DIRECT MODIFICATION OF SEMANTICALLY-ENRICHED FINITE ELEMENT MESHES

RUDING LOU

Arts et Métiers ParisTech, LSIS-UMR CNR 6186
2, cours des Arts et Métiers, Aix en Provence, France
ruding.lou@ensam.fr

FRANCA GIANNINI
BIANCA FALCIDIENO

CNR IMATI-Ge
Via De Marini 6, 16149 Genova, Italy
{*franca.giannini, bianca.falcidieno*}@ *ge.imati.cnr.it*

JEAN-PHILIPPE PERNOT
PHILIPPE VÉRON

Arts et Métiers ParisTech, LSIS-UMR CNR 6186
2, cours des Arts et Métiers, Aix en Provence, France
{*jean-philippe.pernot, philippe.veron*}@ *ensam.fr*

ALEXEI MIKCHEVITCH
RAPHAËL MARC

Électricite de France Group R&D Direction
1, avenue du General de Gaulle, 92141 Clamart cedex, France
{*alexei.mikchevitch, raphael.marc*}@ *edf.fr*

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Communicated by (xxxxxxxxxx)

Behaviour analysis loop is largely performed on virtual product model before its physical manufacturing. The last avoids high expenses in terms of money and time spent on intermediate manufacturing. It is gainful from the reality to the virtuality but the process could be further optimized especially during the product behaviour optimization phase. This process involves repetition of four main processing steps: CAD design and modification, mesh creation, Finite Element (FE) model generation with the association of physical and geometric data, FE Analysis. The product behaviour analysis loop is performed on the first design solution as well as on the numerous successive product optimization loops. Each design solution evaluation necessitates the same time as required for the first product design that is particularly crucial in the context of maintenance.

In this paper we propose a new framework for CAD-less product optimisation through FE analysis which reduces the model preparation activities traditionally required for FE model creation. More concretely, the idea is to directly operate on the firstly created FE mesh, enriched with physical/geometric semantics, to perform the product modifications required to achieve its optimised version.

In order to accomplish the proposed CAD-less FE analysis framework, modification operators acting on both the mesh geometry and the associated semantics need to be devised. In this paper we discuss the underlying concepts and present possible components for the development of such operators. A high-level operator specification is proposed according to a modular structure that allows an easy realisation of different mesh modification operators. Here, two instances of this high-level operator are described: the planar cracking and the drilling. The realised prototypes validated on industrial FE models show clearly the feasibility of this approach.

Keywords: Finite Element mesh; Groups; Semantics; Mesh deformation; Crack; Hole.

1. Introduction

Numerical behaviour simulation is fundamental in various engineering activities for avoiding physical validation tests. Behaviour simulation is generally used during the product development, maintenance and lifecycle problem analysis of machinery, new solution assessment in order to improve the mechanical behaviour of the machinery. Such activities are frequently subjected to various constraints. For example, in the maintenance/lifecycle problem analysis context, the main constraints are due to the time and cost of the production process stops. Therefore, it is important to be able to provide fast solutions improving production equipment characteristics as well as satisfying the concerned safety criteria. For example, in the field of power production, it is critical to identify the problem source and to provide the appropriate solution in the shortest time. Thus, numerical tools helping experts to rapidly evaluate various alternative solutions should be made available.

Today, most of the product behaviour analyses rely on the following steps: conceptual solution proposal and its modelling using Computer-Aided Design (**CAD**) tools; CAD data meshing; mesh adaptation and enrichment for specific behaviour studies; Finite Element (**FE**) simulation and result evaluation. Depending on the obtained results, geometric modifications can be required, which are generally performed on the CAD models, thus requiring a re-generation of the FE mesh models corresponding to the new solution. Therefore, all the above listed preparation steps necessary for FE Analysis (**FEA**) need to be repeated, as long as the optimal solution is not achieved, every time changes are done on the product shape. In figure 1 all these steps are shown on a real industrial case provided by the Électricite de France Group (**EDF**).

Up to now no real automatic FE model generation process from CAD data is available, and a specific mesh treatment as definition of physical/geometric parameters required for FEA needs a strong user involvement. Examples of physical data, in the following referred as mechanical semantics, are Boundary Conditions or **BC**s, material laws, geometric/mechanical characteristics. To be used in the FE model they are associated to the shape of the model using groups of mesh elements. Mesh groups can be created in two different ways: either **semi-manually** by selecting a set of mesh entities and or **more automatically** by using the corre-

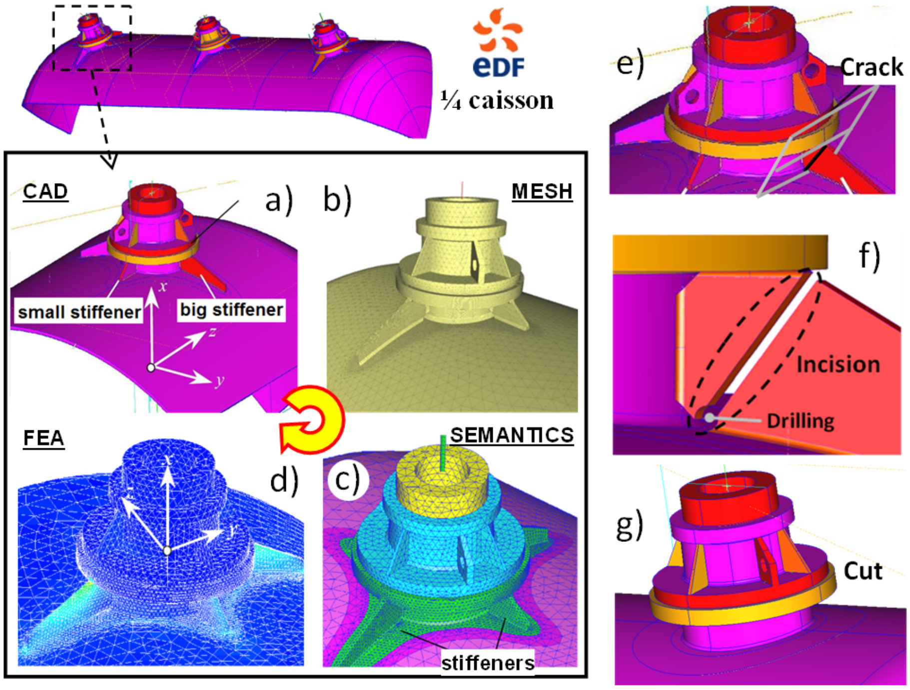

Fig. 1. Mainstream methodology for product behaviour analysis and optimization: (a) caisson CAD modelling, (b) meshing, (c) definition of physical/geometric parameters (semantics enrichment) required for FE model creation, (d) FEA, (e) analysis phase consisting in a crack-feature insertion into the more loaded component, (f) local modification consisting in drilling a hole on the bottom of the stiffener, (g) another modification solution consisting in stiffener removal. After each local modification realised on the CAD model, similar steps from (a) to (d) have to be repeated to assess the modification's effects (courtesy EDF-R&D).

sponding sets of geometric entities of different dimensions preliminarily defined in the CAD model. In both cases, the semantic enrichment of meshes (i.e. mesh group creation and physical data association) requires great skills and is generally quite time-consuming, also considering the possible high number of groups and involved elements. For instance, in the case of the caisson model depicted in figure 1, 30 groups of mesh entities were created to support the needed mechanical semantic data (see fig. 2). Other industrial models of the EDF Group can contain up to 500 mesh groups dedicated to the FEA (e.g. the definition of BCs, link relations, different behaviour laws, geometric parameters, mechanical modelling of specific phenomena) as well as particular post-processing data.

It is then clear that the existing classical methodology for product behaviour analysis is not appropriate in the case of fast industrial studies subjected to important time constraints. It is even less appropriate in all the cases where the numerical behaviour analysis has to be performed on structures which are physically existing

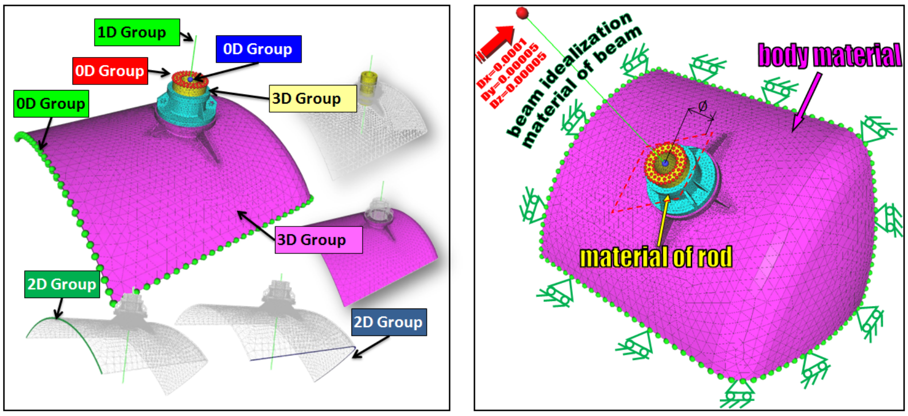

Fig. 2. Mesh groups and associated mechanical semantics defined on the Caisson model (courtesy EDF R&D).

but do not have a CAD model counterpart. This is typical of the maintenance context of real equipments or buildings, whose 3D digital model was never created or is not anymore corresponding to the actual object configuration. In maintenance, the real mechanical/geometrica state of the machinery has to be taken into account for more realistic FE simulation, and the CAD model created during the product development phase is no longer usable for maintenance studies during the product exploitation stage.

