

Generalizing the advancing front method to composite surfaces in the context of meshing constraints topology

Gilles Foucault, Jean-Christophe Cuillière, Vincent François, Jean-Claude Léon, Roland Maranzana

▶ **To cite this version:**

Gilles Foucault, Jean-Christophe Cuillière, Vincent François, Jean-Claude Léon, Roland Maranzana. Generalizing the advancing front method to composite surfaces in the context of meshing constraints topology. Computer-Aided Design, 2013, 45 (11), pp.1408-1425. 10.1016/j.cad.2013.05.009 . hal-00849780

HAL Id: hal-00849780
https://hal.science/hal-00849780

Submitted on 1 Aug 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Generalizing the Advancing Front Method to Composite Surfaces in the Context of Meshing Constraints Topology

G. Foucault[(1)], J-C. Cuillière[(2)], V. François[(2)], J-C. Léon[(1)], R. Maranzana[(3)]

[(1)] Institut National Polytechnique de Grenoble, Laboratoire G-SCOP, France, Gilles.Foucault, Jean-Claude.Leon@grenoble-inp.fr

[(2)] Université du Québec à Trois-Rivières, CP 500, Trois-Rivières (Québec) G9A-5H7, Canada Jean-Christophe.Cuilliere, Vincent.Francois @uqtr.ca

[(3)]École de Technologie Supérieure de Montréal, GPA, Laboratoire LIPPS, Canada, Roland.Maranzana@etsmtl.ca

Abstract

Being able to automatically mesh composite geometry is an important issue in the context of CAD-FEA integration. In some specific contexts of this integration, such as using virtual topology or Meshing Constraints Topology (MCT), it is even a key requirement. In this paper, we present a new approach to automatic mesh generation over composite geometry. The proposed mesh generation approach is based on a generalization of the advancing front method (AFM) over curved surfaces. The adaptation of the AFM to composite faces (composed of multiple B-Rep faces), involves the computation of complex paths along these B-Rep faces, on which progression of the advancing front is based. Each mesh segment or mesh triangle generated through this progression on composite geometry is likely to lie on multiple B-Rep faces and is, by the way, likely to be associated with a composite definition across multiple parametric spaces. Collision tests between new front segments and existing mesh elements also require specific and significant adaptations of the AFM since a given front segment is also likely to lie on multiple B-Rep faces. This new mesh generation approach is presented in the context of MCT, which requires being able to handle composite geometry along with non-manifold boundary configurations, such as edges and vertices lying in the interior domain of B-Rep faces.

Keywords

Advancing front, mesh generation, mesh adaptation, composite geometry, virtual topology, defeaturing.

1. Introduction

Progresses towards a better integration between Finite Element Analysis (FEA) and Computer Aided Design (CAD) has greatly reduced the median time requirements to prepare FEA models [1]. One of the key steps in the integration between CAD and FEA is automatic surface meshing because, in most approaches, filling the FEA domain with volume elements (tetrahedrons or hexahedrons) is based on a previous discretization of the domain's boundary and because the quality of this discretization has a major effect on the quality of the resulting 3D mesh with volume elements and on the reliability of the FEA solution. Surface meshing [2-10] has been investigated by many researchers for the last twenty years. It can basically be performed either on a collection of trimmed parametric patches (like when meshing the faces of a B-Rep model) or on a tessellated representation of geometry (a STL file for example) and it can either be directly generated on surfaces in 3D or indirectly generated in the parametric space and then mapped in 3D [11]. However, the automatic generation of surface meshes from CAD models and by the way the integration between FEA and CAD still faces several problems, especially when these CAD models feature many small shape details, and when their Boundary Representation (B-Rep) contains a large number of faces, many of them being eventually much smaller than the desired FE size. Such configurations are likely to be at the origin of either poorly-shaped elements and/or over-densified elements, not only increasing the analysis time, but also producing poor simulation results or meshes incompatible with

solver requirements. Applying geometry simplification and virtual topology techniques to these configurations has proven to be a very interesting option.

Efforts have been made, aimed at avoiding poorly-shaped elements and over-densified mesh elements generated from raw CAD models [12-16]. In these approaches, well-known mesh topological transformations perform mesh element removal (e.g. decimation of surface meshes) or collapsing faces of tetrahedrons to remove these tetrahedrons. These transformations can be applied on the mesh itself or on a discrete representation of geometry as proposed in [17]. One limitation of these operators holds in the fact that collapsing operations are not intrinsically suited for the removal of through holes, which makes that these operators require complex extensions when applied to real life industrial parts. Therefore, removing holes may require specific processing such as manually removing these details in the CAD model. Element distortion (poorly-shaped elements) and over-densified mesh elements are also closely related to the input mesh sizing objectives. Setting up appropriate mesh sizing objectives [18-20] can actually contribute to lower elements distortion and to avoid over-densified mesh elements.

Other approaches consist of adapting CAD models directly. For example, feature recognition and extraction processes can be intended to simplify details like fillets and blends [21-22], bosses, pockets [23] and holes. Some of these approaches generate tree-structured simplified models, where each simplification is identified as a feature. Lee et al. [24] propose a feature removal technique which starts from a feature tree and provides the ability to suppress, and subsequently reinstate features, independently from the order in which they were suppressed, within defined limitations. Another alternative is applying geometry transformation with the use of CAD kernel operators aimed at locally modifying the topology and/or underlying geometry in the Boundary representation (B-Rep) model [22-23, 25]. Overall, these approaches manage a restricted set of feature types and interactions between features remains a major issue. Moreover, even recognized features are often difficult to suppress, which makes these approaches non-robust and very restrictive when used alone. Nevertheless, these approaches are efficient when used to remove holes and bosses prior to mesh simplification. A very good survey of simplification techniques can be found in [26].

However, feature suppression does not guarantee that the object's boundary decomposition can be directly used for automatic mesh generation purposes. In this context, virtual topology techniques (also referred to as virtual geometry in some papers) can be applied afterwards with the objective of adapting the boundary decomposition to the requirements of automatic mesh generation. Virtual topology approaches proposed by Sheffer et al. [27-28], Inoue et al. [29], Wei [30] and Tautges [31] are aimed at the edition of the B-Rep definition of a CAD model to produce a new topology, which is more suited to mesh generation constraints and which contributes to avoid poorly-shaped and over-densified mesh elements. These approaches implement split and merge operators aimed at clustering adjacent faces into nearly planar regions in order to generate a new B-Rep topology, which is better suited to mesh generation, while preserving its geometry. However, face clustering algorithms show limitations in the context of FE models preparation: they do not support the definition of edges and vertices interior to faces, while these non-manifold surface configurations are required for various needs (modelling boundary conditions, taking into account specific features and sharp curves lying inside faces).

Toward the objective of overcoming the limitations of virtual topology, we have introduced *Meshing Constraints Topology (MCT)* concepts [32] in a previous work. One of the basic and

most important features of *MCT* is that it enables non-manifold surface transformations: edge deletion, vertex deletion, edge splitting, edge to vertex collapsing, vertices merging. Starting from a given B-Rep model to be meshed, generating the *MCT* is based on a local analysis of B-Rep faces and edges with regard to mesh generation constraints (local face width, normal vector deviation across edges, etc.). An interesting feature of the *MCT* is that it enables the definition of interior edges and vertices on B-Rep faces when required. As described with more details in the next sections, *MCT* models are basically defined using composite geometry. For instance, a composite face (referred to in our work as an *MC face*) is defined as a set of adjacent faces in the input B-Rep structure, each of which associated with an underlying surface and its parametric definition. Consequently, meshing models that have been processed with *MCT* concepts requires the ability of automatically generating finite elements across composite geometry (typically composite edges and faces). At this point, mesh generation techniques aimed at handling composite surfaces are based on the following two main approaches:

The first approach is computing a global parameterization [33-34]: this basically consists of defining a bijective mapping between any point inside sets of surface patches and a global parametric domain. The bijective transformations proposed in [33-34] are both based on a cellular decomposition of each reference (non-composite) surface mapped into the global parametric space. Each cell is associated with a specific mapping to a reference surface image, and with a global parametric space image. Any point of a cell in the global parametric space is represented using its barycentric coordinates, and projected in the equivalent cell of the corresponding reference surface using the same barycentric coordinates. This new parameterization enables a natural and transparent use of parametric meshing schemes. Unfortunately, this type of approach is limited because:

- It is restricted to open and non-periodic surfaces: this method can only be applied to open surfaces (homeomorphic to a disc), but not to periodic surfaces and closed surfaces homeomorphic to an n-torus.
- It is restricted to smooth and nearly planar domains: the parametric domain of distorted surfaces may cause high variations in the global parameterization metrics due to the surface embedding, and often results, either in failures during mesh generation or highly distorted meshes [33].
- Identifying the outer loops of composite faces is not automatic.

The second approach consists of using direct 3D advancing front techniques on a tessellated (triangulated) model of composite surfaces [10]. This type of approach also shows significant limitations because:

- Deviation of the final triangulation from the input model depends on the tessellation's accuracy.
- More sophisticated discrete representations such as subdivision surfaces and higher order triangulations allow curved mesh generation but still generate approximation errors.

