The eXtended Finite Element Method

Code_Aster, Salome-Meca course material
GNU FDL licence (http://www.gnu.org/copyleft/fdl.html)
Outline

▶ Motivation

▶ Theoretical aspects
  ▪ Level Sets Method
  ▪ X-FEM enrichments

▶ Application to code_aster
  ▪ Crack definition (Level Sets)
  ▪ Enriched Finite Elements (X-FEM)
  ▪ Post-processing in fracture mechanics
  ▪ Visualisation

▶ Advanced tools
  ▪ Mesh refinement
  ▪ Crack propagation
Outline

Motivation

Theoretical aspects
- Level Sets Method
- X-FEM enrichments

Application to code_aster
- Crack definition (Level Sets)
- Enriched Finite Elements (X-FEM)
- Post-processing in fracture mechanics
- Visualisation

Advanced tools
- Mesh refinement
- Crack propagation
Motivation

Goals
- Study of cracked structures without explicitly meshing crack
- Automatic crack propagation on a single mesh

Finite Element Method (FEM)
- Crack is explicitly meshed
- A long time (human intervention) is needed to mesh complex structures
- Re-meshing is required if changing the crack geometry (parametric study) or position (propagation)

eXtended Finite Element Method (X-FEM)
- Simple mesh (does not respect the crack geometry)
- Unique mesh for entire propagation process
- Simple extension of FEM (relatively suitable for implementation in a EF software)
Outline

Motivation

Theoretical aspects
- Level Sets Method
- X-FEM enrichments

Application to code_aster
- Crack definition (Level Sets)
- Enriched Finite Elements (X-FEM)
- Post-processing in fracture mechanics
- Visualisation

Advanced tools
- Mesh refinement
- Crack propagation
Representation of interface by level set

- Main idea: interface = iso-zero of a “distance function”
- In mechanics: used for modelling interfaces between materials, edges, voids, cracks

![Diagram showing representation of interface by level set](image-url)
Representation of crack by level sets

2 level sets are needed.

- Crack plane ($\text{LSN} = 0$)
- Crack ($\text{LSN} = 0 \cap \text{LST} < 0$)

Iso-values of LST

Iso-zero of LSN
X-FEM enrichment

X-FEM = Two enrichments

- Enrichment with a Heaviside function to represent a discontinuous field (displacement jump) across the interface
- Enrichment with asymptotic functions near the crack tip

NB: The actual formulation is more complex but not detailed here for the sake of simplicity.

Finite elements enriched with
Heaviside function

Finite elements enriched with
asymptotic functions ($\sim \sqrt{r}$)

\[ u(x) = \sum_i a_i N_i(x) + \sum_i b_i N_i(x) H(x) + \sum_i c_i^{1,4} N_i(x) B^{1,4}(r,\theta) \]

\[ B = \left\{ \sqrt{r} \sin \frac{\theta}{2}, \sqrt{r} \cos \frac{\theta}{2}, \sqrt{r} \sin \frac{\theta}{2} \sin \theta, \sqrt{r} \cos \frac{\theta}{2} \sin \theta \right\} \]

where \( r = \sqrt{lsn^2 + lst^2} \), \( \theta = \tan^{-1}\left(\frac{lsn}{lst}\right) \) (signed distance functions)
Outline

- Motivation

- Theoretical aspects
  - Level Sets Method
  - X-FEM enrichments

- Application to *code_aster*
  - Crack definition (Level Sets)
  - Enriched Finite Elements (X-FEM)
  - Post-processing in fracture mechanics
  - Visualisation

- Advanced tools
  - Mesh refinement
  - Crack propagation
Crack Definition (Level sets)

Definition of a crack (or an interface) with the command `DEFI_FISS_XFEM`
- Choice of the mesh
- Choice of a crack or an interface
- Definition of the crack or the interface with level sets
- Definition of the enrichment zone (optional)
- Definition of the type of enrichment (optional)
Crack Definition (Level sets)

Choice of a mesh

FISS=DEFI_FISS_XFEM(MAILLAGE=MAILLA,
    ...
)

