFE= (_F( GROUP_MA=('volume',),  
                                   PHENOMENE='MECANIQUE',  
                                   MODELISATION='3D', ), ), );  
  
# On va définir les deux fissures dans le modèle MODELE[0].  
# Le fond de la première fissure est constitué des deux  
# morceaux. Cela est déterminé automatiquement par l'opérateur  
# DEFI_FISS_XFEM (voir le fichier MESS pour avoir les détails!)  
FISS1[0] = DEFI_FISS_XFEM( MODELE=MODELE[0],  
                          ... );  
  
# On va définir la deuxième fissure (un seul morceau)  
FISS2[0] = DEFI_FISS_XFEM( MODELE=MODELE[0],  
                          ... );  
  
# -----  
# GRILLE1 POUR LA PREMIERE FISSURE  
# On va définir la grille utilisée pour la première fissure
```

```
GRILLE1 = AFFE_MODELE( MAILLAGE=grd1,
                        AFPE= (_F( GROUP_MA=('volume',),
                                   PHENOMENE='MECANIQUE',
                                   MODELISATION='3D',),),),);

# On va définir la même fissure FISS1 sur la grille
FISGR1 = DEFI_FISS_XFEM( MODELE=GRILLE1,
                        ...,);

# On affecte la fissure à la grille
FGR1 = MODI_MODELE_XFEM( MODELE_IN=GRILLE1,
                        FISSURE=FISGR1);

# -----
# GRILLE2 POUR LA DEUXIEME FISSURE
# On va définir la grille utilisée pour la deuxième fissure
GRILLE2 = AFFE_MODELE( MAILLAGE=grd2,
                        AFPE= (_F( GROUP_MA=('volume',),
                                   PHENOMENE='MECANIQUE',
                                   MODELISATION='3D',),),),);

# On va définir la même fissure FISS2 sur la grille
FISGR2 = DEFI_FISS_XFEM( MODELE=GRILLE2,
                        ...,);

# On affecte la fissure à la grille
FGR2 = MODI_MODELE_XFEM( MODELE_IN=GRILLE2,
                        FISSURE=FISGR2);

# -----
# PROPAGATION

for i in range(1,n) :

    # On affecte les deux fissures au modèle MODELE[0]. L'ordre des
    # fissures dans la liste (FISSURE) n'est pas significatif.
    MODELE[i] = MODI_MODELE_XFEM( MODELE_IN=MODELE[0],
                                FISSURE=(FISS1[i-1],FISS2[i-1]));

    CHXFEM[i] = AFFE_CHAR_MECA( MODELE=MODELE[i],
                                LIAISON_XFEM='OUI',);

    RESU[i] = STAT_NON_LINE( MODELE=MODELE[i],
                             EXCIT=(_F(CHARGE=CHXFEM[i],),
                                     ...,),
                             ...,);

    MAXFEM[i] = POST_MAIL_XFEM( MODELE=MODELE[i],);

    DEPL[i] = POST_CHAM_XFEM( MAILLAGE=MAXFEM[i],
                             MODELE=MODELE[i],
                             RESULTAT=RESU[i],
                             NOM_CHAM='DEPL',);

    # On calcule les facteurs d'intensité des contraintes
    # pour le premier morceau de la première fissure.
    # On utilise NB_POINT_FOND pour ce morceau.
    FIC1a[i] = CALC_G ( MODELE=MODELE[i],
                        RESULTAT=RESU[i],
                        OPTION='CALC_K_G',
                        THETA=_F( FISSURE=FISS1[i-1],
                                NUME_FOND=1,
```

```

                                NB_POINT_FOND=15,
                                R_INF=RI[i],
                                R_SUP=RS[i],
                                ),
                                ...);

# On calcule les facteurs d'intensité des contraintes
# pour le deuxième morceau de la première fissure
FIC1b[i] = CALC_G ( MODELE=MODELE[i],
                    RESULTAT=RESU[i],
                    OPTION='CALC_K_G',
                    THETA=_F( FISSURE=FISS1[i-1],
                               NUME_FOND=2,
                               R_INF=RI[i],
                               R_SUP=RS[i],
                               ),
                    ...);

# On fait l'union des deux tableaux des deux morceaux
# de la première fissure
SIF1[i]=CALC_TABLE( TABLE= FIC1a[i] ,
                    ACTION=( _F(OPERATION='COMB',
                                  TABLE = FIC1b[i] ,
                                  NOM_PARA=('NUME_ORDRE',
                                             'NUME_FOND',),
                                  ),
                    ),
                    );

# On calcule les facteurs d'intensité des contraintes
# pour la deuxième fissure
# On utilise NB_POINT_FOND pour cette fissure.
FIC2[i] = CALC_G ( MODELE=MODELE[i],
                    RESULTAT=RESU[i],
                    OPTION='CALC_K_G',
                    THETA=_F( FISSURE=FISS2[i-1],
                               NB_POINT_FOND=5,
                               R_INF=RI[i],
                               R_SUP=RS[i],
                               ),
                    ...);