In order to overcome these limitations, this paper introduces a new approach to automatic mesh generation over composite surfaces, which is based on an adaptation of the advancing front method (AFM) over curved surfaces [5, 8, 10, 35-38]. The AFM [5, 7-8, 10, 35-41] is a very common automatic mesh generation technique and the main features of its adaptation to the context of composite surfaces are:

- Elements needs to be generated across multiple parametric surfaces: advancing front propagation thus needs to be adapted through the extension to composite surfaces of propagation direction, propagation length, and target point calculations,
- Every mesh entity is associated with its image in each component (reference entity) of the composite surface,
- Intersection tests between 3D segments are performed in the parametric domain of their images.

The paper is structured as follows:

- Section 2 presents the context of adapting CAD models for FEA and introduces Meshing Constraints Topology (*MCT*), which is one of the key inputs of the mesh generation process presented in this paper.
- Section 3 briefly reviews the main steps involved in the Advancing Front Method (AFM) applied to automatic mesh generation over parametric surfaces.
- Section 4 presents the extension (data-structures and algorithms) of the AFM, in the context of mesh generation over composite geometry, which is the core of this paper.
- Section 5 illustrates the potential and efficiency of the method proposed through the presentation of mesh quality improvements obtained on several practical examples.

2. Adapting CAD models for meshing

2.1. Setting a finite element size map

The general context of the present work is performing, as efficiently and quickly as possible, a FEA from the input of a detailed CAD model. This detailed CAD model is intended for product definition purposes and not for FEA purposes, which implies that modifications and simplifications are likely to be applied on this model before FEA. Also, prior to FEA itself, the analyst must specify mechanical hypotheses and boundary conditions on the CAD model (i.e. materials, loads and restraints). The analyst also needs to specify an objective with regard to the desired analysis accuracy. In a priori adaptive meshing, based on his FEA skills and experience, the analyst specifies a FE size map that is adapted to both the geometry and the FEA objective. This size map can either be rough (to quickly obtain an approximate solution) or featuring a certain degree of adaptive refinement (to obtain accurate results in zones of interest). This FE size map is a central issue in our automatic feature removal and topology adaptation processes because it represents the analyst's intent with regard to the size of shape features that can be neglected for analysis purposes. This size map is likely to vary significantly throughout the analysis domain, depending on where the zones of interest are located and it is consistent with the specification of a clear analysis objective. The automatic feature removal and topology adaptation processes are strongly driven by this FE size map.

2.2. CAD details causing mesh inconsistencies

Trying to generate a mesh with large elements on a CAD model featuring rather small entities, i.e. an over-detailed model, leads non-convergence or to the generation of sliver (very flat) elements. Automatically meshing CAD models is typically performed through three steps, which are is closely related to the B-Rep structure's hierarchy. At first, edges are discretized with sets of mesh segments, whose lengths are consistent with the FE size map. Then, a triangulation algorithm fills the interior of each B-Rep face with triangles; the

triangulation is constrained by the prior discretization of edges mesh and is also consistent with the FE size map required. Processing the entire B-Rep of a given part results in a closed triangulation of its boundary. The last step consists of using a tetrahedrization algorithm to fill the closed triangulation with tetrahedrons, consistently with the FE size map as for the discretization of edges and faces. Consequently, due to the fact that standard meshing procedures are constrained by the B-Rep topology (the mesh is constrained by all B-Rep edges) when trying to mesh an over-detailed CAD model, it results in the following mesh inconsistencies:

- Mesh over-densities: e.g. small triangles are propagated from tiny segments generated on tiny edges of the B-Rep (see Fig. 1),
- Poorly-shaped elements: e.g. sliver triangles are originated from small segments and narrow faces (see Fig. 1).

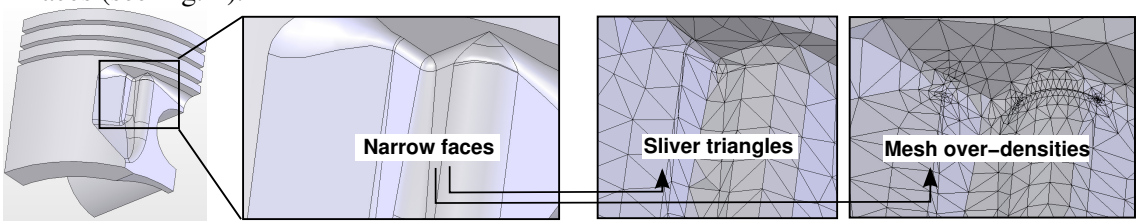


Fig. 1. Example of narrow faces and small edges leading to sliver triangles or mesh over-densities

2.3. A three step approach to the automatic adaptation of CAD for FEA

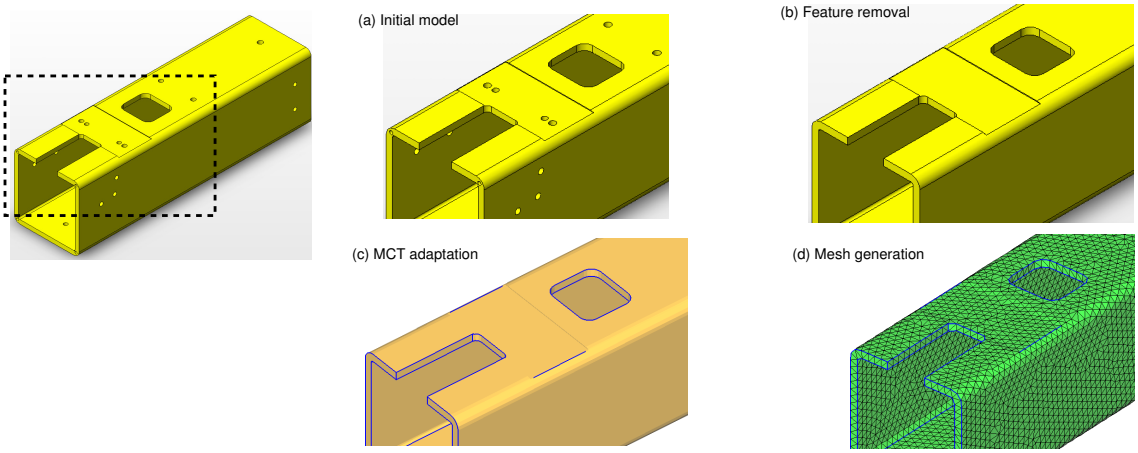


Fig. 1. Preparation of a CAD model for mesh generation

We present a fully automated adaptation process, which is intended to improve the efficiency and speed of preparing CAD models for FEA. The automatic simplification criteria used are based on the imposed size map and the process takes place through the following three steps:

- *Step1- Feature removal:* CAD design features (holes, fillets, pockets and protrusions) that are too small (with respect to the imposed size map) to affect analysis results in zones of interest are automatically identified as shape details (see fig. 1(b)). Two types of operations are used to remove these details automatically:
  (a) Suppressing features directly in the CAD feature-tree: this applies for details that have been designed as features, and for which suppression does not affect any other feature.
  (b) Using *delete-face* type operators: this type of operators performs the elimination of selected faces and the reconstruction of a closed solid envelope by filling holes [23-24].

- *Step2- MCT adaptation:* the B-Rep topology obtained after feature removal usually requires additional preparation for mesh generation. Indeed, small edges and narrow faces must be transformed to avoid over-constraining mesh generation as mentioned above. *MCT* operators and criteria based on adjacency hypergraphs [32, 42] have been designed for the automatic adaptation of B-Rep topology (see fig. 1(c)):
  - *MCT* criteria, based on the size map and taking into account zones where boundary conditions are applied, automatically identify which *MCT* operations are required from the following:
    - Removing irrelevant edges (located in narrow faces or planar surfaces) using edge deletion.
    - Removing irrelevant vertices (located on small edges or smooth curves) using vertex deletion or edge contraction.
    - Collapsing constricted sections of faces by merging vertices.
  - *MCT* operators are then automatically applied by editing the *MCT* topology hypergraphs and their underlying geometry [32]: edge and vertex deletion, edge splitting, edge to vertex collapsing.
- *Step3-Meshing the MCT model:* a mesh is automatically generated from the *MCT* model (see fig. 1(d)) using the AFM. The AFM is first initialized by discretizing *MC edges* (which are composite edges). Then the mesh is propagated inside *MC faces* (composite faces). The detailed description of this automatic mesh generation process is the core of this paper.