Sometimes, data necessary for realistic FE simulation can be obtained though scanning techniques. Thus, the creation of the corresponding CAD models starting from scratch or through Reverse Engineering techniques slows down the process and should then be avoided as much as possible. Additionally, a tuning/validation of the FE model through the comparison of experimental and simulation results is usually necessary. Tuned FE model including mesh and mechanical semantics becomes "frozen" and cannot be modified during the numerical study. Any change of the geometry necessitates a new optimization loop consisting in updating of all geometric/mechanical parameters required for realistic FEA.

Unfortunately, current commercial CAD systems do not make it possible to automate the process of direct complex mesh creation and modification while preserving mechanical/geometric semantic data tuned and required for realistic FEA. Therefore specific attention has to be paid to guarantee that the FE model created after each object shape modification is reflecting all the characteristics of the physically tuned model with a consequent slow down in the FE model preparation.

In order to overcome these limits, we propose a fast prototyping framework working directly at the level of FE mesh enriched by mechanical simulation semantics of geometrical nature (mesh groups). This corresponds to a CAD-less approach allowing the engineer to directly operate on the meshes containing an arbitrary

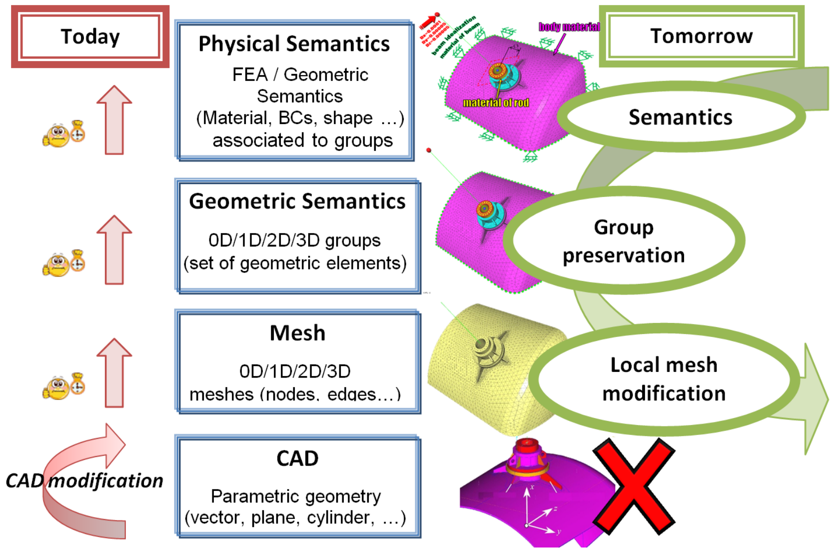

Fig. 3. Workflow for FE model preparation.

number of semantic data already validated with respect to the real behaviour of the structure. The figure 3 shows the comparison between the current classical workflow for FEA model preparation (on the left side) and the proposed workflow (on the right side). With our proposal, engineers can directly modify locally the tuned FE mesh model while preserving existing mesh groups and transferring the already associated semantics required for FEA. We think that this approach is much more suitable than current one both when the perfect CAD model is not available but also in simultaneous engineering approach when an almost complete product definition is achieved and different experts are working in parallel to evaluate and optimise the provided solutions.

The proposed CAD-less approach requires the implementation of shape modification operators able to simultaneously act on the mesh geometry and on the associated semantics. In this paper, we present deformation-based operations applied to FE mesh able to take into account the presence of mesh groups corresponding to geometric semantics. Different constraints are created during the mesh deformation to preserve the shape of the model as well as the shape of the groups of different dimension. The paper is organized as follows. After a discussion of related works (section 2), section 3 describes the modular framework and presents main components of operators allowing manipulating the enriched FE models. section 4 describes two instances of operators: mesh cracking and drilling. section 5 presents some results obtained on industrial models. section 6 draws some conclusions and perspectives of future research.

2. Related works

Mesh or grid modification has been subject to various researches since many years [20] and it has become even more topic of interest recently due the improved computation capabilities and the diffusion of specific applications, such as animation movies, gaming and Virtual Reality. Recently several modelers (IDEAS®, NX®, LMS Virtual.Lab®, SALOME®, ...) working directly on meshes came out. They generally offer functionalities to create shapes and modify mesh models. Because of their context of use demands, they generally do not care neither about the quality of the obtained mesh elements nor about the preservation of the possible associated semantics.

Bremberg and *Dhondt* [9] propose an approach for crack insertion into a mesh by creating a blending between the surface mesh of the crack profile and the mesh of the cracked volume. The skin surface mesh is firstly extracted from the initial volume mesh and merged with the crack profile surface mesh. The resulting surface mesh is then tetrahedralized. The main disadvantage of this approach is that for making crack feature the whole model is re-meshed, which make invalid the tuning on the initial mesh as well as the semantics associated to the mesh.

Numerical crack introduction and propagation schemes were augmented in an elegant manner with a X-FEM (eXtended Finite Element) method [4,5,6]. The use of special enrichment functions as well as a discontinuous function along the sides of the crack allows one to achieve "virtually" a complete crack analysis (on mechanical computation level only using so-called "level sets") without any geometric mesh modification [4]. These works aim at predicting crack behaviour and not to insert a specific crack shape on the mesh without using CAD model. However, the application of this method could be difficult in the case of geometrically complex cracks having no regular shape (because it is difficult to describe such a crack feature).

The insertion of cracks into a mesh model could be performed by splitting of mesh elements [7,8]. The proposed direct split of elements is mainly aimed at real-time visualization of cracking process. Whereas, from the FEA point of view, the resulting mesh is not appropriate because the split elements could have bad quality.

Nienhuys and *al.* [11] describe a cutting algorithm continuously deforming tetrahedra so that the cutting trajectory aligns with the tetrahedron face or edge. This method reduces the need to introduce new nodes but can produce degenerate tetrahedra. Another approach [12] allows multiple consecutive incisions of tetrahedra in the crack zone. In this method, a tetrahedron maintains its state information including the number and position of cuts. Multiple cuts are merged, and the affected tetrahedra are subdivided along the cut plane when a portion of the mesh is completely severed from the rest.

A cutting mesh approach [15,16,17] is proposed to directly split the elements in order to follow the cutting trajectory. However, also in this case the mesh quality is not acceptable for FEA. On the contrary, some other cutting operators [13,14] take care about the mesh quality, but the cutting profile on the mesh does not perfectly match the cutting tool. A method for the simulation of cutting a mesh with a mesh model of the tool is presented [19]. The intersection points between the two interacting meshes are inserted so that the mesh elements are directly split and removed. The cut tool should be meshed and the quality of the resulting mesh is not enough good for the FE simulation.

The use of Boolean intersection and cut operations between the original model and crack masks have been presented [10]. Boolean operations [18] are performed on volume mesh by doing intersection of boundary meshes and completely re-filling the tetrahedral mesh. This is not admissible when manipulating tuned FE meshes that can be only modified locally.

As shown form the above overview, several results have been obtained for mesh modification, all the described work are limited to change only the mesh topology and geometry while more or less preserving its elements quality, but none is taking care of the associated semantic information, Here we present deformation-based cracking and cutting/drilling operations applied to FE mesh able to take into account the presence of mesh groups corresponding to geometric semantics. Different constraints are created during the mesh deformation to preserve the shape of the model as well as the shape of the groups of different dimension.

3. A multi-layered CAD-less modification approach

As previously described, FEA requires various data/parameters of different nature (geometric, physical ...). This information can be classified according to several levels (fig. 4). The geometric data corresponding to the mesh model can be considered as the lowest level of information. This type of data describes the shape of the structure to be simulated: the coordinates of the nodes, the links between the nodes, etc. The accuracy of the physical simulation strongly depends on the type of elements (tetrahedra, hexahedra, etc.) that have been chosen, mesh quality as well as on the number of elements (coarse or fine discretization).

At the highest level, FE models include the various physical semantics (SEM1, SEM2, etc.). For modelling the mechanical behaviour of the structure there might be material properties, BCs, loading description, interaction with a fluid, etc. For example, in order to characterize the material properties, we use Young's modulus, Poisson's ratio, or particular non-linear material laws. Model idealization needs other parameters, for example, a thickness for a thin structure described as a shell. BCs correspond to physical loads of different nature like pressure, concentrated forces, imposed displacements, relationships to simulate a fracture phenomenon or to handle a contact problem, etc. For enriching shape information of the mesh model there might be shape semantics like plane, sphere, cylinder, etc. These semantics will be used to constrain the mesh modification.

To be able to relate physical semantic data required for FEA with the discretized geometry, an intermediate layer is needed (fig. 4). This so-called low-level semantics of purely geometric nature maintains the groups (sets) of FE entities (0D, 1D, 2D

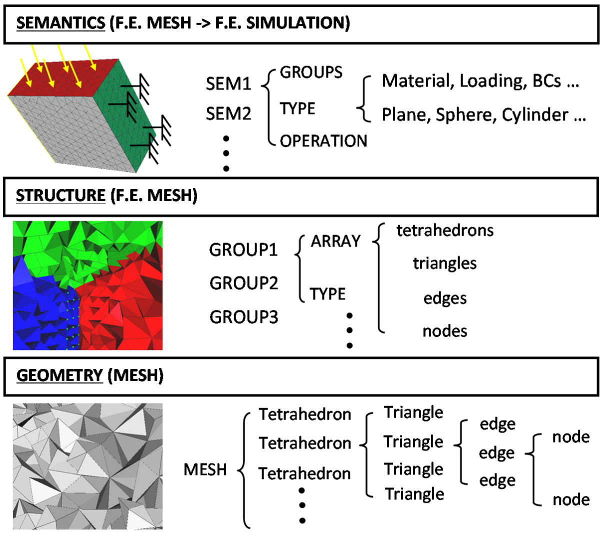

Fig. 4. Three levels of information contained in a FE model.

or 3D elements) and allows associating the physical semantic data to the lower geometric level (e.g. nodes, edges, faces and tetrahedra in the case of tetrahedral mesh model). For example, a group of tetrahedra representing a part of a 3D model can be used to specify a particular material law in order to simulate locally a plastic behaviour of the structure. A group of triangles can be used to apply the pressure to the structure. A group of edges may idealize a beam-like-shape part. A group of nodes can be used to fix a part of the structure or to describe displacement relations of different nature. It is worth to mention that the decomposition provided by the group structure is only dependent on the structure configuration in the real counterpart and therefore traditional mesh decomposition methods based on morphological and structural characteristics cannot be applied.