- The sane mesh is provided via the keyword MAILLAGE (mandatory)
- Level sets will be computed at all the nodes of this mesh
Crack Definition (Level sets)

**Choice of a crack (fissure in French) or an interface**

```plaintext
FISS=DEFI_FISS_XFEM(MAILLAGE=MAILLA,
        TYPE_DISCONTINUITE = / 'FISSURE', [DEFAULT]
        / 'INTERFACE'
)
```

- **A crack**
- **An interface**

« sane » mesh
Crack Definition (Level sets)

- Choice of a crack (fissure in French) or an interface

\[
\text{FISS} = \text{DEFI\_FISS\_XFEM}(\text{MAILAGE} = \text{MAILLA}, \\
\quad \text{TYPE\_DISCONTINUITE} = /'\text{FISSURE}', \\
\quad /'\text{INTERFACE}' \\
\quad \text{DEFI}\_\text{FISS} = _F() \\
\]

- Level sets defined under the keyword **DEFI\_FISS**
- 4 available methods:
  - Analytical functions
  - Pre-defined geometries
  - Projection onto group of elements
  - Lecture of pre-computed LSN and LST fields

« sane » mesh

Mesh

Crack Definition (level sets)

Finite Element Enrichment

Computation Resolution

Post-processing SIF, propagation

Visualisation

- (only this method is employed in this workshop)
Crack Definition (Level sets)

**Method 1: analytical functions**

\[
\text{LSN} = \text{FORMULE(NOM\_PARA=('X', 'Y', 'Z'), VALE='Z-H')}
\]

\[
\text{LST} = \text{FORMULE(NOM\_PARA=('X', 'Y', 'Z'), VALE='Y-a')}
\]

\[
\text{FISS} = \text{DEFI\_FISS\_XFEM(MAILLAGE=MAILLA,}
\]

\[
\begin{align*}
\text{DEFI\_FISS} & = \_F( \text{FONC\_LN} = \text{LSN}, \\
& \quad \text{FONC\_LT} = \text{LST}, )
\end{align*}
\]

- Rapidity
- Formula of LSN and LST not available for complex crack geometries

Formula of LSN and LST are available only for very simple crack geometries

- « sane » mesh

Mesh → Crack definition (level sets) → Finite Element Enrichment → Computation Resolution → Post-processing SIF, propagation → Visualisation
Crack Definition (Level sets)

Method 2: pre-defined geometries

\[
\text{FORM\_FISS} = \text{SIMP (}
\begin{align*}
\# \text{ type of cracks :} \\
3d : \text{"ELLIPSE", "CYLINDRE", "DEMI\_PLAN" (half-plane), "RECTANGLE"}
\end{align*}
\]

2d : \text{"DEMI\_DROITE", "SEGMENT", "ENTAILLE"}

« sane » mesh

Mesh

Crack definition (level sets)

Finite Element Enrichment

Computation Resolution

Post-processing SIF, propagation

Visualisation
Crack Definition (Level sets)

Method 2: pre-defined geometries

```plaintext
FISS=DEFI_FISS_XFEM(MAILLAGE=MAILLA,
                     DEFI_FISS=_F(
                        FORM_FISS='ELLIPSE',
                        DEMI-GRAND_AXE = 0.4,
                        DEMI-PETIT_AXE = 0.2,
                        CENTRE = (0.5, 0.5, 0.5),
                        VECT_X = (0., 1., 0.2),
                        VECT_Y = (-1., 0., 0.4),
                    ))
```

- Low risk of user error
- Rapidity
- Automatic crack front orientation

- This method is highly recommended
- If a crack geometry is not available in the catalogue, propose a new development to the code_aster team
Crack Definition (Level sets)

Method 3: projection onto groups of elements

\[ \text{FISS} = \text{DEFI\_FISS\_XFEM}(\text{MAILLAGE} = \text{MAILLA}, \]
\[ \quad \text{DEFI\_FISS} = _F( \text{GROUP\_MA\_FISS} = '\text{MA\_FISS}', \]
\[ \quad \text{GROUP\_MA\_FOND} = '\text{MA\_FOND}') , \]
\[ \ldots ) \]