2.4. Meshing Constraints Topology (*MCT*)

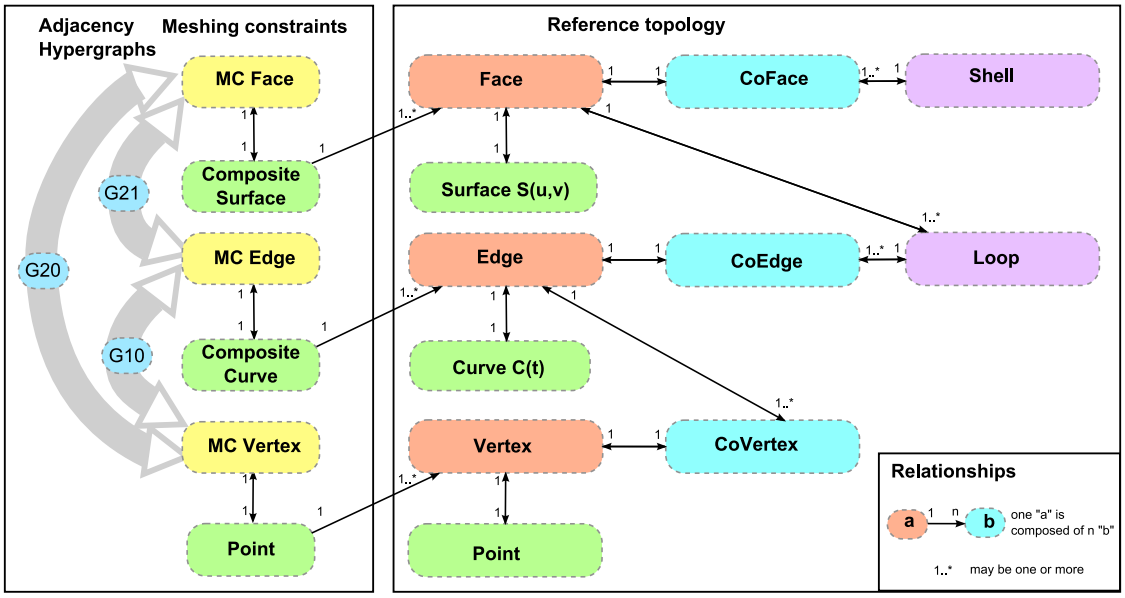


Fig. 2. (left) the *MCT* composition diagram (right) reference topology composition diagram

As introduced in the previous paragraph, *MCT* adaptation basically consists in adapting the B-Rep topology to the requirements of mesh generation. Thus, *MCT* (see reference [32] for a detailed description) basically appears as an enhanced B-Rep structure, featuring *MC* entities, which are fundamentally composite entities. As for classical B-Rep structures, a *MCT* model includes a full description of the entities orientation and topological links, as shown in fig. 2.

In the *MCT*, topological links between i-manifold *MC entities* and j-manifold *MC entities* ($i \in [1,2]$ and $j < i$) are represented by adjacency hypergraphs referenced as *G.i.j*:

- *G.2.1* is the face-edge adjacency hypergraph where:
  - each node represents a *MC face*
  - each arc is a *MC edge* connecting a set of *MC faces*;
- *G.2.0* is the face-vertex adjacency hypergraph where:
  - each node represents a *MC face*
  - each arc is a *MC vertex* connecting a set of *MC faces*;
- *G.1.0* is the edge-vertex adjacency hypergraph where:
  - each node represents a *MC edge*
  - each arc is a *MC vertex* connecting a set of *MC edges*.

***MCT* entities** (opposed to **reference entities**) are defined as outlined below:

Definitions
*The **reference model** is the B-Rep model obtained after performing the first step of the process presented above (after feature removal and prior to topology adaptation).*
***Reference entities** (namely reference face, reference edge, reference vertex) are topological entities of the reference model. Their underlying geometry is represented through a single mathematical definition:*
- *the surface underlying a reference face is a Riemannian surface, i.e. a plane, a sphere, a torus, a NURBS, etc*
- *the curve underlying a reference edge is a Riemannian curve, i.e. a line, a circle, an ellipse, a NURBS, etc*

***MCT entities** (namely MC face, MC edge, MC vertex) are topological entities representing meshing constraints. Their underlying geometry is composite, and represented through a set of adjacent reference entities:*
- *The composite surface underlying a MC face is defined as the union of its reference faces,*
- *The composite curve underlying a MC edge is defined as the union of its reference edges,*
- *The point underlying a MC vertex is simply a point coinciding with an underlying reference vertex.*

Thus, unlike usual B-Rep curves and surfaces, *Composite curves* and *Composite surfaces* can feature tangency and curvature discontinuities. Fig. 3 illustrates a *MC edge* composed of a set of two adjacent reference edges. It is worth noting *MC Edges* and *MC vertices* can be located on the interior of a MC face (see Fig. 4).

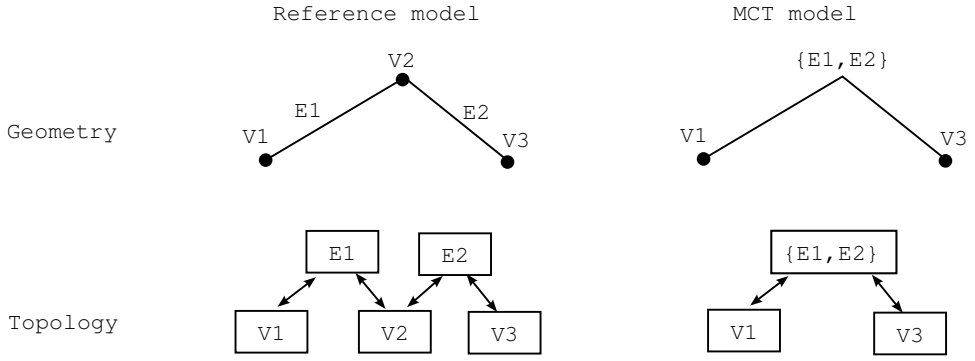


Fig. 3. (a) The geometry and topology of two adjacent edges in the *reference model*. (b) The geometry and topology of the *MCT* model obtained by merging edges E_1 and E_2.

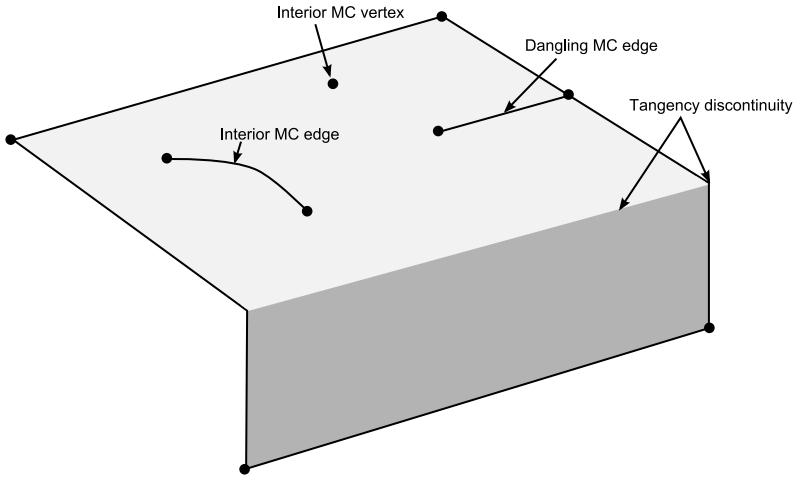


Fig. 4. *MC* face featuring interior *MC* edges (interior and dangling), and one interior *MC* vertex.

3. The Advancing Front Method (AFM)

This section briefly reviews the main steps involved in the AFM when applied to mesh generation over parametric surfaces (see ref [5]). Section 4 will present the adaptation of this classical scheme in the context of mesh generation over composite surfaces.

Meshing a B-Rep model with the classical AFM scheme is achieved through four main steps. These steps are closely related to the B-Rep structure. At first, nodes are generated on the vertices of the B-Rep structure, which is followed by the discretization of B-Rep edges (second step). The third step consists of extracting, from the former results and for each B-Rep face, the discretization of its bounding loops. This initiates the application of the AFM to the triangulation of B-Rep faces which finally leads to a closed triangulation of the CAD model's boundary. The tetrahedrization scheme (fourth step) completes the process and at the end, this closed triangulation is totally filled with tetrahedrons.

When the B-Rep is triangulated, each B-Rep face is triangulated independently by propagating triangles from the discretization of its bounding loops. For a given face, the triangulation stops when the face is completely filled with triangles. For each face, this triangulation process starts with an initial front that is composed with segments issued from the discretization of the face's bounding loops. Along the whole process, each front segment is oriented and defined by two nodes ($\boldsymbol{P_i}$ and $\boldsymbol{P_{i+1}}$ for the i^{th} segment). Starting from the initial front, a recursive process is applied for which, at each step, one (two in some cases) triangle is propagated from one (or two) of the current front segment $\boldsymbol{P_iP_{i+1}}$ (usually the smallest front segment), which is followed by updating the advancing front consistently with the generation of a (or two) new triangles. At each step, the local front configuration for candidate front segment $\boldsymbol{P_iP_{i+1}}$ is analysed with respect to 6 possible configurations (as illustrated in Fig. 5). Then, one of 6 operators is applied, consistently with the local configuration. The first five local configurations and associated operators only use existing nodes while a new node is likely to be created when using the sixth operator. The first operator applies when the front is reduced to 3 segments (see configuration #1 in Fig. 5). In this case, a triangle is simply generated from these 3 segments. Similarly, when the front is reduced to a cavity bounded by 4 segments (see configuration #2 in Fig. 5), two triangles are generated by creating a diagonal segment. The alternative generating two triangles with the highest quality is chosen (out of

two possible diagonals). For configurations #3 to #6, angles α_1 and α_2 between $\boldsymbol{P_iP_{i+1}}$ and its previous ($\boldsymbol{P_{i-1}P_i}$) and next ($\boldsymbol{P_{i+1}P_{i+2}}$) front neighbours are considered. The criterion used is that a cavity in the front has to be filled with triangles if a segment features an angle that is smaller than 80° with one of its neighbours (see configurations #3 to #5 in Fig. 5). For configuration #3, the alternative (out of two possible diagonals) resulting in the highest triangle quality is also chosen.