Therefore, during the direct manipulation of FE mesh models containing different semantic data, the mesh modification operators should correctly maintain and update not only the geometry (mesh) but also all the data associated with a given mesh model. Moreover, the associated semantics constrain the FE mesh modification process. Thus, mesh modification operators should be able to take into account the presented triplet level information. Figure 5 shows the general structure of the proposed CAD-less operator which takes as input the semantics enriched FE mesh and modifies locally the mesh while transferring the semantics. This general operator is constituted by five main components: (1) Group treatment, (2) Topology modification, (3) Mesh deformation, (4) Semantics transfer and (5) Mesh quality control. All these components are detailed in the following sub-sections.

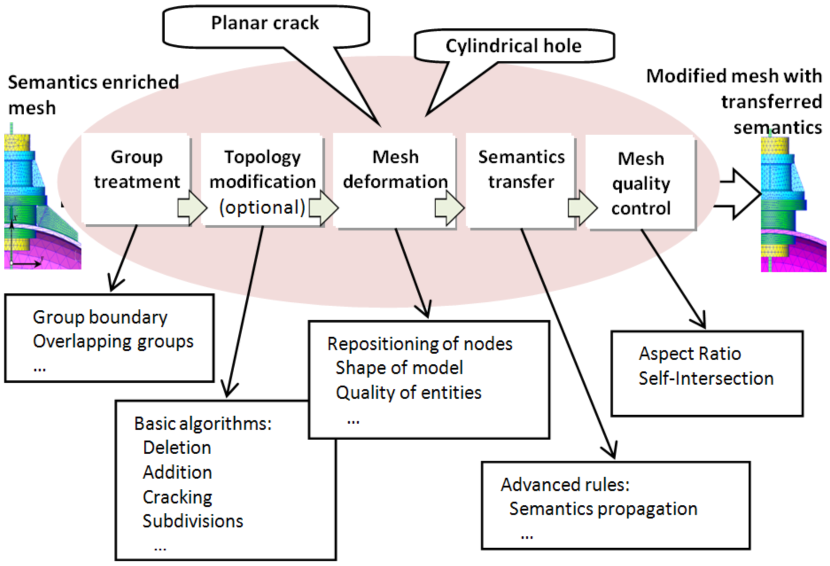

Fig. 5. General structure of CAD-less FE model modification operator

3.1. *Group treatment*

To associate physical semantic information to the FE mesh model we exploit the notion of groups of mesh entities (or FE mesh groups) corresponding to the low-level geometric semantic. FE groups can be used by different FE solvers. For example, the Code_ASTER® FE solver [26], developed and distributed by EDF R&D, uses currently mesh groups in order to simplify advanced FEA. As shown in figure 2, groups, either of same or of different dimension, can be overlapping. Additionally, groups may be constituted by a combination of entities having different dimensions.

Maintaining and propagating semantic information during the mesh modification operation goes through the preservation of FE mesh groups. In particular, this involves an appropriate handling of mesh group content and mesh group geometry, i.e. the shape of the groups' boundaries and, if existing, the geometric curve or surface support. In this work, we adopt the notion of the so-called Virtual Group Boundaries [2] (**VGB**). In a mesh of dimension dm the VGB for a group is made of two sets of mesh elements:

- Bounding Elements (**BE**): a set of mesh elements that 1) belong to the group or are connected to elements of the group; 2) have one dimension lower than the mesh dimension dm; 3) form closed cycles enclosing the areas of dimension dm covered by group elements.
- Isolated Elements (**IE**): a set of group elements that are not contributing to cover an area of dimension dm with other group elements.

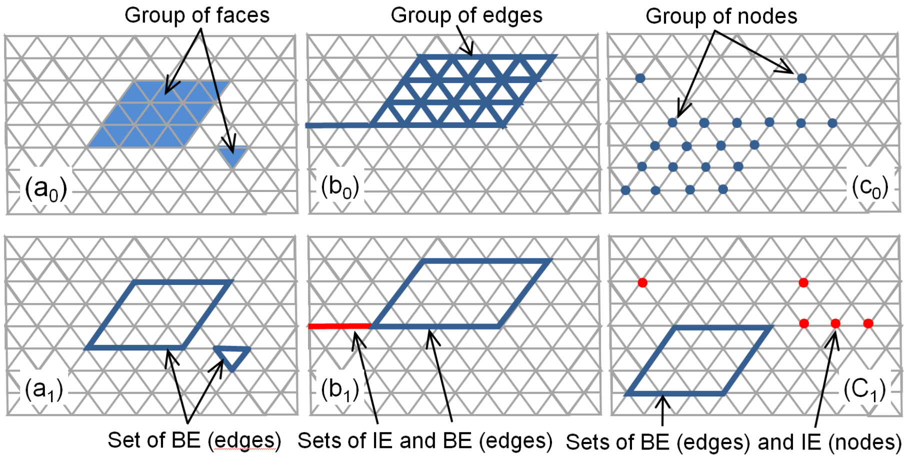

Fig. 6. Examples of VGB computed from groups of different dimension in a 2D mesh.

The figures (6.a to 6.c) show three examples of groups of different dimension in a 2D mesh. In a 2D mesh the area bounded by the BE elements will have dimension 2. The figures (4.d to 4.f) show the BE (blue edges) and the IE (red edges or red nodes) for the relative groups shown in the upper pictures.

Nodes located on VGB are used to apply additional constraints during the local mesh modification to make these nodes staying on the shape of the group boundary. Once the group boundary is handled the group content is also considered. Therefore if the elements of the group are lying on a specific surface or curve geometry such characteristic should be preserved and thus this information should be used for constraining some mesh elements (those that will form the corresponding group) in the modified mesh. Finally, if in the mesh modification process some mesh areas are re-meshed by removing and adding mesh elements the re-assignment of group definition on the newly added mesh elements is necessary. A possible approach is to make a link between the VGB elements and their concerned group so that the group definition of the bounded area can be found via the closest VGB. In case of overlapping groups the re-assignment of group definition on the re-meshed zone will be fuzzy because each area may be bounded by several sub-sets of VGB from different groups. So we introduce the notion of Elementary Group [2] (**EG**).

The elementary groups collect all those mesh elements that are belonging to the same group sets. They are computed by intersecting the areas covered by the two operated groups. The process can be divided into three steps: area computation, area splitting by intersection calculation and new groups' generation according to the split areas. The figure 7 shows a simple example of EGs computation from a group of faces and a group of nodes in a 2D mesh. The figure 7.a shows the initial two groups and the figure 7.b shows their covered areas (domains) composed

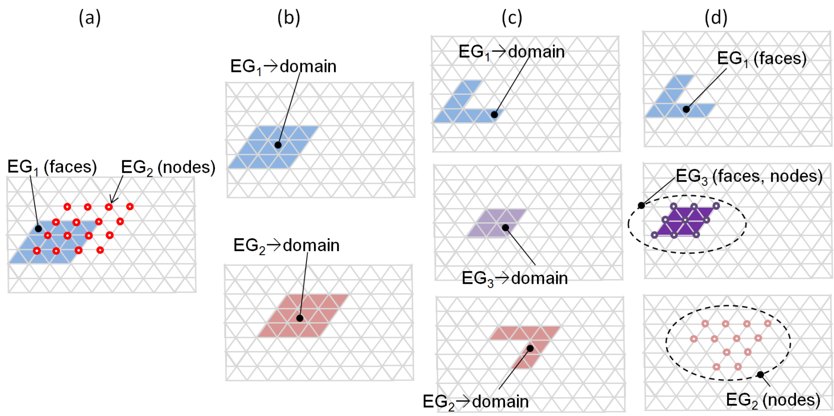

Fig. 7. Examples of EGs computed from two groups of different dimension in 2D mesh.

by faces. The figure 7.c shows the split of the two areas (domains) for avoiding intersection. The figure 7.d shows the produced elementary groups covering the split areas.

3.2. *Topology modification*

Topology modification is related to mesh elements removing, addition, cracking, subdivision, etc. This topological modification is an optional operation depending on the different instance of the general operator. The figure 8 shows these operations on a triangular mesh example. The initial mesh is shown in figure 8.a, while figure 8.b presents the mesh resulting from a deletion operation removing some triangles from the initial mesh. In figure 8.c two separated parts of a mesh are glued by mesh element addition operation. In figure 8.c the mesh is cracked into two separated parts by duplicating a set of nodes and edges on the mesh. Finally, the figure 8.e shows the result of mesh refinement applied on the mesh shown in the figure 8.d. Depending on the specific mesh operator none of several of the above topological modifications may be used.

Additionally the topology modification in our CAD-less mesh operator is constrained by the mesh group definition. For example the operator of re-meshing which needs removing and filling mesh elements should take care of the group shape given by the VGB and the containing elements. Another example is mesh refinement operator which needs to insert new nodes and swaps edges/faces in 2D/3D mesh. The refinement without any constraint could modify the shape of the group boundary. *Lou* and *al.* [3] have shown how to preserve the VGB and group characteristics while accomplish a re-meshing process and a refinement for merging two triangle meshes. The VGB is preserved during the deletion before triangulation and the group definition is re-assigned after creation of new mesh elements.