# On propage les deux fissures
FISS1[i] = CO('FISS1_%d'%(i))
FISS2[i] = CO('FISS2_%d'%(i))
PROPA_FISS(MODELE=MODELE[i],
            FISSURE=( _F(FISS_ACTUELLE=FISS1[i-1],
                          FISS_PROPAGEE=FISS1[i],
                          TABLE=SIF1[i],
                          GRILLE_AUX=FGR1,
                          NB_POINT_FOND=(15,0),),
                      _F(FISS_ACTUELLE=FISS2[i-1],
                          FISS_PROPAGEE=FISS2[i],
                          TABLE=SIF2[i],
                          GRILLE_AUX=FGR2,
                          NB_POINT_FOND=5),),
            METHODE_PROPA='UPWIND',
            DA_MAX=...,
            RAYON=RS[i],
            LOI_PROPA=_F(LOI='PARIS',
                          C=...,
                          M=...,
                          N=...,
```

```
MATER=...),  
INFO=1,);
```

7.2.7 Cas 7: deux fissures en deux morceaux pour la première et en un seul morceau pour la deuxième, une grille auxiliaire utilisée pour chaque fissure, vérification du maillage

```
MODELE[0] = AFFE_MODELE( MAILLAGE=maillage,  
                           AFFE= (_F( GROUP_MA=('volume',),  
                                       PHENOMENE='MECANIQUE',  
                                       MODELISATION='3D',),),),);  
  
# On va définir les deux fissures dans le modèle MODELE[0].  
# Le fond de la première fissure est constitué des deux  
# morceaux. Cela est déterminé automatiquement par l'opérateur  
# DEFI_FISS_XFEM (voir le fichier MESS pour avoir les détails!)  
FISS1[0] = DEFI_FISS_XFEM( MODELE=MODELE[0],  
                           ...,);  
  
# On va définir la deuxième fissure (un seul morceau)  
FISS2[0] = DEFI_FISS_XFEM( MODELE=MODELE[0],  
                           ...,);  
  
# -----  
# GRILLE1 POUR LA PREMIERE FISSURE  
# On va définir la grille utilisée pour la première fissure  
GRILLE1 = AFFE_MODELE( MAILLAGE=grd1,  
                        AFFE= (_F( GROUP_MA=('volume',),  
                                    PHENOMENE='MECANIQUE',  
                                    MODELISATION='3D',),),),);  
  
# On va définir la même fissure FISS1 sur la grille  
FISGR1 = DEFI_FISS_XFEM( MODELE=GRILLE1,  
                        ...,);  
  
# On affecte la fissure à la grille  
FGR1 = MODI_MODELE_XFEM( MODELE_IN=GRILLE1,  
                        FISSURE=FISGR1);  
  
# -----  
# GRILLE2 POUR LA DEUXIEME FISSURE  
# On va définir la grille utilisée pour la deuxième fissure  
GRILLE2 = AFFE_MODELE( MAILLAGE=grd2,  
                        AFFE= (_F( GROUP_MA=('volume',),  
                                    PHENOMENE='MECANIQUE',  
                                    MODELISATION='3D',),),),);  
  
# On va définir la même fissure FISS2 sur la grille  
FISGR2 = DEFI_FISS_XFEM( MODELE=GRILLE2,  
                        ...,);  
  
# On affecte la fissure à la grille  
FGR2 = MODI_MODELE_XFEM( MODELE_IN=GRILLE2,  
                        FISSURE=FISGR2);  
  
# -----  
# PROPAGATION  
  
for i in range(1,n) :  
  
    # On affecte les deux fissures au modèle MODELE[0]. L'ordre des  
    # fissures dans la liste (FISSURE) n'est pas significatif.
```

```
MODELE[i] = MODI_MODELE_XFEM( MODELE_IN=MODELE[0],  
                               FISSURE=(FISS1[i-1],FISS2[i-1]));  
  
# On propage les deux fissures pour vérifier si les deux grilles  
# auxiliaires utilisée sont suffisamment raffinées  
FISS1[i] = CO('FISS1_%d'%(i))  
FISS2[i] = CO('FISS2_%d'%(i))  
PROPA_FISS(MODELE=MODELE[i],  
           FISSURE=( _F(FISS_ACTUELLE=FISS1[i-1],  
                        FISS_PROPAGEE=FISS1[i],  
                        GRILLE_AUX=FGR1, ),  
                    _F(FISS_ACTUELLE=FISS2[i-1],  
                        FISS_PROPAGEE=FISS2[i],  
                        GRILLE_AUX=FGR2, ), ),  
           TEST_MAIL='OUI',  
           DA_MAX=...,  
           METHODE_PROPA='UPWIND',  
           RAYON=RS[i],  
           INFO=1, );
```