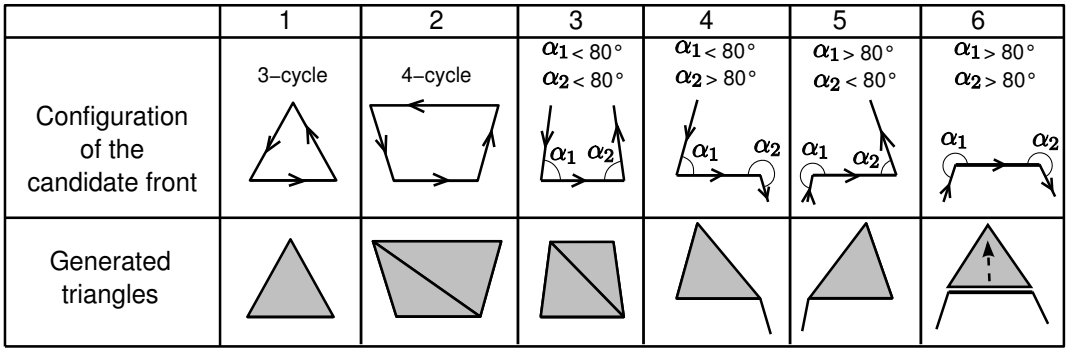


Fig. 5. The 6 front configurations and their specific triangulation process

The last configuration corresponds to the case where a new node (referred to as a candidate node) needs to be considered, and eventually created, from candidate segment $\boldsymbol{P_iP_{i+1}}$. The optimal position $\boldsymbol{P_{opt}}$ for this candidate node is defined as follows: starting from $\boldsymbol{P_M}$, the middle of $\boldsymbol{P_iP_{i+1}}$, $\boldsymbol{P_{opt}}$ is located so that segment $\boldsymbol{P_MP_{opt}}$ is perpendicular to $\boldsymbol{P_iP_{i+1}}$ and that the curvilinear distance d from $\boldsymbol{P_M}$ to $\boldsymbol{P_{opt}}$ is:

$$d = \sqrt{3}/2 \cdot \left(w_T \cdot H(\mathbf{P}_M) + w_s \cdot \left\| \mathbf{P}_i\mathbf{P}_{i+1} \right\| \right)$$

where w_T *and* w_s *are two weights with* $w_T + w_s = 1$

$H(\boldsymbol{P})$ represents the target size (the FE size map) at location $\boldsymbol{P_M}$.

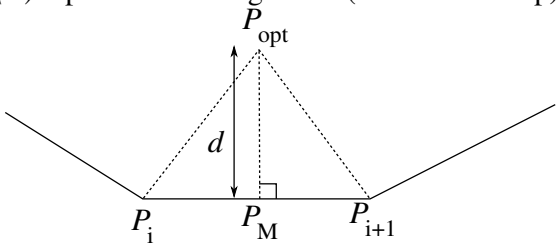


Fig. 6. Optimal point location for triangle generation

Thus, d reflects a compromise made between element shape and element size requirements which can be contradictory (see fig. 6). w_T controls respecting the size map while w_S controls generating elements with a good aspect ratio (the optimal aspect ratio being for an equilateral triangle where $d = \sqrt{3}/2 \cdot \left\| \mathbf{P}_i\mathbf{P}_{i+1} \right\|$). Past experience has shown that (w_T, w_S)=(0.65, 0.35) is a good practical compromise between these two requirements.

In the case where an existing node $\boldsymbol{P_F}$ is close enough to $\boldsymbol{P_{opt}}$ (see Fig. 7), then the triangle ($\boldsymbol{P_i}$ $\boldsymbol{P_{i+1}}$ $\boldsymbol{P_F}$) is generated instead of ($\boldsymbol{P_i}$ $\boldsymbol{P_{i+1}}$ $\boldsymbol{P_{opt}}$). When several existing nodes are found around $\boldsymbol{P_{opt}}$, $\boldsymbol{P_F}$ is chosen as the closest point to $\boldsymbol{P_{opt}}$. The details about the way vicinity between $\boldsymbol{P_F}$ and $\boldsymbol{P_{opt}}$ is computed can be found in [43].

Collision detection tests, before validating each triangle creation, are required to avoid overlapping triangles. The front is updated at each step and the whole process stops when the front is empty.

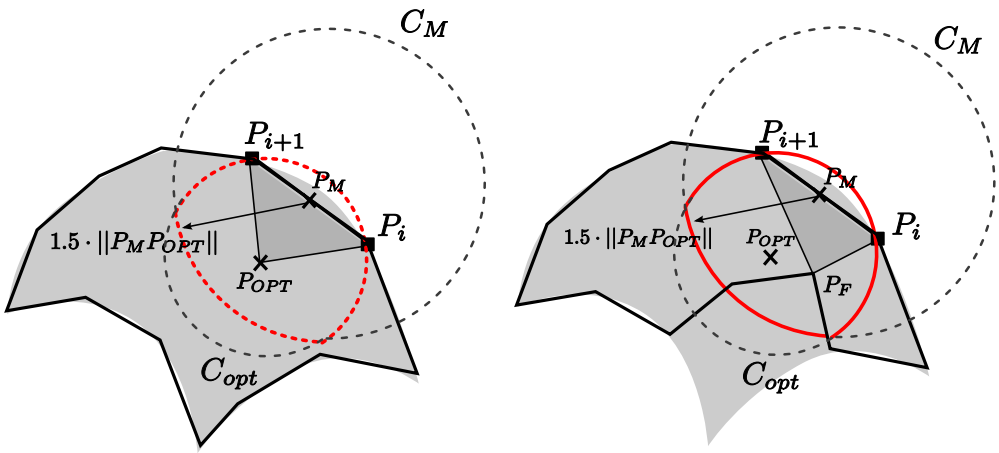


Fig. 7. (a) Triangle created from the candidate node at the optimal location (b) Triangle created from an existing node located inside the search area

4. Extension of the AFM to composite geometry

The main challenge in extending the AFM to *MCT* models is that the geometric definition of *MCT* components is composite (composite surfaces and composite curves). Consequently, it requires an extension of the AFM principles to composite geometry, which means being able to discretize composite curves and triangulating composite surfaces. The most challenging part of this adaptation is extending the AFM to triangulating composite surfaces. The main problems to overcome are related to the fact that each mesh segment and mesh triangle generated through the AFM progression on composite geometry is likely to lie on multiple B-Rep faces and is, by the way, likely to be associated with a composite definition across multiple parametric spaces. Collision tests between new front segments and existing mesh elements also require specific and significant adaptations of the AFM since a given front segment is also likely to lie on multiple B-Rep faces.
In this work, the basic principles applied for this extension are:

- The triangulation process is performed over composite surfaces using the same overall AFM scheme as in the case of non-composite surfaces [5, 8, 10, 35-38]. This scheme notably allows handling non-manifold boundaries, which is an important requirement because, as mentioned previously, *MCT* naturally generates such configurations.
- Triangles are directly generated over multiple parametric definitions of reference surfaces underlying composites surfaces. This allows avoiding the problems and limitations of using a global parameterization of composite surfaces.

The extension of the AFM to the triangulation of composite surfaces raises the following major problems:

- *The association of mesh elements with multiple reference entities:* our extension aims at generating mesh elements over composite curves and surfaces. In this extension, most algorithms (front propagation, intersection tests, shortest path estimation, and tangent vector evaluation) require computing and handling images of mesh elements in the parametric domain (for faces and edges) of the underlying reference entities of the *MCT*. This information about images of mesh elements in the parametric domain of reference entities should be easily accessible from mesh data-structures.
- *The creation of explicit B-Rep data-structure featuring explicit co-edges and loops:* the front initialization, which is inherent to the AFM, relies on loops and co-edges of the B-Rep

data-structure. Co-edges must be sorted in loops to avoid self-overlapping problems, which is required to enable the front's propagation initialization (from loops).

- *Front initialization:* the front has to be initialized from the *MCT,* which is made of composite surfaces, curves, and vertices where adjacency links are described in the B-Rep data-structure. This initialisation consists in 2 steps: the composite curves discretization, and the front ordering based on the orientation of co-edges and loops.
- *Normal vector evaluation at the location of first-order discontinuities:* the evaluation of normal and tangent vectors is mathematically undetermined on curves and surfaces at the location of first-order discontinuities (corners and sharp edges). Pseudo-normal vectors need to be defined and computed at these locations because the AFM requires such vectors to set front propagation direction.
- *Front propagation:* front propagation and element generation have to be processed on multiple bi-parametric surfaces (see Fig. 8),
- *Shortest path estimation on composite surfaces*: sharp edges or small details can be located inside composite surface and this induces important tangency discontinuities across which triangles must be generated. These surface discontinuities make it impossible to approximate curvilinear distance with Euclidian distance (see Fig. 9(a)). As a result, neighbouring mesh elements have to be searched by considering the curvilinear distance (in the context of multiple parametric surfaces as illustrated in Fig. 9(c)) between nodes instead of the Euclidean distance (see Fig. 9(b)). This curvilinear distance has to be estimated using a shortest path (over multiple parametric surfaces) search algorithm (see section 4.5.2).