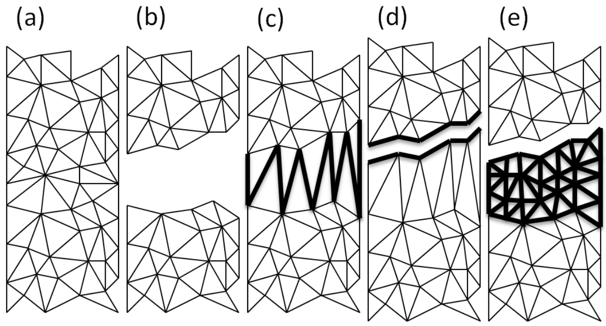

Fig. 8. Different topological modification examples: b) removal, c) addition, d) cracking and e) subdivision.

3.3. *Mesh deformationn*

The deformation engine considered in our general mesh operator provides a repositioning of nodes of a mesh during the mesh modification operation. It is applied to:

- Maximize the global mesh quality,
- Produce and/or preserve certain shapes.

The deformation engine we adopt is based on the so-called Force Density Method [1] (**FDM**). It allows specifying linear and non-linear position constraints required for achieving a certain shape. It uses the minimization of a quadratic objective function based on a simple mechanical spring/bar-like model and provides the final solution in term of local deformation. Indeed, a bar network is created and coupled to the edges of the 2D/3D mesh.

The optimization problem to be solved corresponds to equation 1. The unknowns are the positions of the free nodes to relocate. They are gathered together in the unknown vector X. The position of blocked nodes is known in the optimization system. The objective function to minimize is ϕ and the equality constraints are in the constraint vector G. They are both linking to the unknown vector.

$$\begin{cases} G(X) = 0 \\ \min \phi(\mathrm{X}) \end{cases} \tag{1}$$

In the Force Density Method adopted each bar applied on each mesh edge can be seen as a spring with a null initial length and a stiffness q_i (more precisely a force density). To preserve the static equilibrium state of bars of length ℓ_i, f_i external forces have to be applied to the endpoints of the bar: $f_i = q_i \cdot .\ell_i$. The set of external forces applied to the initial bar network can be obtained through the static equilibrium equations at each node. At the end, we obtain a set of linear equations between the nodes positions X and the external forces applied to them. Let F be the vector containing the components of the external forces applied to the nodes free to move, there exists a linear mapping function g between X and F (equation 2).

$$X = g(F) \tag{2}$$

With the formulation given by equations (1) and (2) we can specify separately two aspects:

- the **geometric constraints** G that may be imposed to meshes such as position or specific shape (e.g. plane). These constraints produce a set of possibly non-linear equations liking directly the position of the free vertices. The resulting constraint vector G can then be expressed as function of the external forces F applied to the free nodes using equation 2.
- the **objective function** ϕ to minimize. This is a higher-level parameter enabling the specification of various deformation behaviours through the combination of several geometric and/or mechanical quantities relative to the bar network [1]. For example the minimization of external forces tends to minimize the surface area and enable an interesting repositioning of the mesh vertices. At the opposite, the minimization of the external forces variations tends to preserve the shape during the deformation.

Some geometric constraints for specific primitive surface type constraints are shown here. Let P_m be a mesh node of coordinates (x_m, y_m, z_m), P_0 be a 3D point of coordinates (x_0, y_0, z_0) and n_0 be a unit normal vector of components (n_{x0}, n_{y0}, n_{z0}), the following constraints can be defined on P_m:

- **planar constraint** so that P_m has to stay on a plane defined by the point P_0 and the normal n_0:

$$G_{p_{m0}}(x_m, y_m, z_m) = (P_m - P_0)n_0 = 0 \tag{3}$$

- **spherical constraint** defined with a sphere centered in P0 and with a radius R:

$$G_{s_{m0}}(x_m, y_m, z_m) = \|P_m - P_0\|^2 - R^2 = 0 \tag{4}$$

- **cylindrical constraint** defined by a unit vector n_0 characterizing its axis and a point P_0 that the mesh node P_m has to match:

$$G_{c_{m0}}(x_m, y_m, z_m) = \|(P_m - P_0) \wedge n_0\|^2 - R^2 = 0 \tag{5}$$

- **free-form constraint** when the identified shape does not correspond to any of the previously introduced constraints. This is done while constraining at iteration k the node P_m to move according to the plane defined by the position of P_m at iteration $k-1$ and the normal n_m to the mesh at this point:

$$G_{fm}^{[k]}(x_m, y_m, z_m) = (P_m^{[k]} - P_m^{[k-1]})n_m^{[k-1]} = 0 \tag{6}$$

A very simple example of mesh adjustment using constrained deformation is shown in the figure 9. At the left part different boundary conditions are defined on all nodes of the initial mesh. These nodes are labeled in different ways detailed under

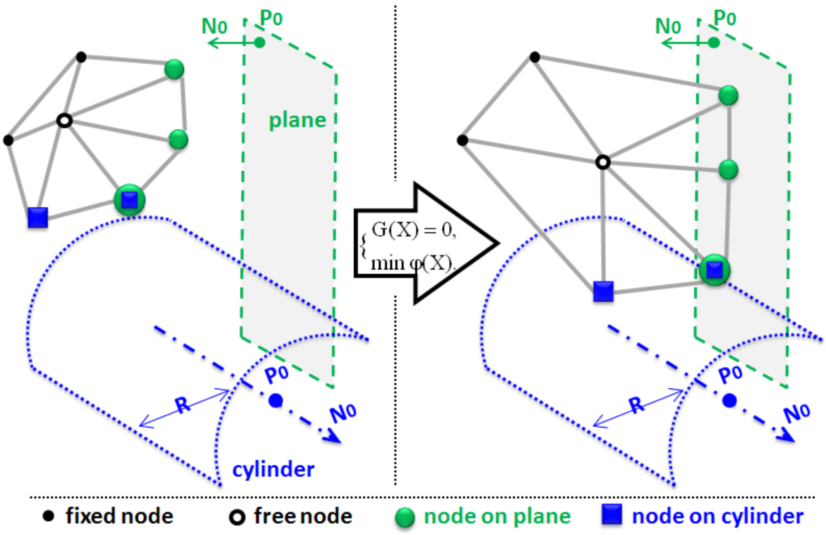

Fig. 9. Example of deformation using force minimization under plane and cylinder type constraint.

the pictures. The "fixed nodes" do not move during the deformation. The "free nodes" can move without any constraint in the space. The "nodes on plane" are free to move only on the plane and the "nodes on cylinder" are free to move only on the cylinder. The right picture gives the result of the deformation by applying external forces minimization under the above shape constraints. The various constraints are satisfied so that some constrained nodes stay either on the plane or on the cylinder or both. The node under two constraints is staying on the intersection zone between the two surfaces. The free node is situating at a place for relaxing the mesh. The two fixed nodes remain at their initial position.

3.4. *Semantics transfer*

During the mesh modification the semantics should not only be maintained through the preservation of the groups but they should also be transferred. The semantics transfer is mandatory when a topology modification is performed, notably when the characteristics of the surface evolve. To illustrate this aspect, we consider the mesh drilling operator. The figure 10.a shows a cube-like model A on which a fluid pressure is applied on the four faces. We want to make a cylindrical hole going through the model. Figure 10.b shows half of the model A in which we can see the half hole realized. Since the pressure is produced by the fluid and the two sides of the hole (top and bottom) receive the pressure, it is reasonable to suggest the user to automatically propagate the pressure to the interior cylinder wall of the hole thus saving manipulation times (fig. 10.b).

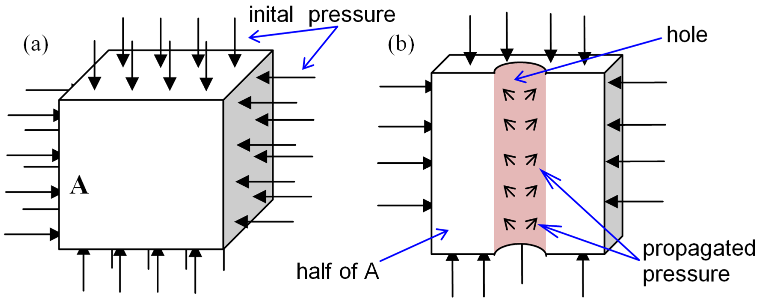

Fig. 10. Fluid pressure propagation in case of mesh drilling.

3.5. *Mesh quality control*

The mesh quality control is very important for FEA. The quality could concern different criteria. The first quality criterion would be the mesh elements ratio aspect. For triangle mesh a degenerated triangle is characterized by exactly one or two small angles. To quantify the degeneracy without computing the three angles, we use an indicator introduced to check the quality of FE meshes [21] and it is defined as follows:

$$Q = \alpha \frac{S}{hp} \tag{7}$$

where Q is the aspect ratio of a triangle with $\alpha = 2\sqrt{3}$ a normalization coefficient so that $Q = 1$ for an equilateral triangle, h is the longest edge length, S is the area of the triangle and p its half-perimeter. This quality factor belongs to the interval $[0, 1]$. The limit zero corresponds to flat triangles. It is commonly accepted that a triangulation is a good one, with respect to the FE approximation, if the aspect ratio of the worst triangle is greater than 0.5. Similarly for a tetrahedral mesh the definition of the quality of a tetrahedron [22] could be computed:

$$Q = \alpha \frac{\rho}{h_{max}} \tag{8}$$

where h_{max} is the diameter of tetrahedron (length of the longest edge in the tetrahedron), ρ is the radius of the inscribed sphere and $\alpha = 2\sqrt{6}$ is a normalization factor such that the quality of an equilateral tetrahedron is 1. This quality function varies in the interval $(0, 1]$ (a well-shaped tetrahedron has a quality close to 1, while an ill-shaped element has a quality close to 0).

