Advanced remeshing techniques for complex 3D crack propagation

Vincent Chiaruttini[1,*], Vincent Riolo[1,2], Frederic Feyel[1]

[1] Onera, DMSM/MNU, Chatillon, France
[2] LMS, Ecole polytechnique, Palaiseau, France
* Corresponding author: vincent.chiaruttini@onera.fr

Abstract This paper is related to the development of an efficient numerical technique to simulate the propagation of 3D cracks that can evolve inside complex industrial structures due to intense fatigue loading. In this prospect, an adaptive remeshing approach is presented: based on an efficient mesh intersection algorithm, and using robust automatic meshers developed at INRIA, this approach allows to insert almost any kind of cracks in very complex meshes. Associated to a fast and accurate stress intensity factor evaluation, a kinking angle criterion and a fatigue propagation law, such remeshing algorithms allow to simulate complex crack growth phenomena (with coalescing multiple cracks, crack fronts splitting, contact in small deformation, etc.) in the context of linear elastic fracture mechanics.

Keywords 3D crack growth, adaptive remeshing, conform crack remeshing, linear elastic fracture mechanics.

1. Introduction

The accurate prediction of crack propagation becomes increasingly necessary in a wide range of industrial applications (aeronautic, automotive industry, civil engineering, etc.). Mostly due to a better optimisation and a higher level of complexity, manufacturing requires ever more sophisticated design techniques and precise damage tolerance analysis in order to correctly evaluate lifetime assessment. Critical parts, such as rotors in aircraft engines, are actively investigated for cracks, using non-destructive means of detection. Such parts, if cracked, are usually replaced immediately for safety reason, However, crack detection technologies have limitations, and some very small initiated cracks might not be detected during inspections. It is thus necessary to find a way to correctly define when the next inspection should occur. Modeling how such small cracks propagate due to fatigue loading is believed to be a solution.

During the last decades, many new approaches have been developed to efficiently deal with complex 3D crack growth simulations under fatigue loading. Two main families of methods have been highly investigated by the computational mechanical community: on the one hand, PUM based methods that allows the presence of a crack inside a discretized component as a possible discontinuities of the unknown fields (like the famous X-FEM/Levelset method [1] with non-conform crack meshing), on the other hand method that describes the crack as evolving boundaries of the discrete domain (boundary elements based methods [2], finite element methods with remeshing [3], etc.).

Concerning the standard finite element approach, recent improvements of meshing and remeshing techniques allow the generation of complex 3D meshes that could be refined in local areas of interest where a crack could propagate. Thus, inserting the discontinuity surface in such meshes was one of the last drawback of those methods. Traditional approaches, usually based on a cracked CSG model, or mesh generation with boolean operations, generally lead to a lack of robustness which make them difficult to deal with complex geometries.

In order to push the lines of remeshing based techniques, a new algorithm as been developed. Our approach is mostly based on a fast and efficient crack insertion algorithm. The robustness and quickness of this method is obtained by the way the cracked mesh is represented (linked to element mesh size near the surface of discontinuity). In fact, each volumetric element of the initial mesh that crosses the crack surface is cut. An important assumption is that the generated surface can be approximated by the polygonals built on the input element edges intersection points. Some special treatments are also performed on element faces to carry out the crack front intersection and some topological difficulties that could arise for highly curved cracks. This resulting mesh is finally rebuilt using an adaptive remeshing strategy in order to eliminate bad quality elements and provide a suitable discretization for accurate finite element solution and stress intensity factors computation.

The next section is related to the description of the crack insertion algorithm that builds a refined mesh closely to the crack front. The third section presents two numerical assessments (a validation problem and a complex crack growth simulation) computed using the software Z-Cracks (within all the herein presented algorithms have been implemented). Finally some conclusion and prospects to this work will be presented.