Groups of 2D elements: \textbf{MA\_FISS}

Groups of 1D elements: \textbf{MA\_FOND}

- Capability of building complex cracks
- Time consuming
Crack Definition (Level sets)

Method 4: lecture of a pre-computed LSN and LST fields

**Code**

```
TBLSN=LIRE_TABLE(UNITE=38,
                  FORMAT='ASTER',
                  NUME_TABLE=1,
                  SEPARATEUR=' ',
                  TITRE='level set normale',)

CHLSN=CREA_CHAMP(OPERATION='EXTR', TABLE=TBLSN,
                  TYPE_CHAM='NOEU_NEUT_R', MAILLAGE=MAILLAG1)
...
FISS=DEFI_FISS_XFEM(MAILLAGE=MAILLA,
                    DEFI_FISS=_F( CHAM_NO_LSN=CHLSN,
                                 CHAM_NO_LST=CHLST),
                    ...)```
Crack Definition (Level sets)

Definition of enrichment zone

- Restriction of the crack definition zone with **GROUP_MA_ENRI**

\[
\text{FISS} = \text{DEFI_FISS}_\text{XFEM}(\text{MAILLAGE}=\text{MAILLA}, \\
\text{DEFI_FISS}=\_F(\text{GROUP_MA_FISS}=\text{MA_FISS'}, \\
\text{GROUP_MA_FOND}=\text{MA_FOND'}), \\
\text{GROUP_MA_ENRI} = \text{M_LEFT'}, \\
...,)
\]

- optional keyword

`'M_LEFT',`  
`'M_RIGHT',`

Extension of the crack front

Crack extension  
(un-intentional)

« sane » mesh

Mesh  
Crack definition  
(level sets)  
Finite Element  
Enrichment  
Computation  
Resolution  
Post-processing  
SIF, propagation  
Visualisation
Crack Definition (Level sets)

Definition of the type of enrichment

\[
\text{FISS} = \text{DEFI_FISS_XFEM}(\text{MAILLAGE}=\text{MAILLA},
\begin{align*}
\text{DEFI_FISS} = \_F(\text{GROUP_MA_FISS}=\text{MA_FISS}',
\text{GROUP_MA_FOND}=\text{MA_FOND'}),
\text{GROUP_MA_ENRI} = 'M_VOL',
\text{TYPE_ENRI_FOND} = / '\text{TOPOLOGIQUE}' [\text{DEFAULT}]
/ '\text{GEOMETRIQUE}'
\end{align*}
\]

- optional keyword

- **TOPOLOGIQUE**

- **GEOMETRIQUE**

- **RAYON_ENRI** / **NB_COUCHES**

- « sane » mesh
Enriched Finite Elements (X-FEM)

# Crack definition

\[
\text{FISS} = \text{DEFI_FISS_XFEM}(\text{MAILLAGE} = \text{MAILA}, \\
\ldots \\
\text{DEFI_FISS}=_F(...),)
\]

# Sane model definition

\[
\text{MOD_SAIN} = \text{AFFE_MODELE}(\text{MAILLAGE} = \text{MAILA}, \\
\ldots \\
\text{AFFE} = _F(...),)
\]

- Available modelisations: 3D, C_PLAN, D_PLAN, AXIS
- Linear or Quadratic elements
- The crack and the sane model must be defined on the same mesh

Mesh → Crack definition (level sets) → Finite Element Enrichment → Computation Resolution → Post-processing SIF, propagation → Visualisation
Enriched Finite Elements (X-FEM)

# Crack definition
FISS = DEFI_FISS_XFEM(MAILLAGE = MAILLA,
...  
DEFI_FISS=_F(...),)

# Sane model definition
MOD_SAIN = AFFE_MODELE(MAILLAGE = MAILLA,
...  
AFFE = _F(...),)

# Creation of a model with enriched elements (X-FEM)
MOD_FISS=MODI_MODELE_XFEM(MODELE_IN = MOD_SAIN,
bbie  
FISSURE = FISS)

FISSURE accepts also a list of cracks / interfaces  
MOD_SAIN is the sane model
FISS is the crack

MOD_FISS must be used from now on (loadings, material…)

Mesh

Crack definition
(level sets)

Finite Element Enrichment

Computation
Resolution

Post-processing
SIF, propagation

Visualisation
Resolution

Computation for non-linear studies in quasi-static:

\texttt{STAT\_NON\_LINE, MECA\_STATIQUE, THER\_LINEAIRE}

\texttt{RESU=STAT\_NON\_LINE(\texttt{CHAM\_MATER=CHAMPMAT,}}
\texttt{EXCIT=(_F(\texttt{CHARGE=CH1,)_),),}}
\texttt{MODELE=MOD\_FISS,}}
\texttt{...}}
\texttt{SOLVEUR=_F(METHODE='MUMPS'))}

No new keyword:
The crack is known \textit{via} the model (enrich elements)

if contact on crack faces: MUMPS solver

Mesh

Crack definition (level sets)