The second mesh quality criterion is about self-intersection problem where a part of a surface mesh collides with another part of itself, i.e., two mesh elements intersect each other. It destroys the integrity of the mesh and makes the mesh unusable for FEA applications. In triangle or tetrahedral meshes the check of the intersection between edges and triangles is necessary.

According to different application interest the mesh quality criteria could be completed.

4. CAD-less operator instances

In this section, two instances of the general CAD-less operator are detailed. The first one is planar crack that generates a planar separation of the mesh in correspondence of the crack surface. The second operator is the drilling one that gives rise to cylindrical holes. These two high-level operators are presented for showing how the different basic tools presented in the previous sections can be arranged to achieve the wished FE mesh modifications.

4.1. *Planar crack operator*

The purpose of our operators is to directly modify the semantically enriched FE mesh shape and not to define operators that somehow simulate the physical phenomenon: In this perspective, crack operator is aimed at introducing a "geometric" fissure inside the volume of a structure that results in having two coincident internal surfaces corresponding to the crack. Actually, cracks are usually idealized as having no volume. The surfaces representing the two sides are distinct but coincident so that nodes on opposite sides of crack faces should have identical coordinates. So, we can speak about duplicated overlapping mesh entities (double nodes and face elements). Here the planar crack operator is presented. This crack is supposed closed at the initial instant t_0=0.

It can be mentioned that the duplication of nodes and face elements into a volume corresponds geometrically to the introduction of what we call "contact zone". Thus, the crack operator discussed in this section can be used to handle the contact problem in the case of planar surfaces in contact.

Our approach is mainly based on three steps: crack interface identification (section 4.1.1), mesh deformation on the level of crack interface (section 4.1.2) and duplication of nodes and faces (section 4.1.3). The crack interface is a set of triangles in case of 3D mesh or a set of edges for 2D mesh. It corresponds to the boundary of the tetrahedral/triangles lying completely in one side with respect to the crack plane and that have to be deformed to respect this plane. The nodes relative to the interface elements are forced to move onto the crack plane. To avoid degenerated triangles/tetrahedra, some elements not directly in contact with the crack plane are also moved. The deformation process is based on the one presented in section 3.3. Finally, the crack interface is created by duplicating nodes, edges and/or faces, depending on the mesh dimension, on the two sides of crack.

These steps are detailed in the following subsections. In the example the planar crack operation is delimited by a circle. An example is shown to better understand (fig. 11). It consists of a tetrahedral mesh on which three groups of tetrahedra are defined (fig. 11.a and .b). The figures (11.d and 11.e) show the initial mesh on which different important nodes are identified. The figures (11.f and 11.g) show the deformed mesh for planar cracking.

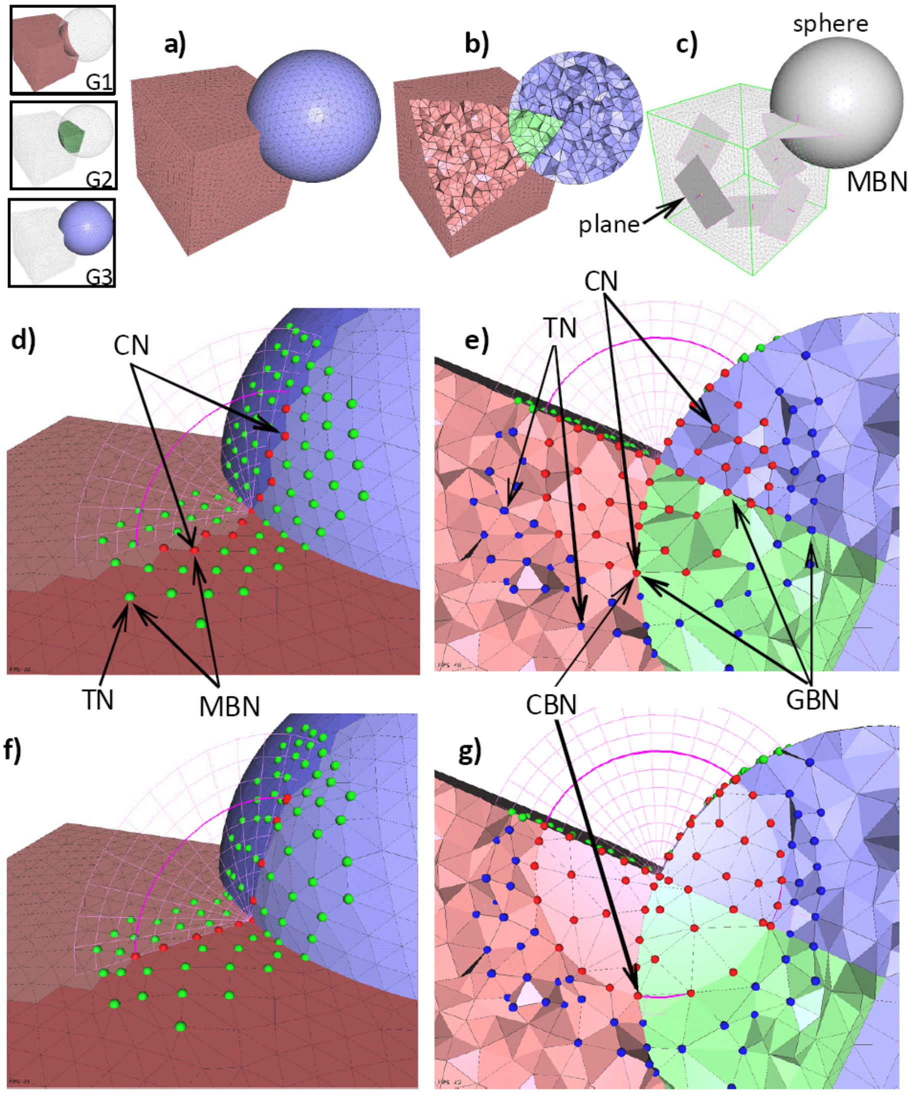

Fig. 11. Insertion of a planar crack (d-g) into the 3D mesh containing three groups highlighted in different colors (a) (b) and delimited by planar and spherical surfaces (c).

4.1.1. *Crack Interface identification*

This first step separates all the mesh nodes into two sets (**N1**, **N2**) according to their position with respect to the two half-spaces (**P**,**N**) defined by the crack plane.

In the case of 3D mesh, a set **T1** gathers together the tetrahedra having their 4 nodes belonging to the half-space P, respectively **T2** for tetrahedra having at least one node in the half-space N. Analogously, for 2D mesh, the set T1 gathers together triangles whose 3 nodes belong to the half-space P, and respectively T2 for triangles defined by at least one node belonging to the half-space N. The figure 12.a shows a 3D mesh and the identified sets T1 and T2 when applying a planar crack. Now, the Crack Interface (**CI**) has to be identified. For 3D mesh, the CI is defined as a set of triangles f which are shared by one tetrahedron t_i in T1 and one tetrahedron t_j in T2. Analogously, in the case of 2D mesh, the CI is a set of edges e which are shared by one triangle f_i in T1 and one triangle f_j in T2. Before computing CI, one of the two sets, let us suppose T1, is chosen and processed such that it contains only tetrahedra (resp. triangles) sharing at maximum one triangle (resp. edge) with tetrahedra (resp. triangles) of other set T2. The set of these triangles/edges will constitute the CI. Actually, tetrahedra which have 2 or 3 shared triangles (resp. triangles with 2 shared edges) should not take part in the definition of the CI since they will be flattened after the deformation as all CI triangles (resp. edges) will be moved onto a plane (resp. a intersection line between the crack plane and triangle plane).

The processing of the set is performed as follows:

- in the case of 3D mesh, if a tetrahedron in T1 has two triangles shared with elements in T2, then the edge shared by the other two triangles should be split so that the tetrahedron is subdivided into two sub-tetrahedra which contain one of the two problematic triangles each. Figure 12.a shows the tetrahedron *abcd* associated with two problematic triangles *acb* and *adc* and the dihedral angle θ that would become 180° after deformation. Here, the edge $b - d$ is split in two by inserting a new node o (fig. 12.b). The tetrahedron is then subdivided into two tetrahedra *abco* and *acdo* and each of them has one of the two interface triangles. All neighbor tetrahedra associated to the split edge $b - d$ should be also split. Figure 12.c shows all the neighbor tetrahedra in the original mesh and the figure 12.d shows them split,
- in the case of 3D meshes, if a tetrahedron in T1 contains three triangles shared with tetrahedra in T2, this tetrahedron is moved from T1 to T2. In this way the fourth triangle becomes an interface triangle. Analogously, in the case of 2D meshes, if a triangle in T1 has two edges shared with triangles in T2, it should be moved from T1 to T2. In this way the third edge becomes an interface edge.

4.1.2. *Crack Interface deformation*

Being the elements of the CI defined, the deformation process described in section 3.3 is applied so that the elements of the CI match the shape of the desired crack. It requires the identification of sets of elements to be moved in a different manner, i.e. using different constraints for each set. We define a set **CN** (Crack Nodes) constituted by all the nodes associated with the elements in CI. Figures (11.d and 11.e) show the identified CN nodes in a 3D mesh.