2. Crack insertion algorithm

Usually, industrial component model for structural integrity analysis are described using constitutive solid geometry (CSG) and efficient automatic meshers are able to deliver correct input for standard finite element solvers. Thus, when cracked structure is concerned, it is necessary to be able to introduce an initial crack geometry in the corresponding model. The current approach is based on the modification of an initial uncracked finite element mesh, that will be modified due to the presence of the crack and refined in the front vicinity.

For a 3D discrete problem, such an algorithm can be separated in five main stages:

- generating an accurate initial crack geometry.
- applying an adaptive refinement operation of the initial structure volumetric mesh.
- performing the “cutting” operation that builds a boundary mesh which contains the crack.
- performing an adaptive refinement operation on the cracked mesh for accurate finite element solution computation.
- boundary regeneration to separate the crack lips.

The first stage is related to the design of a suitable geometry that represents the surface of the crack itself. Such geometry must be discretized and represented by a surface mesh made of linear triangular elements. Such elements must be small enough to represent with a good accuracy the initial crack surface geometry. The most common case is a penny shape initial crack. In this case a radial mesh is performed using a prescribe number of sectors (usually about 64, possibly more for higher values of maximal/minimal radius ratio for elliptic shapes). Then an adaptive remeshing process is always run: using the *Yams* surface remeshing software from Distene/Inria, the element size is reduced to a minimal prescribed value on the boundary of the surface (that corresponds to the crack front) and the edges element size is adapted where important local curvature has been observed, while the mesh quality is optimized.

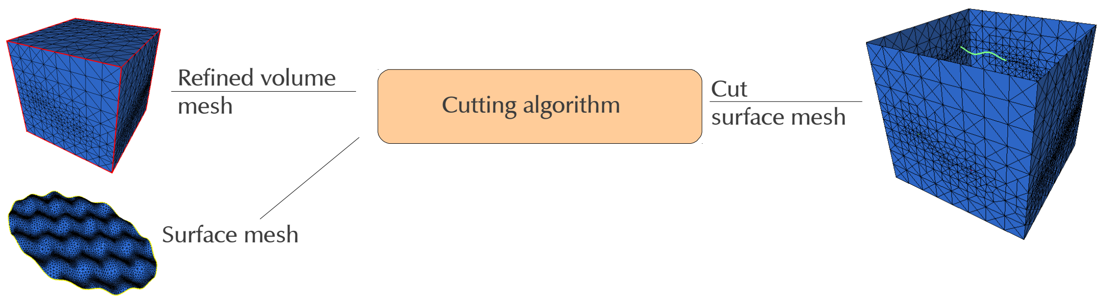

Figure 1. Cutting surface input/output.

The second stage of the a algorithm is related to a volumetric mesh adaption of the uncracked structure closely to the crack geometry. This operation is done using an iterative process. For each point of the actual volumetric mesh, the closest point of the crack surface will be searched (using a binary tree with a *log(n)* complexity for *n* points on the crack surface). Let *d* be the distance between any those two points. Thus an explicit refinement function is applied to specify the required element size on the volumetric mesh:

$$(d) = \begin{cases} h_{min}, \text{if } d < N\,h_{min} \\ \min\left(h_{min}\left(1 + \alpha_r\left(d - N\,h_{min}\right), h\right)\right) \text{ else} \end{cases} \tag{1}$$

where h_{min} is a minimal edge length, N is the number of minimal size element layers that must be imposed closely to the crack surface, α is a refinement factor (usually chosen within 0.1<α<1.) and h is the actual mesh size (meaning that only a refinement operation is done).