Finite Element Enrichment

Computation Resolution

Post-processing SIF, propagation

Visualisation
Resolution

Focus on the sub-element cutting process

- Context: numerical integration of rigidity terms and right-hand side force terms for elements cut by the crack
- On an element cut through by the crack, the integrand might be discontinuous
- Numerical integration with Gauss method on the whole element is not possible
- Solution: sub-element cutting (only for integration purpose)

\[
[K_e] = \int_{\Omega_e} [B] [D] [B] d\Omega_e \quad \text{with} \quad [B] = \left\{ (\phi_i)' (\phi_j)' H (\phi_k)' F^\alpha + \phi_k (F^\alpha)' \right\}, \quad \alpha = 1, 4
\]

\[
[K_e] = \sum_{\text{sub-elements}} \int_{\Omega_{se}} [B]^t [D] [B] d\Omega_{se}
\]
Post-processing in fracture mechanics

Idem for meshed cracks (FEM)

- \texttt{POST\_K1\_K2\_K3}
- \texttt{CALC\_G}

\begin{verbatim}
PK=POST_K1_K2_K3 ( MODELISATION='3D',
MATER=ACIER,
FISSURE = FISS,
RESULTAT = UTOT1,
NB_POINT_FOND = 15,
ABSC_CURV_MAXI = 0.3)

TABFIC=CALC_G (RESULTAT=RESU,
OPTION='CALC_K_G',
THETA=_F(FISSURE=FISS,
R_INF=0.1,
R_SUP=0.3,
NB_POINT_FOND = 15))
\end{verbatim}

Limit the post-processing on few points along the crack front → reduction of the oscillations

Mesh

Crack definition (level sets)

Finite Element Enrichment

Computation Resolution

Post-processing SIF, propagation

Visualisation
Post-processing for visualisation

- Impossible to visualise the crack opening on the computational mesh (sane mesh)
- Need for a visualisation tool
Post-processing for visualisation

- Creation of a mesh for visualisation purpose
- Creation of result fields on this mesh

\[
\text{MA\_VISU} = \text{POST\_MAIL\_XFEM}(\text{MODELE} = \text{MOD\_FISS})
\]

\[
\text{MO\_VISU} = \text{AFFE\_MODELE} (\text{MAILLAGE} = \text{MA\_VISU}, \text{AFFE=_F(...)})
\]

\[
\text{RES\_VISU} = \text{POST\_CHAM\_XFEM} (\text{MODELE\_VISU} = \text{MO\_VISU}, \text{RESULTAT} = \text{RESU})
\]

\[
\text{IMPR\_RESU} (\text{RESU=_F(RESULTAT} = \text{RES\_VISU})
\)

Diagram:
- Mesh
- Crack definition (level sets)
- Finite Elements Enrichment
- Computation Resolution
- Post-processing SIF, propagation
- Visualisation
Outline

- Motivation

- Theoretical aspects
  - Level Sets Method
  - X-FEM enrichments

- Application to code_aster
  - Crack definition (Level Sets)
  - Enriched Finite Elements (X-FEM)
  - Post-processing in fracture mechanics
  - Visualisation

- Advanced tools
  - Mesh refinement
  - Crack propagation
Mesh refinement

- Stress is proportional to $1/\sqrt{r}$, with $r$ the distance to the crack front.
- For both crack representations using FEM or X-FEM, mesh needs to be fine enough for accurate SIF computation.
- Importance of the mesh refinement near the crack front.

![Diagram showing von Mises stresses near crack tip](image)
Mesh refinement

A “sane” mesh needs to be refined before launching the FE analysis (adaptation \textit{a priori})

- Refinement near the crack tip (2D) or near the crack front (3D)
- Automatic operation using the distance to the crack front (computed by level sets)

Only elements nearest to the crack front are refined

Automatic mesh refinement loop

- Initial “sane” mesh
- Level sets
- Distance to the crack-front
- Refinement software (HOMARD)
- Refined “sane” mesh
- X-FE analysis
- SIFs
Mesh refinement in *Code_Aster*

- Refinement parameters (python)

- Level sets computation on the initial mesh (*DEFI_FISS_XFEM*)

- Computation of the refinement criteria (*RAFF_XFEM*)

- Mesh refinement (*MACR_ADAP_MAIL*)