To enable a smooth transition between the CI elements and the surroundings,

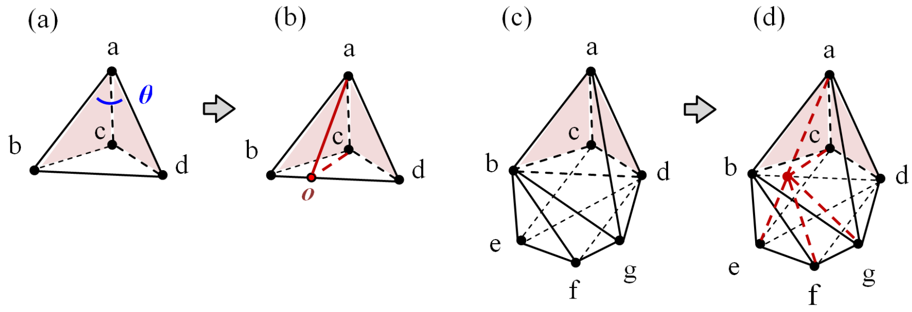

Fig. 12. Splitting schema for the tetrahedron with 2 potential interface triangles.

a set of **TN** (Transition Nodes) is defined. It contains the nodes located in the i^{th} neighborhood of the ones in CN. The bandwidth "i" can be user-defined, or it can be computed by dividing the biggest distance between one crack node and the crack plane by the mean edge length. The bigger it is, the smoother the transition will be, and the better the quality of the mesh will be. Figures (11.d and 11.e) show some TN nodes identified with $i = 2$.

For a tetrahedral mesh, all nodes belonging to only triangles shared by exactly two tetrahedra in the set are internal, and all nodes associating with at least one triangle which is shared by exactly one tetrahedron are external. For a triangular mesh, all nodes belonging to only edges shared by exactly two triangles are internal, and all nodes associating to at least one edge which is shared by exactly one triangle are external. The external nodes may form a characteristic shape of the model. For identifying such characteristic shape, we have devised a tool [23] partially based on [24] for the detection of surface primitives such as plane, sphere, cylinder etc in 2D/3D meshes. For the mesh shown in figure 11 different surface primitives (planar and spherical) are identified for the external surface (fig. 11.c). For preserving the shape of the original model we define also a set of **MBN** (Model Boundary Nodes). This set contains the CN and TN nodes on the mesh boundary. Some MBN nodes are shown in figure 11.d.

If we do not want the crack operation influence the entire model we need to define a delimitation shape. The shape could be a rectangle, circle etc so that the CN nodes are delimited as well as the corresponding TN. The MBN set needs to be updated too. We define the set of **CBN** (Crack Boundary Nodes) containing the CN nodes associating with delimited nodes and kept nodes on the crack plane. In the example of figure 11.e circular delimitation is applied for the planar crack operation, and some CBN nodes are shown.

For preserving the group information the constraint from the virtual group boundary shape is also imposed during the deformation. A set of **GBN** (Group Boundary Nodes) is defined for all free nodes on the crack interface and in the transition zone.

Once the various sets of nodes are identified, geometric constraints can be assigned to drive the deformation process. In the case of 3D mesh, nodes in:

- CN are constrained to stay on the crack plane;
- CN are constrained to stay on the crack plane;
- CBN are constrained to stay on the delimitation shape;
- MBN are constrained to stay on the external skin of the 3D mesh by preserving its surface shape;
- GBN are constrained to stay on the shape of corresponding group boundary;
- TN could move;

and all the other nodes are fixed. For 2D mesh, nodes in:

- CN are constrained to stay on the crack plane and on the mesh by preserving its surface shape;
- CBN are constrained to stay on the delimitation shape;
- MBN are constrained to stay on the mesh boundary by preserving its curve shape;
- GBN are constrained to stay on the shape of corresponding group boundary;
- TN are constrained to stay on the mesh by preserving its surface shape;

and all the other nodes are fixed.

In case of complex free form shapes, we use a tangent plane to the node to constrain locally the deformation.

During the deformation of the mesh, the deformation engine solves under-constrained set of equations based on the mechanical model described in section 3.3. Additionally, a mesh quality criterion is also used in order to guarantee the quality of deformed mesh from the mechanical FEA point of view.

4.1.3. *Definitive crack insertion*

In order to complete the insertion of the crack, the mesh entities belonging to the CI are duplicated to make the required topological modification and the incident relations with edges, triangles and tetrahedra are accordingly updated. Figures (11.f and 11.g) show separately the exterior and the interior of the 3D mesh after insertion of the limited crack and local mesh deformation in the crack zone.

4.2. *Through Hole drilling operator*

The second operator here described is the drilling operator. Similarly to the crack operation, the drilling of a through cylindrical hole is performed in three steps: hole interface creation, interior hole mesh elements removal and hole interface deformation. The Hole Interface is the set **HI** of triangles for 3D mesh (resp. edges for 2D mesh) between the removed mesh elements and the kept mesh elements after cut operation to create the through hole. The HI is deformed to match the cylindrical hole shape by using previously adopted deformation engine with new constraints as input.

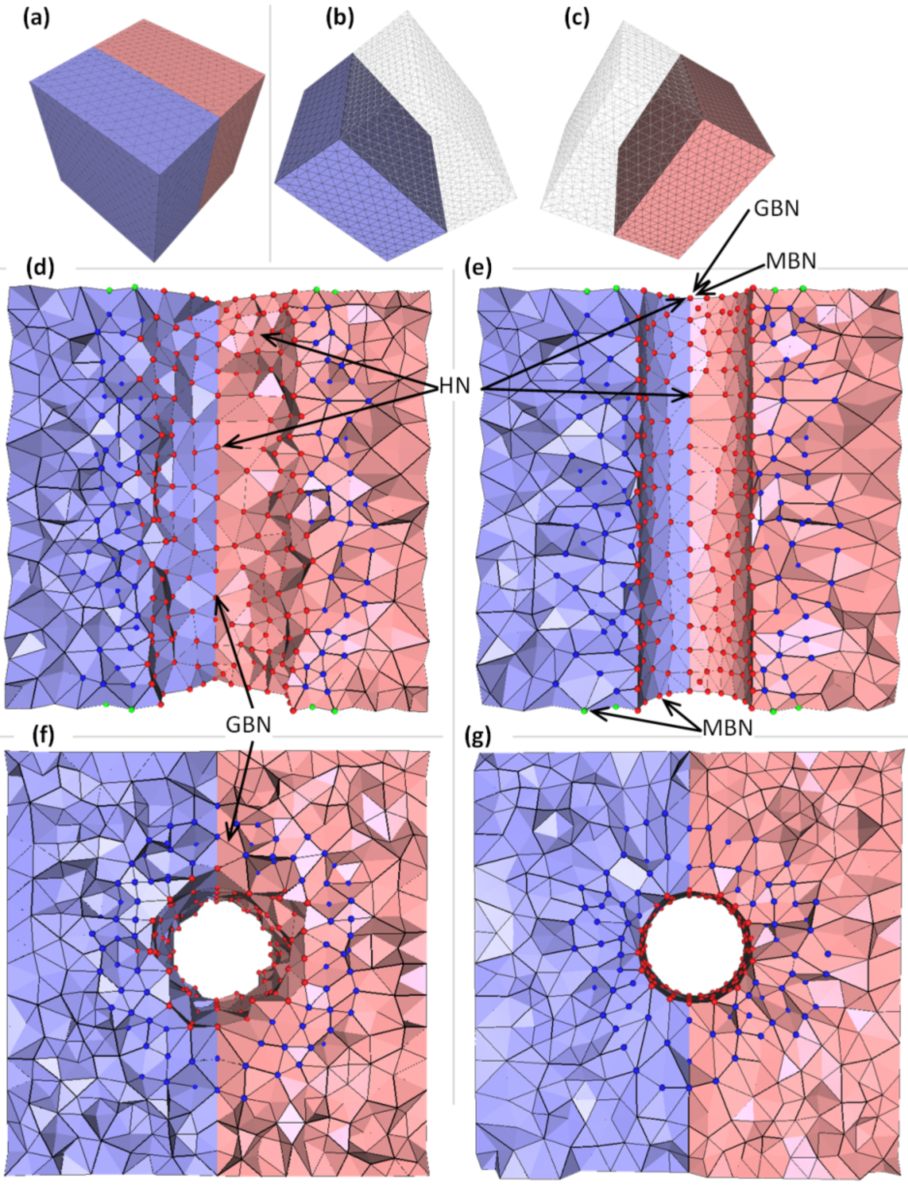

Fig. 13. Creation of a through hole in a 3D mesh of a cube characterized by two FE groups colored in blue and red.

4.2.1. *Cylindrical HI identification*

Similarly to the crack insertion, the nodes of the mesh are separate into two sets which respectively gathers together the nodes inside (**I**) and outside (**O**) of the drilling tool (cylinder).

In the case of 3D mesh (resp. 2D mesh), we define a set **RT** containing all the removed tetrahedra (resp. triangles) whose four nodes (resp. three nodes) are in the set labeled I and, a set **KT** gathering together the remaining tetrahedra (resp. triangles). For 3D meshes (resp. 2D), a set HI is defined by all the triangles (resp. edges) which are exactly shared by one tetrahedron (resp. triangle) in RT and one tetrahedron (resp. triangle) in KT.

4.2.2. *Hole drilling*

Similarly to the crack insertion problem, the geometric constraints specification will not be the same for all the mesh entities, and an identification of specific node sets to be constrained is required.

First, a set HN gathers together all the nodes associated with HI elements. Second, a set **TN** (Transition Nodes) is defined similarly to the ones used in the crack insertion problem, i.e. using a bandwidth of ith neighborhood.

For preserving the shape of the original model we define also a set of MBN (Model Boundary Nodes). This set contains the HN and TN nodes on the mesh boundary.

For preserving the group information the constraint from the virtual group boundary shape is also imposed during the deformation. One important point is that the VGB should be computed before the deletion of the elements in RT. A set of **GBN** (Group Boundary Nodes) is defined for all free nodes on the crack interface and in the transition zone.

The elements corresponding to the hole, RT tetrahedra (resp. RT triangles) are then removed from the mesh model.