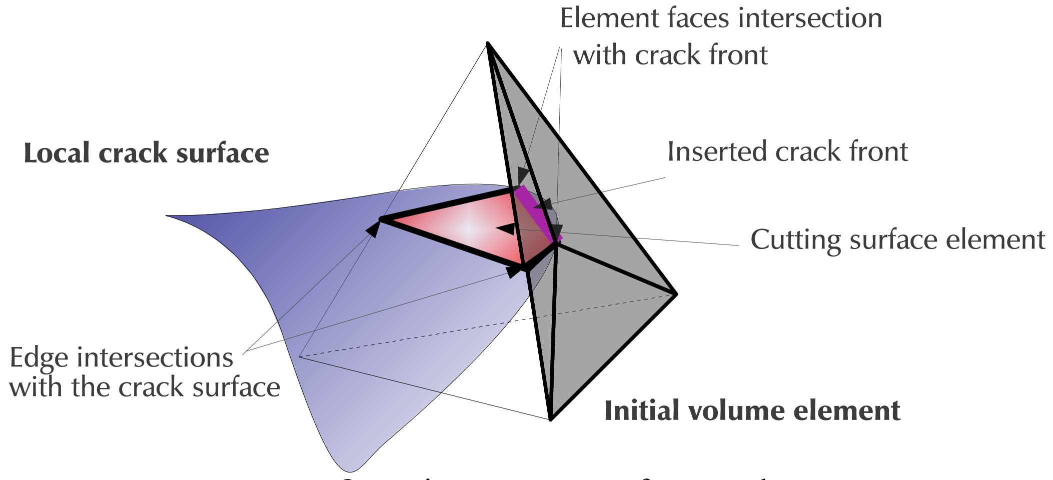
Element faces intersection with crack front
Local crack surface
Inserted crack front
Cutting surface element
Edge intersections with the crack surface
Initial volume element

Figure 2. Operation on a near surface cut element.

The volumetric remeshing operation is then carried out on the complete mesh (or only a subdomain close enough to the crack surface - while the interface with the rest of the mesh must be exactly preserved). This stage is done using the *meshadapt* software from Distene/Inria where the imposed isotropic refinement map is imposed. Another required distance map is finally rebuilt (using the same refinement function) on the output mesh and compared to the actual edges size. If the size maps are close enough, convergence is assumed to be reached, in the other case another remeshing iteration is done.