```python
mesh = LIRE_MAILLAGE()

Parameters (python)

crack = DEFI_FISS_XFEM(...)

error = RAFF_XFEM(...)

refmesh = MACR_ADAP_MAIL(...)
```
Mesh refinement in *Code_Aster*

**Refinement parameters**
- $h_0$: mesh size in the area of interest
- $h_C$: mesh size after refinement $\rightarrow h_C \approx a/20$ (a: crack’s depth)

**Calculation of the refinement number**

\[
\text{nb}_\text{raff} = E(n) \quad \text{with} \quad n = \frac{\ln(h_0) - \ln(h_C)}{\ln(2)}
\]

**Real number of iterations for adaption loop**

\[
\text{nb}_\text{iter} = \text{nb}_\text{raff} + 1
\]

**Calculation of real mesh size after refinement(s)**

\[
h = \frac{h_0}{2^{\text{nb}_\text{raff}}}
\]

```python
mesh = LIRE_MAILLAGE()

Parameters (python)

crack = DEFI_FISS_XFEM(...)

error = RAFF_XFEM(...)

refmesh = MACR_ADAP_MAIL(...)
```
Mesh refinement in Code_Aster

Level sets computation on the initial mesh (\texttt{DEFI\_FISS\_XFEM})

\begin{verbatim}
FISS=DEFI\_FISS\_XFEM(MODELE=MO,
  DEFI\_FISS=\_F(FORM\_FISS = 'ELLIPSE',
    DEMI\_GRAND\_AXE = 2.,
    DEMI\_PETIT\_AXE = 2.,
    CENTRE = (0.,0.,0.),
    VECT\_X = (1.,0.,0.),
    VECT\_Y = (0.,1.,0.),),
  GROUP\_MA\_ENRI='CUBE')
\end{verbatim}
Mesh refinement in **Code_Aster**

**Computation of the refinement criteria (RAFF_XFEM)**

```
CH_ERR=RAFF_XFEM(TYPE = / 'DISTANCE' [DEFAULT]
/ 'ZONE', [RECOMMENDED]
FISSURE = FISS,
RAYON = R, [IF TYPE='ZONE']);
```

**Goal: to localize the area near the crack front**

- **Type = ‘DISTANCE’**
  - Nodes field
  - \( r = \text{distance to the crack front} \)
  - Near crack front, error \( \to 0 \)
  - Far crack front, error \( \to -\infty \)

- **Type = ‘ZONE’**
  - Elements field
  - 1 in the refinement area (radius \( R = 5h \) (s.32))
  - 0 elsewhere

- **Parameters (python)**
  - `mesh = LIRE_MAILLAGE()`
  - `crack = DEFI_FISS_XFEM(...)`
  - `error = RAFF_XFEM(...)`
  - `refmesh = MACR_ADAP_MAIL(...)`

2D: disk around the crack type
3D: tore around the crack type
Mesh refinement in **Code_Aster**

- **Refinement : Homard (MACR_ADAP_MAIL)**

  - RAFF_XFEM → Type = ‘DISTANCE’
    
    ```python
    mesh = LIRE_MAILLAGE()
    
    Parameters (python)
    ```

  - RAFF_XFEM → Type = ‘ZONE’
    
    ```python
    crack = DEFI_FISS_XFEM(...)
    
    error = RAFF_XFEM(...)
    
    refmesh = MACR_ADAP_MAIL(...)
    ```

---

**Percentage of elements to be refined**

- All elements with a value greater than 0.5 will be refined

- To avoid refining an element that already reaches the target size

- RAFF_XFEM → Type = ‘DISTANCE’