Then, geometric constraints are assigned to different nodes depending on the sets to which they belong. For 3D meshes nodes in:

- HN are constrained to stay on the cylindrical hole;
- MBN are constrained to stay on the external skin of the model;
- GBN are constrained to stay on the shape of corresponding group boundary;
- TN could move;

and all the other mesh nodes are fixed. In the case of 2D meshes, nodes in:

- HN are constrained to stay on the cylindrical hole and on the mesh by preserving its surface shape;
- MBN are constrained to stay on the model boundary by preserving its curve shape ;
- GBN are constrained to stay on the shape of corresponding group boundary;
- TN are constrained to stay on the mesh by preserving its surface shape;

and all the others mesh nodes are fixed.

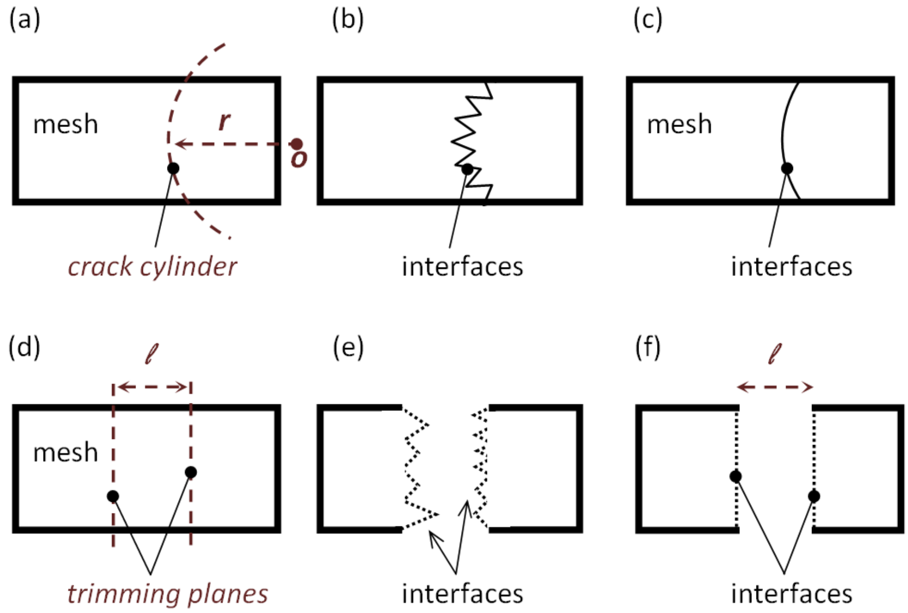

Fig. 14. Overview of two additional operators: cylindrical crack and cutting/incision.

Once the constraints are defined, the deformation engine detailed in section 3.3 gives a solution in term of local deformation of the drilled hole surface.

Figure 13 illustrates the whole process on a cube-like tetrahedral mesh. Two groups of tetrahedra are shown in figures (13.b and 13.c). The mesh model before deformation is shown in figures (13.d and 13.e). First, the removal tetrahedra (RT) and hole interface nodes are identified. These tetrahedra are then removed, the newly appearing boundary triangles constitute the HI and different node sets used by our technique are identified. Next, the constraints are applied, and the solution in term of local deformation of the drilled hole is computed using the deformation engine (fig. 13.f and g).

4.2.3. *Extension to two new operators*

Based on the previously described operators, the cylindrical crack and the incision/cutting operators can easily be defined and implemented.

Figures (14.a to 14.c) illustrate the principle of the cylindrical crack operator, that permits the separation of the object in parts having the contact region of cylindrical shape. Figure 14.a shows an initial mesh on which a cylindrical crack is performed. The profile is defined by a cylinder of center o and radius r. In a first step, using an extension of the HI identification process, the CI is detected (fig. 14.b) and handled similarly to the planar cracks (fig. 14.c), i.e. mesh deformation for reaching the cracking surface, duplication of nodes and updating of the incident elements. Figures (14.d and 14.f) present the principle of the cutting operation. Figure 14.d shows an initial mesh on which an incision having a non null width is performed. The incision is defined by two trimming planes and a distance between them. The first step involves the detection of all the mesh elements which are between the two cutting planes (fig. 14.e). This can be made using an extension of the crack model splitting. Next, the in-between elements are removed, as for cylindrical hole. The two incision interfaces corresponding to the trimming planes are then handled similarly to the crack case (fig. 14.f).

5. Results

This section gives some results relative to the direct modification of industrial 3D FE mesh model using our technique: insertion of different mesh features (crack, hole) following local deformation under constraints taking into account the shape of the group boundaries (low-semantics enriching the FE mesh).

The first example (fig. 15) shows successively the adoption of the developed operators for obtaining the two changes performed by EDF engineers on the caisson model of figure 1. Figure 15.a shows a part of the model in which one stiffener is cracked. The two sub-parts P1 and P2 (depicted in blue and red) of the mesh are defined by using the crack plane. The cracking deformation is limited to the stiffener so that the rest of the sub-parts is kept. Figure 15.c shows the sub-part P1. On the cracked model a cylindrical hole is inserted on the bottom of the stiffener (fig. 15.b). The sub-part P1 on the drilled mesh is shown in figure 15.d. On this example, 84 nodes have been moved onto the crack plane and 341 nodes are moved to relax the transition mesh zone between fixed and free nodes. Therefore, 84 crack plane constraints and 425 tangent plane constraints have been used to accomplish accurately the required deformation.

Another example illustrates the insertion of a cylindrical hole and a crack into a volume mesh on which several groups of tetrahedra are defined (fig. 16.a). Here, not only it is important to obtain the hole and crack feature, but also the boundaries of these groups have to be maintained. Figure 16.b shows the tetrahedron rough deletion for hole insertion. The deformation result of drilling operation is shown by two pictures (fig. 16.c and .d) presenting the model from outside and inside points of view. On the drilled model a limited planar crack is inserted then (fig. 16.e). The cracking deformation is shown in figure 16.f.

6. Conclusion and future works

In this paper, a new framework for CAD-less FEA has been introduced for fast prototyping of product design solutions during the product optimization phase. With the proposed approach, the geometric modifications required to achieve the optimal solution can be directly performed on the FE meshes enriched with their physical semantic information through the use of mesh groups. This avoids the need

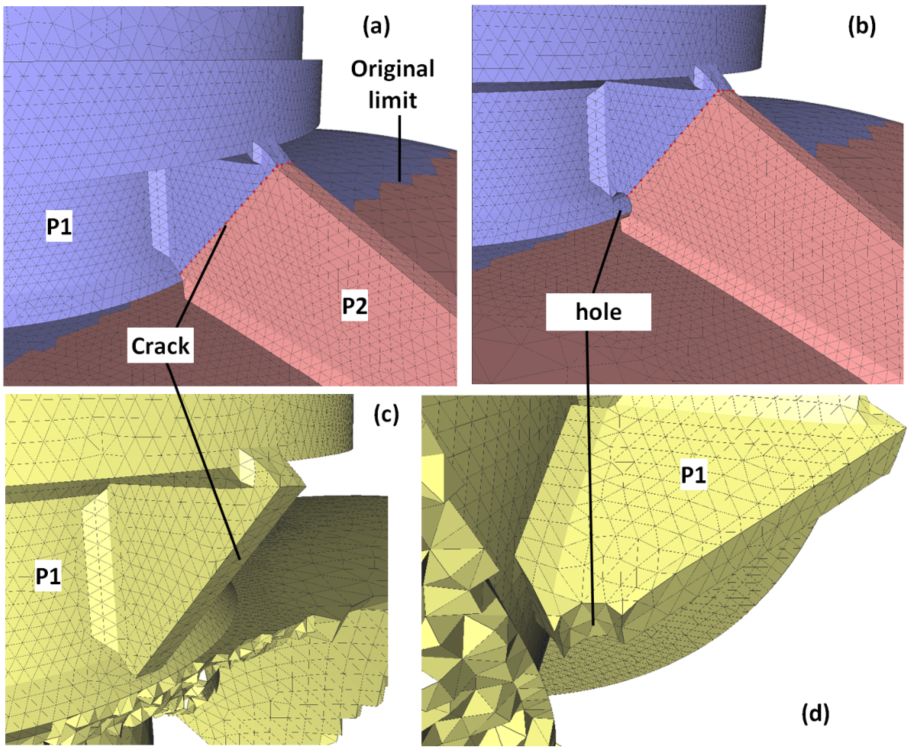

Fig. 15. Insertion of a crack (planar contact zone) (a) (c) and of a cylindrical hole (b) (d) into the 3D FE mesh of a caisson model courtesy EDF-R&D).

to return to the CAD product model and then to re-perform all the time consuming steps necessary to obtain the complete FE model. This capability is also very useful when the CAD model is not available and a mesh could be obtained easily, e.g. through object scanning. Therefore the different alternative solutions could be realized on the available mesh models. To fully exploit such a capability it is important that not only the shape aspects are considered but also the already present semantics is correctly managed. Depending on the modification on the shape and on the specific semantics' meaning, the attached semantic data may need different treatment.