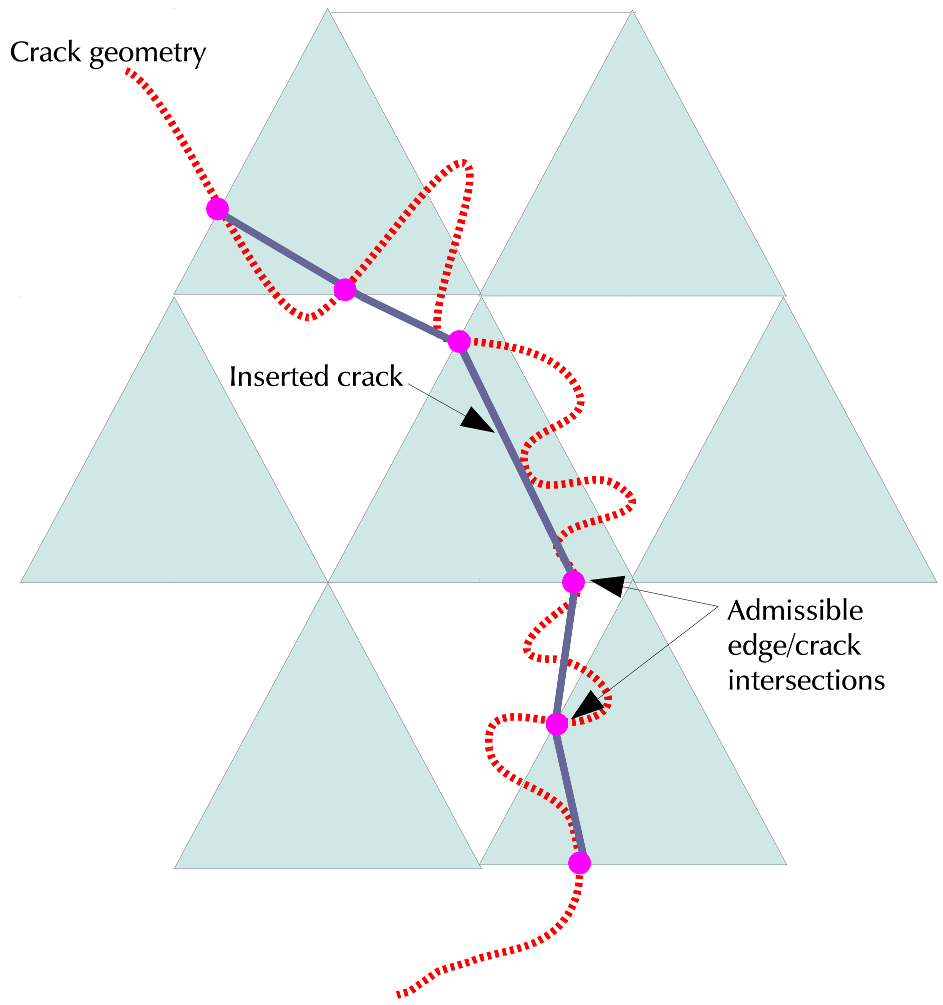

Figure 3. Simplification of a complex 2D crack geometry.

The next stage is related to the "cutting" algorithm, the aim of this stage is to generate a suitable approximation of the subtraction boolean operation between the volumetric mesh and the crack surface mesh (see figure 1). However, on complex meshes, such operation is very difficult to be operated robustly (due to the numerical errors in the numerous required operations). In order to make this operation much more robust, the approach is based on an idea out-coming from the X-FEM/levelset approach: the approximation of the crack surface will be restrained to the fine volumetric mesh topology. The algorithm 1 and the figure 2 give some details about the generation of the cracked surface mesh. In case of a very complex crack surface, an approximation will be obtained with such kind of algorithm (see figure 3), for very accurate cracked mesh generation the initial uncracked mesh must be accordingly refined near the crack surface. Since a convex surface is generated during each element splitting, the method is only valid for linear simplex elements (during the process non-convex elements will be automatically split in simplex sub-elements).

Algorithm 1 Cutting surface algorithm.

```
new_faces ← {}
for $i \leftarrow 0, \cdots, N_{elements}$ do
    $element \leftarrow elements[i]$
    if $intersect\_surface(bounding\_box(element))$ then
        $\{\} \leftarrow inter\_nodes$
        for $j \leftarrow 0, \cdots, N^{element}_{edges}$ do
            $edge\_inter \leftarrow intersect\_surface(element.edge[j])$
            if $edge\_inter.size()\%2 = 1$ then
                $inter\_nodes.add(center(edge\_inter))$
            end if
        end for
        for $j \leftarrow 0, \cdots, N^{element}_{faces}$ do
            $face\_inter \leftarrow intersect\_front(element.face[j])$
            if $face\_inter.size()\%2 = 1$ then
                $inter\_nodes.add(center(face\_inter))$
                if $element.face[j].unshared()$ then
                    $new\_faces.add(rebuid\_cut\_face(element.face[j]))$
                end if
            else
                if $element.face[j].unshared()$ then
                    $new\_faces.add(element.face[j])$
                end if
            end if
        end for
        if $inter\_nodes.size() > 2$ then
            $new\_faces.add(convex\_faces(inter\_nodes))$
        end if
    else
        for $j \leftarrow 0, \cdots, N^{element}_{faces}$ do
            if $element.face[j].unshared()$ then
                $new\_faces.add(element.face[j])$
            end if
        end for
    end if
end for
```

After a cleaning operation (that delete the too bad quality simplex elements), the last adaptive remeshing operation is done using the Distene/Inria remeshing tools, another iterative algorithm is involved in order to converge to a suitable mesh for FE computation and stress intensity factor extraction. The size map will now be build using the distance from any point of the volumetric mesh to the crack front nodes (using the previously defined size function (1)). When convergence is achieved the last stage is related to the boundary regeneration in order to separate both sides of the crack surface. In this respect a global/local coloring algorithm is used. The fast global approach is used on continuous zone without branching nodes while the local approach is required to deal with branching nodes.