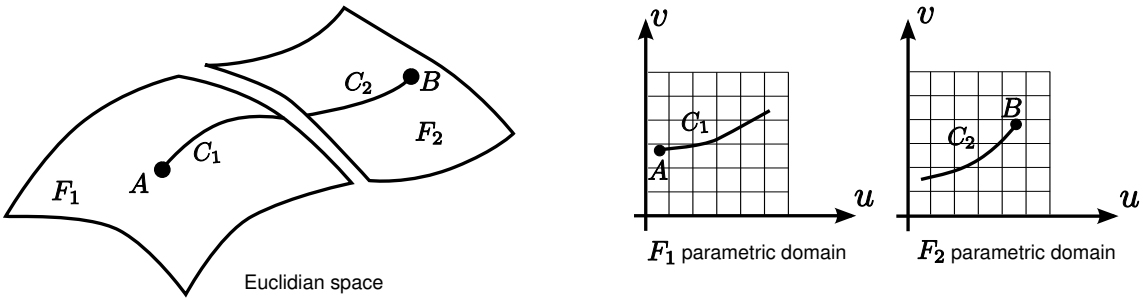


Fig. 8. Front propagation over multiple bi-parametric surfaces, without global parameterization.

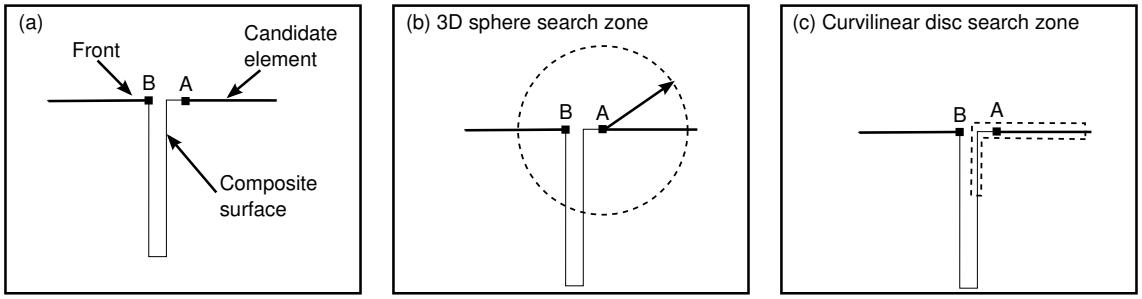


Fig. 9. Validation of front propagation in thin walled surfaces (a) front configuration (b) neighbor found using the Euclidian distance (c) no neighbor found using a curvilinear distance.

The following sections of the paper present how these problems can be solved. We start with the presentation of mesh generation data-structures and algorithms supporting the extension of AFM concepts to the context of composite geometry.

4.1. Introduction of explicit Co-Edges and Loops in the *MCT* data structure

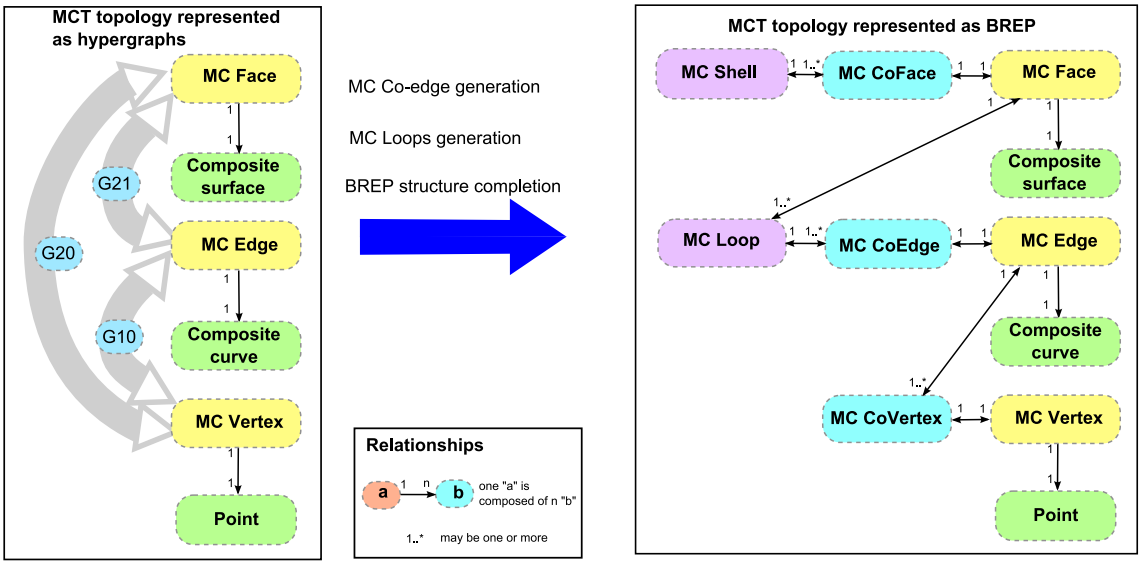


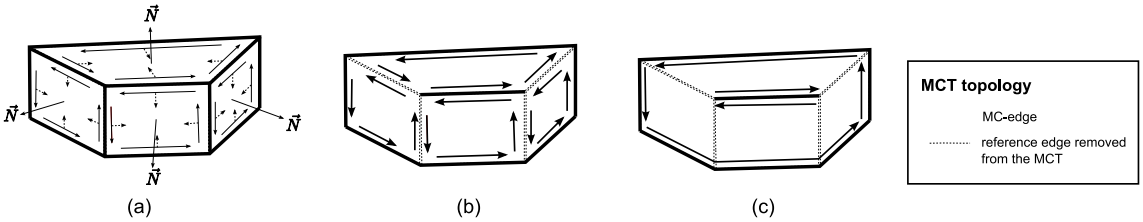
Fig. 10. Transformation of *MCT* adjacency hypergraphs into a B-Rep-like structure

A crucial point when trying to generalize the AFM to composite surfaces, in the context of using *MC* topology, is the transformation of *MCT* adjacency hypergraphs into a coherent B-Rep data structure (referred to as a winged-edge structure):

- Any *MC* edge has to be associated to two partner co-edges C_1 and C_2 whose orientation should be opposite [32].
- Any *MC* face is associated with loops which are composed of co-edges and these loops underlie the front initialization in the AFM. These *MC* face loops must be built with the objective of avoiding front overlaps during the front propagation process.

Methods for the creation of partner co-edges of *MC* edges and loops of *MC* faces are proposed in sections 4.1.1, 4.1.2.

4.1.1. Creating the explicit orientation of co-edges



(a) (b) (c)

Fig. 11. Creation of co-edges : (a) co-edges oriented with the right-hand rule in the reference topology (b) orientation of reference edges underlying *MC* edges (c) orientation of *MC* edges inferred from the reference topology

In the definition of a B-Rep structure, co-edges are addressing the orientation of loops surrounding B-Rep faces. Co-edges associated with a given loop of a given face are oriented so that, at a given location along the face's boundary, the cross product of a co-edge tangent vector and the face normal vector is directed towards the interior of this face, which means towards the front propagation direction (see Fig. 11 (a)). In *MC* topology, the interior domain of a *MC* face is defined explicitly by reference faces being grouped into a single composite surface, therefore:

- The interior domain of a *MC* face equals the interior domain of its underlying reference faces,
- The orientation of co-edges defining the orientation of a *MC* face is the same as those of co-edges addressing the reference faces of this *MC* face (see Fig. 11 (b) and Fig. 11 (c)).

Let E_i be a *MC* edge and F_j a *MC* face adjacent to E_i, $\forall E_{ref} \in E_i$ and $\forall F_{ref} \in F_j$:

$$Orientation\ (E_i, F_j) = Orientation\ (E_{ref}, F_{ref})$$

4.1.2. Creation of loops of MC faces with minimal interior angle criterion

The boundary of a *MC* face is defined as non-manifold when one *MC* vertex is connected to more than 2 *MC* edges, these *MC* edges being adjacent to this *MC* face. This configuration is not specific to *MCT* as it can be seen in classical B-Rep structures, as shown in Fig. 12.

The AFM requires that the geodesic propagation of fronts, initiated on the loops of a *MC* face, over the underlying composite surface fills completely the interior domain, without front overlaps. Loops are sorted list of co-edges such as the $(i+1)^{th}$ co-edge starts at the i^{th} co-edge's end. In order to avoid front overlaps caused by non-manifold configurations such as the one illustrated in see Fig. 12, the sequence of co-edges along a given loop of a *MC* face is chosen using considerations about the interior angle between two consecutive co-edges (between two alternatives, the one inducing the minimal interior angle is chosen).