  ```python
  MACR_ADAP_MAIL(ADAPTATION='RAFFINEMENT',
  CHAM_GD = CH_ERR,
  CRIT_RAFF_PE = 0.1,
  USAGE_CMP = 'RELATIF',
  MAILLAGE_N = MA,
  MAILLAGE_NP1 = CO('MA2'),)
  ```

- RAFF_XFEM → Type = ‘ZONE’

  ```python
  MACR_ADAP_MAIL(ADAPTATION='RAFFINEMENT',
  CHAM_GD = CH_ERR,
  CRIT_RAFF_ABS = 0.5,
  DIAM_MIN = h_C,
  MAILLAGE_N = MA,
  MAILLAGE_NP1 = CO('MA2'),)
  ```
Mesh refinement in Code_Aster

MA=\texttt{LIRE\_MAILLAGE}()
MO=\texttt{AFFE\_MODELE} (\texttt{MAILLAGE}=MA,
     \texttt{AFFE}=\_\texttt{F} (\texttt{TOUT}=\texttt{\textquoteleft\textquoteleft OUI\textquoteleft\textquoteleft}, \texttt{PHENOMENE}=\texttt{\textquoteleft\textquoteleft MECANIQUE\textquoteleft\textquoteleft},
     \texttt{MODELISATION}=\texttt{\textquoteleft\textquoteleft 3D\textquoteleft\textquoteleft}))

\# crack definition
\texttt{FISS} = \texttt{DEFI\_FISS\_XFEM} (\texttt{MODELE}=MO,
     \texttt{DEFI\_FISS}=\_\texttt{F} (\texttt{FORM\_FISS} = \texttt{\textquoteleft\textquoteleft ELLIPSE\textquoteleft\textquoteleft},
     \texttt{DEMI\_GRAND\_AXE} = 2.,
     \texttt{DEMI\_PETIT\_AXE} = 2.,
     \texttt{CENTRE} = (0.,0.,0.),
     \texttt{VECT\_X} = (1.,0.,0.),
     \texttt{VECT\_Y} = (0.,1.,0.),
     \texttt{GROUP\_MA\_ENRI} = \texttt{\textquoteleft\textquoteleft CUBE\textquoteleft\textquoteleft}))

\# error field
\texttt{CH\_ERR} = \texttt{RAFF\_XFEM} (\texttt{FISSURE}=\texttt{FISS})

\# Mesh refinement (call of HOMARD software)
\texttt{MACR\_ADAP\_MAIL} (\texttt{ADAPTATION} = \texttt{\textquoteleft\textquoteleft RAFFINEMENT\textquoteleft\textquoteleft},
     \texttt{CHAM\_GD} = \texttt{CH\_ERR},
     \texttt{CRIT\_RAFF\_PE} = 0.1,
     \texttt{TYPE\_VALEUR\_INDICA} = \texttt{\textquoteleft\textquoteleft V\_RELATIVE\textquoteleft\textquoteleft},
     \texttt{NOM\_CMP\_INDICA} = \texttt{\textquoteleft\textquoteleft X1\textquoteleft\textquoteleft},
     \texttt{MAILLAGE\_N} = MA,
     \texttt{MAILLAGE\_NP1} = \texttt{CO} ('MA2',))

Possibility to use python for a loop

Refined mesh
Examples of mesh refinement
Fatigue Propagation (constant amplitude)

Computation of fatigue propagation: 3 ingredients

- A propagation law (gives the increment of crack advance)
- A bifurcation criterion (gives the bifurcation angle)
- An algorithm for the crack update

Propagation of a 2d crack in a plate under tension with a hole

Common to meshed
and non-meshed

Specific to X-FEM + level sets framework
Fatigue Propagation (constant amplitude)

Ingredient n°1: propagation law

- Paris’ law
  \[
  \frac{da}{dN} = C (\Delta K_{eq})^m
  \]

- C and m: material parameters
- Equivalent stress intensity factor

- Propagation driven by a maximal advance increment: \( \Delta a_{\text{max}} \)

Ingredient n°2: bifurcation criterion

- “Maximum Hoop Stress” (Erdogan, Sih)
  \[
  \beta = 2 \arctan \left[ \frac{1}{4} \left( \frac{K_I}{K_{II}} - \text{sign}(K_{II}) \sqrt{\left( \frac{K_I}{K_{II}} \right)^2 + 8} \right) \right]
  \]

\[