The proposed framework is designed as modular so that the development of new operators simply requires the substitution and/or addition of new components or constraints. Here, two operators have been proposed to cover the primary needs in terms of direct 2D/3D mesh modification: crack and through hole operators. These operators use a deformation engine applying two types of constraints: those relative to the shape of the tools (e.g. cylinder for holes and plane for cracks) and of the object itself and those relative to the group boundaries associated to physical

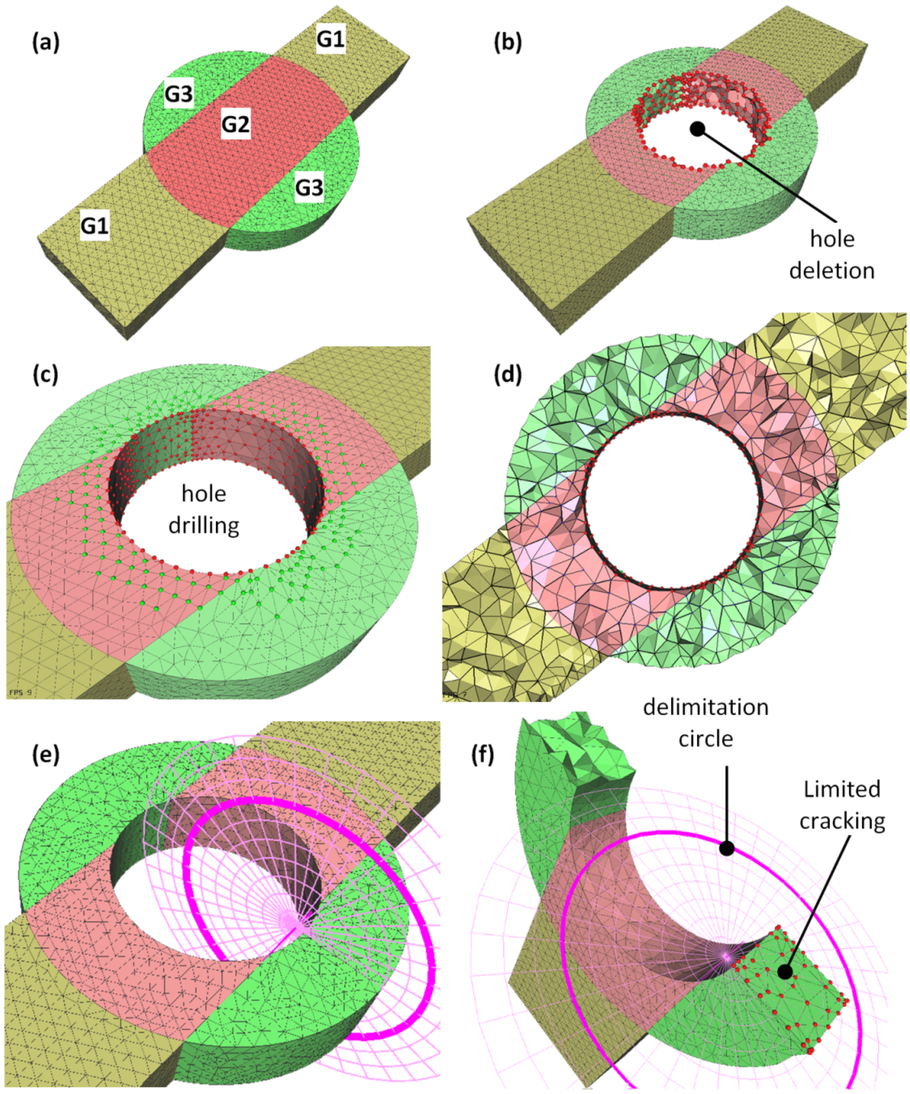

Fig. 16. Insertion of a cylindrical hole and planar crack into a 3D mesh containing three overlapping groups.

semantics that have to be preserved. Since our approach is aimed at complying with the maintenance context in which it is important to deal with FE meshes that correctly reflect the behaviour of the equipment physical counterpart already validated, we reduce as much as possible the existing mesh elements on which the modifications are applied. Thus the mesh is modified only in a restricted bandwidth so that the quality of the elements (e.g. their aspect ratio) remains good with respect to the FEA requirements. Additionally, the proposed operators reposition nodes without adding new ones. Only in special cases, when the operated mesh does not have enough nodes in the modification area, a mesh pre-refinement step is necessary.

Future work will be related to the extension of our mesh modification toolbox to other shapes (e.g. spheres, torus or free form) as well as to other operators such as material addition, which is often required for fast modifying advanced FE mesh models. Moreover, since at present we treat semantics only at the group level preservation, additional effort will be devoted to investigate more into details the propagation (e.g. group extension and gluing) dependent of the semantics meaning.

We can state that the results of the research here described give a new vision about the need and possibilities to develop modelling tools that are not only dealing with the geometric aspects of objects but fully use the attached application specific semantics in order to provide tools that really improve the product life cycle development.

Acknowledgements

This work is supported by the Research and Development direction of the EDF Group and it is partially carried out within the scope of the FOCUS K3D project supported by the European Commission [25].

References

1. J.P.Pernot, B.Falcidieno, F.Giannini and J.C.Leon, Hybrid models deformation tool for free form shapes manipulation, in *Proc. ASME Int. Design Eng. Tech. Conf. & 34th Design Automation Conf.* (New-York, USA, 2008) paper n°DETC08-DAC49524.
2. R.Lou, F.Giannini, J.P.Pernot, A.Mikchevitch, B.Falcidieno, P.Veron and R.Marc, Towards cad-less finite element analysis using group boundaries for enriched meshes manipulation, in *Proc. ASME Int. Design Eng. Tech. Conf. & Computers and Information in Eng. Conf.* (San Diego, USA 2009) paper n°DETC09-CIE86575.
3. R.Lou, J.P.Pernot, A.Mikchevitch and P.Veron, Merging enriched Finite Element triangle meshes for fast prototyping of alternate solutions in the context of industrial maintenance, *J. Computer-Aided Design* **42** (2010) 670–681.
4. A. Mikchevitch, S.Geniaut and I.Nistor, Towards fast numerical studies for maintenance and lifecycle problem analysis: New simulation methods on example of an industrial study case, in *Proc. ASME Int. Pressure Vessels and Piping Division Conf.* (Prague, Czech Republic, 2009) paper n° PVP2009-77154.
5. R.S.Barsoum, Application of quadratic isopa-rametric elements in linear fracture mechanics, *Int. J. of Fracture.* **10** (1974) 603–605.
6. T.Strouboulis, I.Babuska and K.Copps, The design and analysis of the generalized finite element method, *Int. J. Computer Methods in Applied Mechanics and Engineering* **181** (2000) 43–69.
7. E.Bechet, H.Minnebo, N.Moes and B.Burgardt, Improved implementation and robustness study of the X-FEM for stress analysis around cracks, *Int. J. Num. Methods in Engineering* **64**-(8) (2005) 1033–1056.
8. M.Schoellmann, M.Fulland and H.A.Richard, Development of a new software for adap-

tive crack growth simulations in 3D structures, *Int. J. of Engineering Fracture Mechanics* **70**-2 (2003) 249–268.
9. D.Bremberg and G.Dhondt, Automatic crack-insertion for arbitrary crack growth, *Int. J. of Engineering Fracture Mechanics* **75** (2008) 404–416.
10. A. Martinet, E. Galin, B. Desbenoit, S. Akkouche, Procedural modeling of cracks and fractures, in *Int. Conf. on Shape Modeling and Applications* (2004) 346–349.
11. H-W. Nienhuys and A. F. van der Stappen, Supporting cuts and finite element deformation in interactive surgery simulation, Technical Report (2001) Universiteit Utrecht.
12. K.Kundu and M.Olano, Tissue Resection using Delayed Updates in a Tetrahedral Mesh, in *Proc. Medicine Meets Virtual Reality 15, Eds. J.D.Westwood and al., IOS Press* (Netherlands, 2007).
13. Z.Ji, L.Liu, Z.Chen and G.Wang, Easy Mesh Cutting, in *Proc. Eurographics Computer Graphics Forum* **25**-3 (Vienna, Austria, 2006) 283–291.
14. G.Turini, G.Ganovelli and C.Montani, Simulating Drilling on Tetrahedral Meshes, in *Proc. Eurographics Conf.* (2006) 127–131.
15. G.Turkiyyah, W.B.Karam, Z.Ajami and A.Nasri, Mesh cutting during real-time physical simulation, in *Symp. Solid and Physical Modeling* (San Francisco, USA, 2009) 159–168.
16. A. D. Bruyns, S. Senger, A. Menon, K. Montgomery, S. Wildermut and R. Boyl, A survey of interactive mesh-cutting techniques and a new method for implementing generalized mesh cutting using virtual tools, *J. Visualization and Computer Animation* **13** (2002) 21–42.
17. A. Bielser, A. Volker and M. Gross, Interactive cuts through 3-dimensional soft tissue, in *Proc. Eurographics Computer Graphics Forum* **18**-3 (Milan, Italy, 1999) 31–38.
18. P, Krsek, Complex human tissues fem models prepared by boolean operations, in *Biomechanics of man* (Prague, CZ, 2002) 24–26.
19. M. Dakowicz and C. M. Gold, Interactive TIN modification with a cutting tool, in *Proc. Workshop on Dynamic and Multidimensional GIS* (Pontypridd, Wales, 2005) 5–9.
20. M. Campen and L. Kobbelt Exact and Robust (Self-)Intersections for Polygonal Meshes, in *Eurographics Computer Graphics Forum* **29**-2 (Norrköping, Sweden, 2010) 397–406.
21. R. Pfeifle and H.P. Seidel, Triangular B-Splines for blending and filling of polygonal holes, in *Proc. Graphics Interface conf.* (Toronto, Canada, 1996) 186–193.
22. M. Bern and P. Plassman, Mesh generation, in *Handbook of Computational Geometry* (Elsevier, Amsterdam, 2000) 291-332.
23. A. Bargier, F. Giannini, R. Lou, J-P. Pernot Surface primitive recognition, Technical Report CNR-IMATI (Genova, Italy, 2009).
24. M. Attene, B. Falcidieno and M. Spagnuolo, Hierarchical mesh segmentation based on fitting primitives. *J. Visual Computer* **22**-3 (Springer-Verlag, 2006) 181–193.
25. European Union's Framework Programme FOCUS K3D Foster the comprehension and use of Knowledge intensive 3D media, website http://focusk3d.eu
26. Code_Aster®, website http://www.code-aster.org