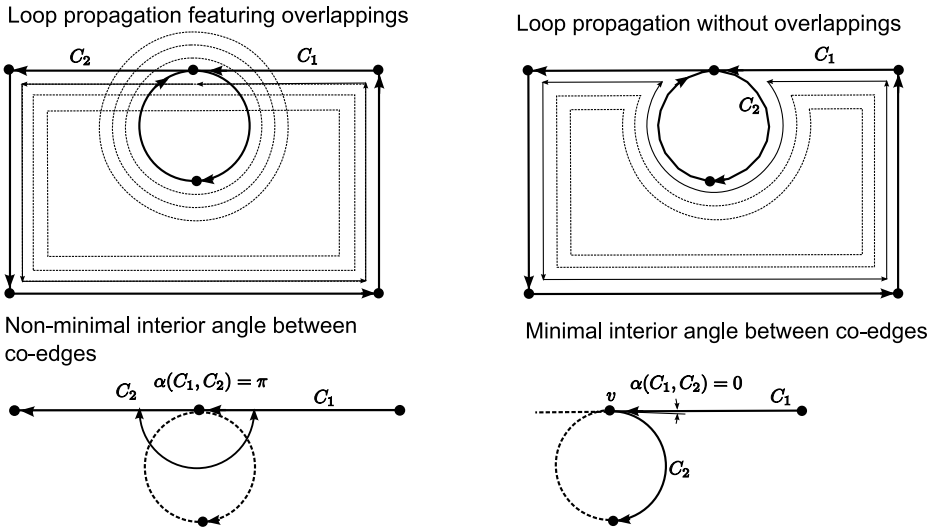


Fig. 12. Loops obtained regarding minimal interior angle criterion

The theoretical definition of interior angles is geodesic: considering an arc of geodesic circle starting from the i^{th} co-edge and ending on the $i+1^{th}$ co-edge, the interior angle is defined as the curvilinear length of arc *s* divided its geodesic radius *r*. Practically, interior angles are calculated from a **tessellation** of the *MCT*, where *MC* faces are approximated by sets of triangles and *MC* edges by sets of segments. The interior angle between two co-edges C_1 and C_2 is computed by summing interior angles of triangles chained between these co-edges and adjacent to their common vertex v (see Fig. 13). This tessellation of the *MCT* is derived from a tessellation of the reference B-Rep topology, which is available in any CAD kernel. The tessellation of the *MCT* is updated after any *MC* operation during *MCT* adaptation: for instance, merging two composite surfaces or curves is followed by merging their discretizations, and splitting a composite curve is followed by inserting a node in its

discretization. The tolerance of this tessellation is set to 1/50th of the minium value in the FE size map. This ensures that the accuracy of geometry definition is much better than the accuracy of the expected FE mesh.

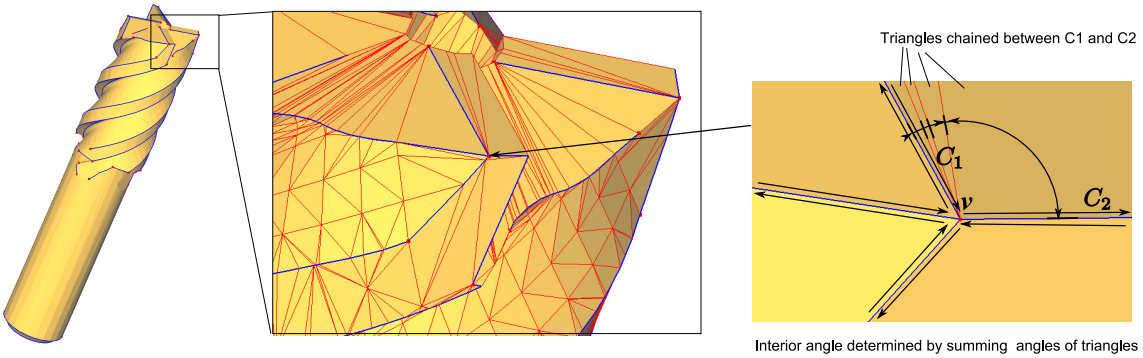


Fig. 13. Interior angle evaluation based on the evaluation of the tessellation of the *MCT*

4.2. Discretization of *MC* edges and initialization of the advancing front

4.2.1. General considerations

Convention:
N(P) *represents the normal vector to a surface at location* ***P****, where* ***P*** *can be an Euclidian point or, a parametric coordinate.*

As introduced in the previous sections, the triangulation of *MC* faces is preceded by the discretization of *MC* edges. This discretization of *MC* edges in mesh segments sets up data required for initializing the advancing front when triangulating *MC* faces. This discretization of *MC* edges and by the way this initialization of the advancing front respects the following properties:

1. The front is composed of segments generated through *MC* edges discretization and these segments are oriented with respect to the orientation of all coedges surrounding the *MC* face. Thus, each front segment is oriented (the orientation of vector $\boldsymbol{P_iP_{i+1}}$) consistently with the advancing front direction, defined as $\boldsymbol{V_i = N(P_M) \wedge P_iP_{i+1}}$, (see Fig. 14 (a) and (b)). As illustrated in Fig. 14 (b) $\boldsymbol{N(P_M)}$ is the normal vector at location $\boldsymbol{P_M}$. A detailed definition of normal vectors is given at section 4.3.
2. Each front segment $\boldsymbol{P_iP_{i+1}}$ only features two neighbours (the previous and next segments). Consequently, the advancing front is composed of one or more closed segment loops that are consistent with the loops of *MC* faces, (see Fig. 14(c))
3. There is no intersection between two segment loops of the front.

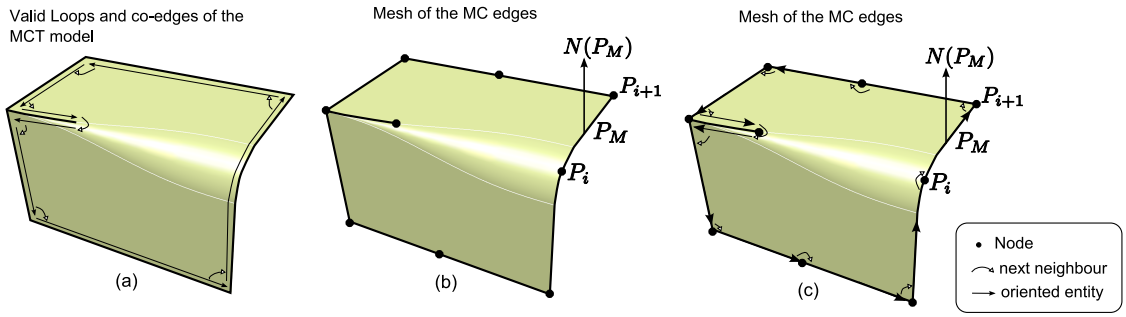


Fig. 14. Front initialization (a) co-edge orientation (b) *MC* edge mesh (c) initial front

Convention:
Variable t designate the curve parameter associated with a reference geometry (t is dependent to the curve's intrinsic mathematical formulation), while variable t' designate the curve parameter associated with a composite geometry (t' equals the curvilinear length along the composite curve).

The generation of the initial front segments is processed as follows. A set of nodes is first generated on *MC vertices and* then nodes are generated along *MC edges*. Given the parametric function of a *Composite curve* ***PolyC***(t') and the size map function $H(x,y,z)$, the method presented in [32] generates an optimal parametric coordinate for each node $\boldsymbol{P_i}$: t'_i with $i = 0,1,..,N_{seg}$, and N_{seg} being the number of segments of the *MC edge* discretization. When all nodes parameters t'_i are defined, the *Composite curve's* mapping function provides, for any t', the corresponding reference edge E and the parametric coordinate t (along this specific reference edge) written as:

$$(E\ ,\ t) = \boldsymbol{PolyC}_{\boldsymbol{Ref}}\,(t')$$

As shown in Fig. 15, each segment is associated with an image curve, which consists of a sequence of sub-segments, each of which representing the image of the mesh segment along a reference edge and on its two adjacent faces.

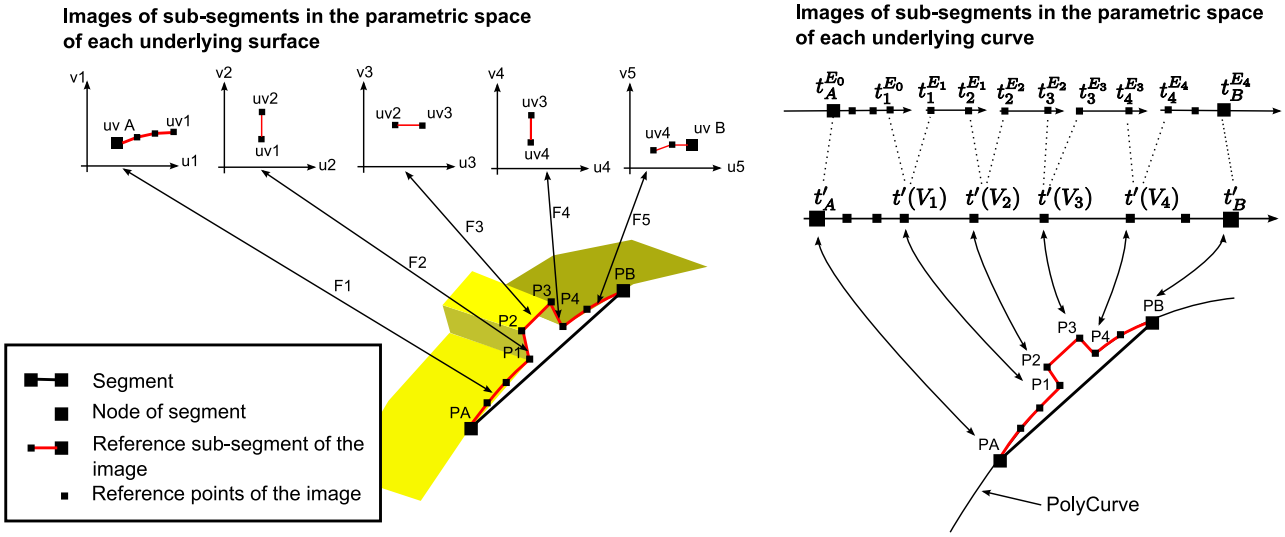


Fig. 15. Images of a segment lying on multiple reference edges and faces

4.2.2. Front initialization for specific non-manifold cases

As mentioned previously, *MCT* transformations naturally generate non-manifold boundary topologies. Fig. 16 illustrates two of such specific non-manifold configurations, one corresponding to a *MC* edge or a group of *MC* edges that are isolated on a *MC* face, and the other corresponding to a *MC* vertex that is isolated on a *MC* face. In the first case, the initial front segments associated with isolated *MC* edges are doubled (with inverse orientations) so that a closed loop of front segments allows it to progress in both inverse directions (for example at location $\boldsymbol{P_M}$ in Fig. 16). In the case of an isolated *MC* vertex, an additional front segment is automatically generated from the isolated vertex and, like in the case of isolated *MC* edges, it is doubled so that the front is able to progress in both directions. This specific

front segment is generated arbitrarily except that it should not cross any other front segment and not be located too close to an existing front node (see Fig. 17).