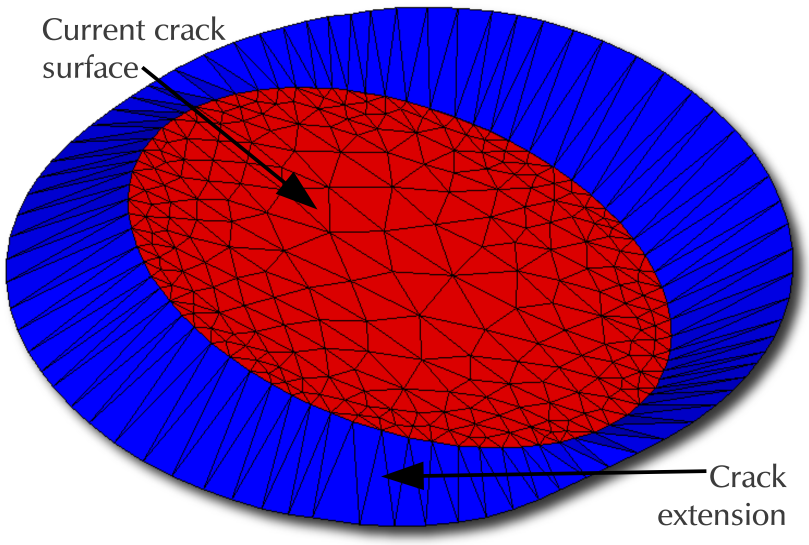

Figure 4. Crack extension during a propagation simulation.

During a crack propagation simulation, such kind of strategy will be adapted as a fewer number of operation is required. Starting from the current cracked mesh, an explicit surface will be generated as the extension of the crack front during a given time step (see figure 4). Thus, the current volumetric mesh, that is already sufficiently refined if the crack advance is small enough, will be directly cut by the crack extension surface. Then only the cleaning operations and a single adaptive remeshing step will be done, before the crack lips regeneration.

3. Numerical assessments

Such a technique for cracked structure meshing has been implemented in our finite element suite Z-Set [4]. A dedicated new module, Z-Cracks, that includes the meshing algorithms, an efficient stress intensity factor extraction post-processing [5,6] has been developed. This tool has been successfully applied to many complex simulations, and two examples will be herein presented: a validation of a crack growth experiment, and the numerical simulation of a very complex crack path will be presented.

3.1. Crack propagation on a biaxial notched specimen

This example is based on the numerical simulation of a mixed mode crack propagation experiment done on a biaxial notched specimen. This specimen has been designed to be able to apply a controlled uniform stress field in its center. The notch is oriented with a 45° angle with the loading directions (see figure 5). Two loading phases are imposed:

- to initiate the crack in the direction of the notch, the same fatigue effort level is imposed on the both directions.
- when the crack has sufficiently advanced, the loading is changed in order to impose an approximate constant vertical stress field in the center part.

The mechanical behavior is assumed to be isotropic linear elastic (E=210 000MPa as Young modulus and ν=0.3 as Poisson coefficient) under small deformations.