K_{eq} = \sqrt{K_I^2 + K_{II}^2 + \frac{1 + \nu^2}{1 - \nu^2} K_{III}^2}
\]
Fatigue Propagation (constant amplitude)

Ingredient n°3: crack update

- Crack update = level sets update (X-FEM + level set framework)

- 3 methods available in Code_Aster for the level sets update (operator PROPA_FISS)
  - "naïve" approach: MAILLAGE method
  - SIMPLEX method
  - UPWIND method
  - GEOMETRIQUE method
Fatigue Propagation (constant amplitude)

Ingredient n°3: crack update

- "naïve" approach: **MAILLAGE** method

![Diagram of crack update with advance increment and bifurcation angle]

- level sets update = adding a segment/an elementary surface (due to crack advancement) to the crack discretization, then computation of level sets by orthogonal projection of the nodes on this new crack location

- Already in **DEFI_FISS_XFEM** (projection onto groups of elements)

- **UPWIND** method
  - Auxiliary regular grid is required for complex geometry

- **SIMPLEX** method
  - No auxiliary regular grid
  - Robust method

- **GEOMETIQUE** method
  - Faster than the other techniques and capable of modeling 3D non-planar crack growth.

More information
R7.02.12
Fatigue Propagation (constant amplitude)

Propagation of a quarter-disk crack with automatic mesh adaptation
Conclusion on X-FEM in code_aster (V12)

- Behaviour law
  - linear elasticity, non-linear, elasto-plasticity

- Modelisations
  - plane strain, plane stress, axi-symmetrical, 3D

- Loadings
  - pressure on crack faces; displacement boundary conditions near the crack (symmetry conditions); rotation, gravity

- Contact on crack faces

- Finite elements
  - linear or quadratic
Conclusion on X-FEM in code_aster (V12)

- One or multiple crack, crack net, crack branching

- Post-processing in fracture mechanics:
  - CALC_G and POST_K1_K2_K3

- Visualisation
  - displacements and stresses on a “cracked” mesh

- Crack propagation (2d, 3d in-plane and out-of-plane)

- Modal analysis
References

- General user documentation
  - [U2.05.02] how to use X-FEM

- Operator documentation
  - [U4.82.08] Operator `DEFI_FISS_XFEM`
  - [U4.41.11] Operator `MODI_MODELE_XFEM`
  - [U4.82.11] Operator `PROPA_FISS`
  - [U4.82.21] Operator `POST_MAIL_XFEM`
  - [U4.82.22] Operator `POST_CHAM_XFEM`

- General user documentation
  - [R7.02.12] eXtended Finite Element Method

Propagation and crack branching
End of presentation

Is something missing or unclear in this document? Or feeling happy to have read such a clear tutorial?

Please, we welcome any feedbacks about Code_Aster training materials. Do not hesitate to share with us your comments on the Code_Aster forum dedicated thread.
Appendix for TP forma05c

X-FEM

Mesh

Crack definition (level sets)

Finite Element Enrichment

Computation Resolution

Post-processing SIF, propagation

Visualisation

<sane> mesh

DEFI_FISS_XFEM
One by crack

MODI_MODELE_XFEM

IDEM FEM

CALC_G, POST_K1_K2_K3

POST_MAIL_XFEM, POST_CHAM_XFEM