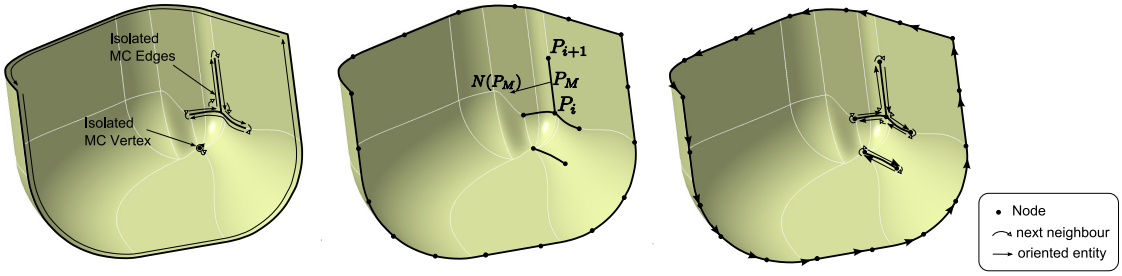


Fig. 16. Two specific cases of front initialization

A case involving an isolated MC vertex is presented in Fig. 17a and in Fig. 17b the arbitrary front segment mentioned above is generated in such a way that the front node $\boldsymbol{P_c}$ is too close to other front segments. As illustrated in Fig. 17b, a search zone is used around candidate front segment $\boldsymbol{MP_c}$ and if an existing front node is found in this zone, it is used instead of $\boldsymbol{P_c}$. In this case (see Fig. 17c), the advancing front is updated as $[\boldsymbol{P_{i-1}P_i}]$, $[\boldsymbol{P_iM}]$, $[\boldsymbol{MP_i}]$, $[\boldsymbol{P_iP_{i+1}}]$, etc.

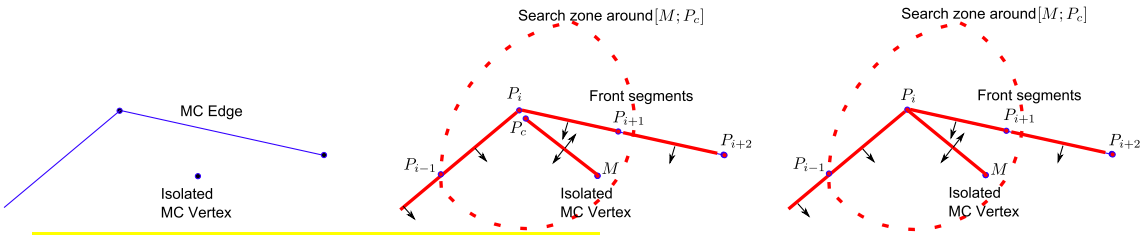


Fig. 17. Front initialization for an isolated MC vertex

4.3. Normal vector evaluation at the location of first-order discontinuities

As mentioned in the previous section, the evaluation of normal vectors is crucial to compute the advancing front propagation direction when triangulating *MC* faces:

- the front configurations 1 to 5 are validated if normal vectors of candidate triangles generated are consistent with that of normal vectors at midpoint $\boldsymbol{P_M}$ (see Fig. 18 (a))
- the front configuration 6 intends to create the optimal point $\boldsymbol{P_{opt}}$ by advancing towards a direction based on the composite surface's normal (see Fig. 18 (b))

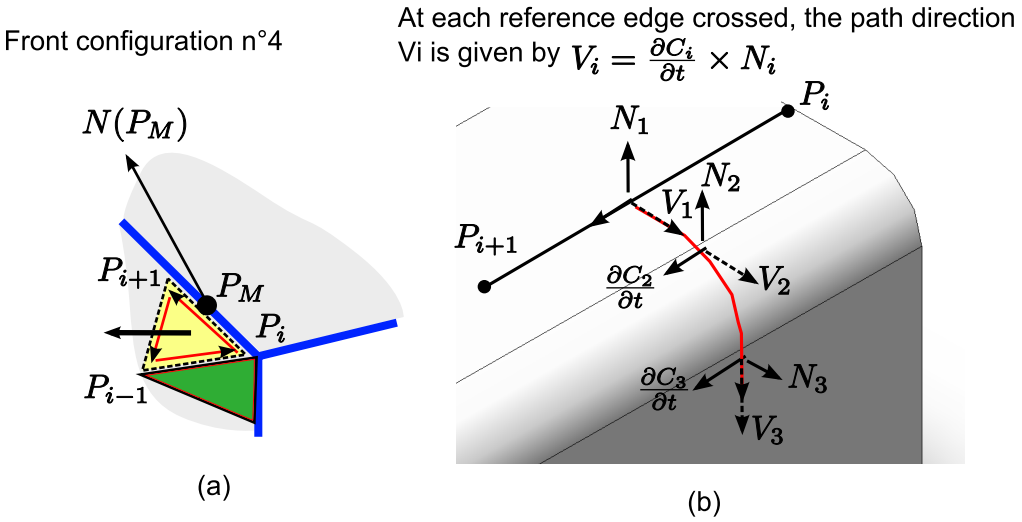


Fig. 18. Normal vector evaluation requirements at discontinuities (a) in front configuration #4, (b) in front configuration #6

In the context of composite surfaces, some of the normal vectors have to be evaluated at locations where the geometry is non-differentiable (sharp edges and corners featuring C^0 continuity), as well as at location where the domain is at least C^1 continuous (in curved regions). Like in the case of smooth rendering methods, the weighted average of normal vectors at adjacent faces smoothes the surface at points located on sharp edges and vertices of a *Composite surface*. In this work, the weights used for the computation of weighted average of normal vectors at location of sharp vertex are normalized interior angles β_j of faces F_j adjacent to point $\boldsymbol{P_i}$ (see Fig. 19a). Consequently, the equivalent normal vector $N(P_i)$ at location P_i is given by:

$$N(P_i) = \frac{\sum_{j=1}^{n} \beta_j \cdot N(F_j, P_i)}{\sum_{j=1}^{n} \beta_j}$$

where $\boldsymbol{N(F_j, P_i)}$ is the normal vector to reference face $\boldsymbol{F_j}$ at location $\boldsymbol{P_i}$

For the calculation of an equivalent normal vector along a sharp edge (see Fig. 19b), this expression simply becomes:

$N(P_i) = \frac{N_3 + N_4}{2}$ where N_3 and N_4 are the normal vectors to the two intersecting reference faces at location $\boldsymbol{P_i}$