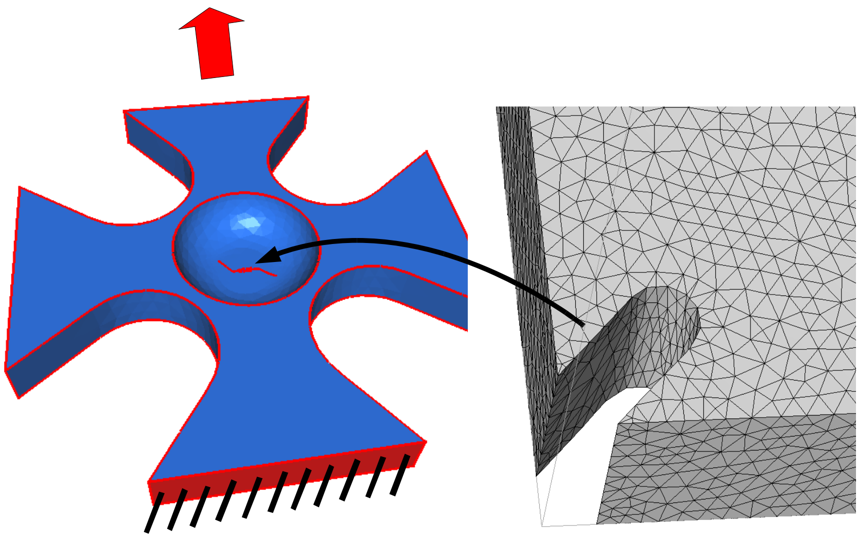

Figure 5. Biaxial notched specimen for crack propagation.

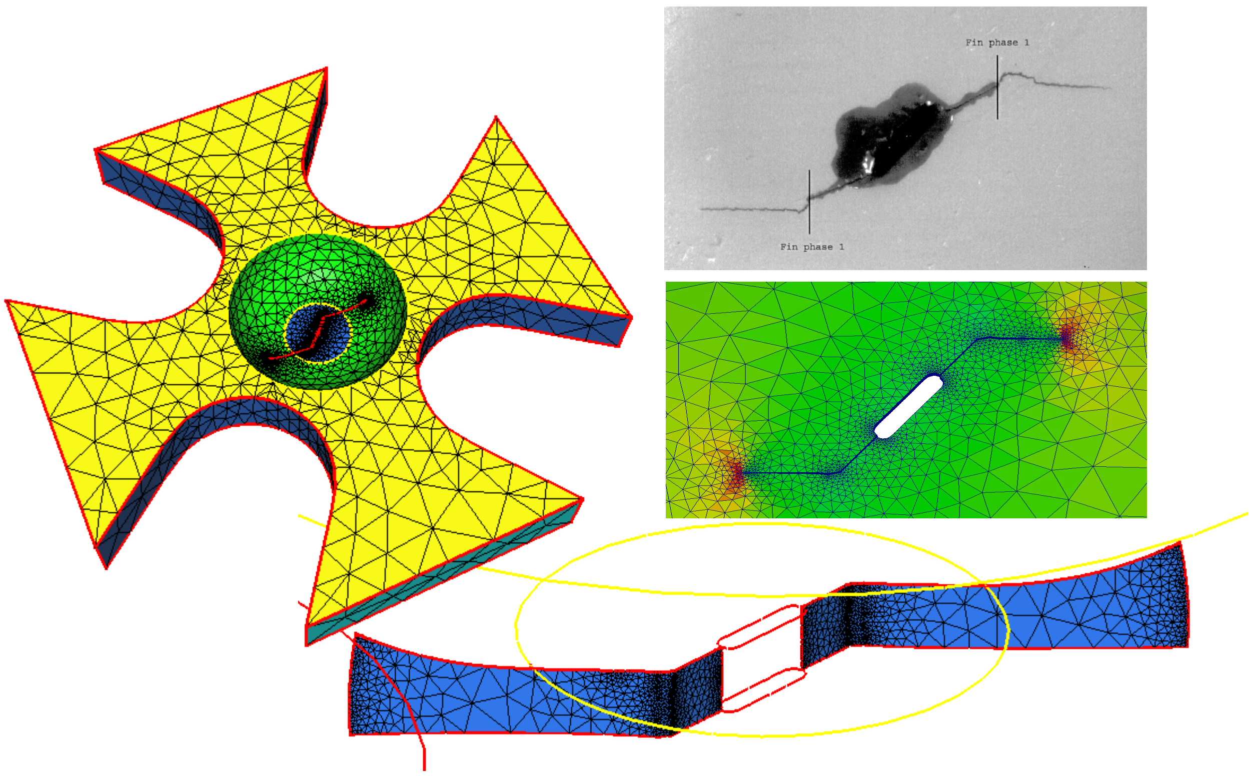

Figure 6. Comparison of crack paths obtained between simulation and experiment.

Both test observation and crack growth simulation show that when the loading is changed, the crack branches. The figure 6 shows the comparison between the experiment and the simulation crack paths.

3.2 Mixed mode crack propagation on a multi-perforated sample structure

This last example highlights the robustness of the method to deal with a very complex crack propagation simulation. The multiperforated sample geometry has been obtained from the Inria geometry database[1], the unit length size and the coordinate system used to describe boundary conditions refer to the downloaded geometry.

The mechanical behavior is assumed to be isotropic linear elastic (E=210 000MPa as Young modulus and ν=0.3 as Poisson coefficient) under small deformations. The boundary conditions are: a prescribed null displacement in all directions on the bottom face (z=-50), vertical displacement Uz=1 and sheer displacement Uy=-2 on the top face (z=50). A fatigue loading is imposed with a simple $[0 \rightarrow 1 \rightarrow 0]$ loading cycle. A Paris law is used with a m=3 exponent on the ΔK equivalent mode I stress intensity factor. The initial crack is a 1. radius disc with a (50.0000 1.79113 -8.14958) center position and a vertical normal direction (0. 0. 1.).

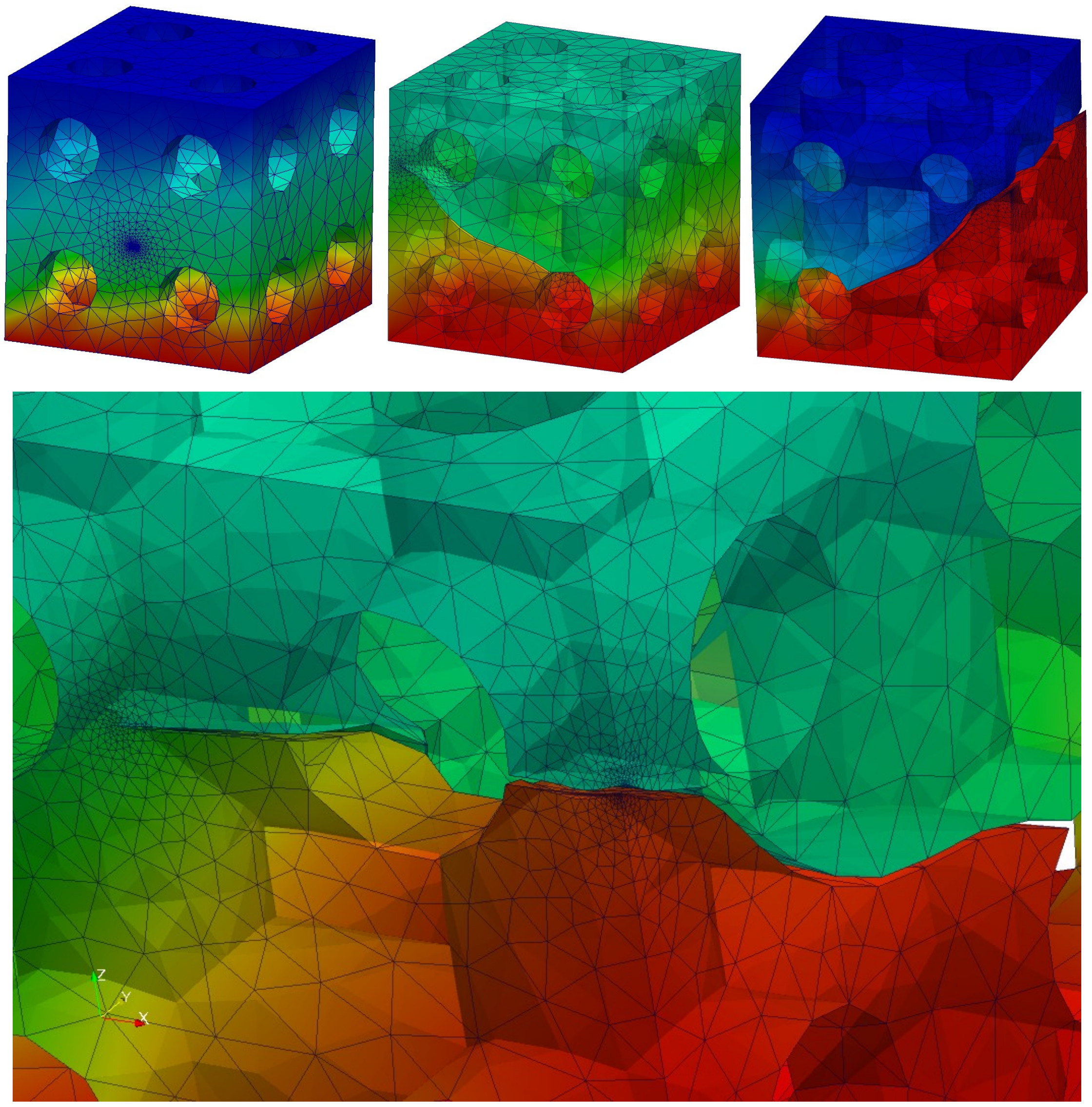

Figure 7. Crack propagation on a multiperforated structure under mixed mode fatigue loading.

1 To be downloaded at:
http://www-roc.inria.fr/gamma/download/affichage.php?dir=DREXEL\&name=drxl_cheese2\&last_page=30

The computation has been performed in 1h26mn on a dual core laptop with 110 remeshing steps. The mesh is built of linear tetrahedron, producing about 20 000 to 35 000 degrees of freedom, depending of the crack geometry complexity during the growth. Material integration, linear solver and stress intensity factors processes are multi-threaded. Some pictures of the cracked mesh are shown on figure 7, an animation can be watched on the web[2] as others example on plates, or 3D structures with coalescing cracks.

4. Conclusion

A robust meshing technique for 3D crack growth simulations has been presented. This method, based on a simplification of the usual boolean operation between meshes, has been implemented in the finite element software Z-set and allows to perform complex crack propagation computations for academic problems (adaptive cohesive zone modeling [7], for instance) as well as for industrial components.

References

[1] N. Moes, J. Dolbow, and T. Belytschko. A finite element method method for crack growth without remeshing. Int. Journal for Num. Meth. In Engrg, 46:131–150, 1999.

[2] B.J. Carter, P.A. Wawrzynek, and A.R. Ingraffea. Automated 3-d crack growth simulation. Int. J. Numer. Methods Engrg., 47:229–253, 2000.

[3] Robert W. Zimmerman Adriana Paluszny. Numerical simulation of multiple 3d fracture propagation using arbitrary meshes. Comput. Methods Appl. Mech. Engrg., 200:953–966, 2011.

[4] J. Besson and R. Foerch. Large scale object-oriented finite element code design. Comp. Meth. in Appl. Mech. and Eng., 42:165–187, 1997.

[5] M. Gosz, J. Dolbow, and B. Moran. Domain integral formulation for stress intensity factor computation along curved three-dimensional interface cracks. International Journal of Solids and Structures, 35(15):1763 – 1783, 1998.

[6] V. Chiaruttini, J. Guilie, V. Riolo, and M. Bonnet. Fast and efficient stress intensity factors computations for 3D cracked structures application to unstructured conform meshes or X-FEM discretization. *in preparation*.

[7] V. Chiaruttini, D. Geoffroy, V. Riolo, and M. Bonnet. An adaptive algorithm for cohesive zone model and arbitrary crack propagation, European Journal of Computational Mechanics, 21:208-218, 2012.

2 http://www.youtube.com/user/OneraMNU/videos