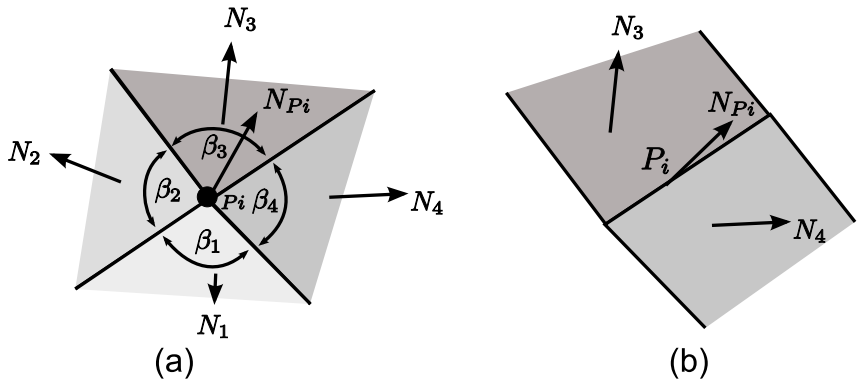


Fig. 19. Interior angle and normal vectors along sharp edges and sharp vertices

4.4. Propagation of the advancing front

4.4.1. Calculating the optimal location of a candidate node

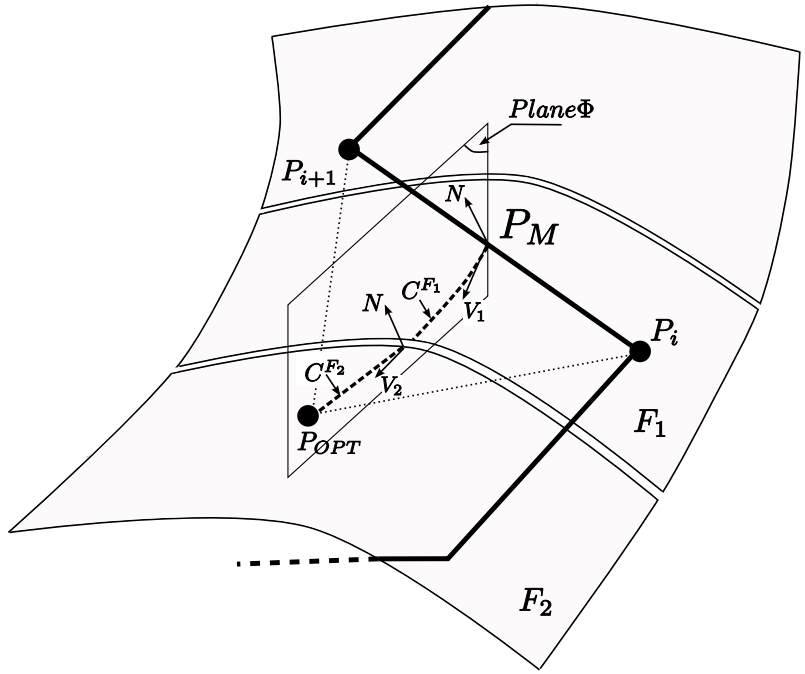


Fig. 20. The optimal location of candidate node P_{OPT} from candidate segment P_iP_{i+1} is computed using a path *composed of curves C^{Fj} (lying on the underlying reference faces F_j) intersecting the composite surface with plane Φ*

Conventions

$\sigma_j(u,v)$ represents the parametric surface equation underlying reference face F_j.

$C_{uv}^{Fj}(t)$ represents a 2D curve, defined in the parametric space of reference face F_j.

The 3D equation of $C_{uv}^{Fj}(t)$ is given by: $C^{Fj}(t) = \sigma_j\left(C_{uv}^{Fj}(t)\right)$. By the way, $C^{Fj}(t)$ lies on the surface defined by $\sigma_j(u,v)$.

The most important obstacle to adapting the AFM to composite surfaces is adapting the method in the case of front configuration #6 (as defined in Fig. 5). Indeed, given a candidate front segment P_iP_{i+1}, the problem is finding the optimal location of candidate node P_{opt} from segment P_iP_{i+1} to obtain a triangle respecting both size and shape quality requirements. In the

classical implementation of the AFM, the optimal location is basically estimated with the use of a trajectory, defined in the parametric space of the surface, which starts from midpoint $\boldsymbol{P_M}$, and which progresses towards the advancing front direction, until matching a target distance d. The adaptation of this trajectory in the context of composite surface meshing is performed using the 3D intersection curve between a plane (referred to in our work as $\boldsymbol{\Phi}$) and the composite surface. This trajectory is computed as follows:

- Midpoint $\boldsymbol{P_M}$, lies on the curvilinear image of the candidate front segment and it is equidistant to $\boldsymbol{P_i}$ and $\boldsymbol{P_{i+1}}$ (using the Euclidian distance),
- Plane $\boldsymbol{\Phi}$ is computed as the equidistant plane to $\boldsymbol{P_i}$ and $\boldsymbol{P_{i+1}}$
- The advancing front direction $\boldsymbol{V_i}$ is $\boldsymbol{V_i = N(P_M) \wedge P_iP_{i+1}}$
- Like in the classical implementation of the AFM (see section 3), the target distance d is calculated as a compromise between shape and size requirements

$d = \sqrt{3}/2 \cdot \left(w_T \cdot H(P_M) + w_s \cdot \left\| P_i P_{i+1} \right\| \right)$ where $\boldsymbol{w_T} \in [0; 1]$ and $\boldsymbol{w_T + w_s = 1}$

Then, the 3D intersection curve is built as follows. Midpoint $\boldsymbol{P_M}$ is obtained by finding intersection points between $\boldsymbol{\Phi}$ and the segment's curvilinear image (see section 4.2). Then, the trajectory $\boldsymbol{C}(t')$, which is defined as the intersection curve between $\boldsymbol{\Phi}$ and the composite surface $\boldsymbol{\sigma}$ is obtained by creating a sequence of curves $\boldsymbol{C}^{Fj}(t)$ that are defined in the parametric space of reference faces, and that are coincident to $\boldsymbol{\Phi}$ with a given tolerance δ_{max}. The first curve $\boldsymbol{C}^{F0}(t)$ starts from $\boldsymbol{P_M}$ and stops either if the target distance d is reached, or if the boundary of the current reference face. New curves $\boldsymbol{C}^{Fj}(t)$ are appended to the trajectory $\boldsymbol{C}(t')$ until the target distance d is reached or the MC face boundary is met (see Fig. 20 and Fig. 21). Each curve $\boldsymbol{C}^{Fj}(t)$ along the intersection between reference faces $\boldsymbol{F_j}$ and $\boldsymbol{\Phi}$ is computed in the parametric space associated to face $\boldsymbol{F_j}$ as a polyline (referred to as $\boldsymbol{C}_{uv}^{Fj}(t)$).

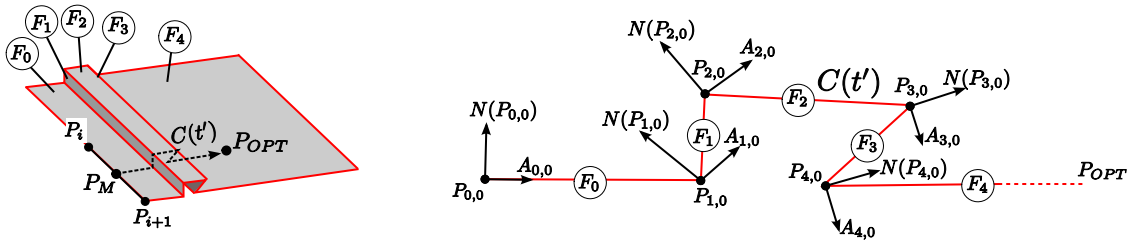


Fig. 21. The *Composite surface* path $C(t')$ is constituted by multiple curves lying on reference faces of the *Composite surface*

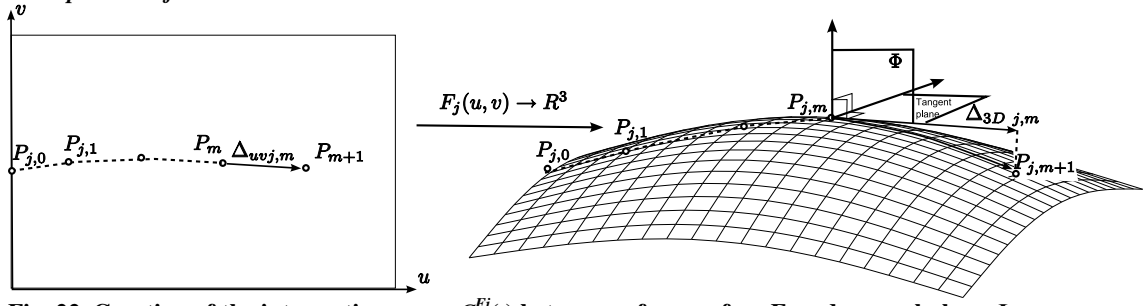


Fig. 22. Creation of the intersection curve $C^{Fj}(t)$ between reference face F_j and normal plane Φ

The overall trajectory $\boldsymbol{C(t')}$ is created by concatenating the curves $\boldsymbol{C}^{Fj}(t)$ until target length d is reached, which means that $\left\| C(t') \right\| = \sum_{j=0}^{n-1} \left\| C^{Fj}(t) \right\| = d$.

Consequently, the curvilinear length of $\boldsymbol{C}^{Fj}(t)$ is limited by $L^{Fj}{}_{max}$, which is the difference between target length d and length of the curves preceding $\boldsymbol{C}^{Fj}(t)$: $L^{Fj}{}_{max} = d - \sum_{k=0}^{j-1} \left\| C^{Fk}(t) \right\|$. The

starting point of the first curve, referred to as $\boldsymbol{P_{0,0}}$, is located at midpoint $\boldsymbol{P_M}$. The path creation consists of a concatenation of parametric sub-segments $\boldsymbol{P_{j,m}P_{j,m+1}}$ along the intersection between $\boldsymbol{\Phi}$ and the surface (see Fig. 22). The length of sub-segments, referred to as *ds*, is small enough to locally approximate the intersection curve by a straight line segment in the parametric space. In practice, we found that $ds = \frac{d}{50}$ is small enough to reach accuracy $\delta_{\max}$ with $\delta_{\max} \approx \frac{d}{500}$.

Each sub-segment $\boldsymbol{P_{j,m}P_{j,m+1}}$ is obtained as follows:

- Compute (see Fig. 22) the 3D increment vector $\Delta_{3D\ j,m}$ (length *ds* and collinear to the intersection between $\boldsymbol{\Phi}$ and the plane that is tangent to F_j at location $\boldsymbol{P_{j,m}}$).
- Calculate the parametric increment $\boldsymbol{\Delta_{uv\ j,m}}$ by projecting $\Delta_{3D\ j,m}$ in the parametric space associated with face F_j (with accuracy δ_{max}).
- Append a new point $\boldsymbol{P_{j,m+1}}$ to polyline $C^{Fj}(t)$: $\boldsymbol{P_{j,m+1}} = \boldsymbol{P_{j,m}} + \Delta_{\mathrm{uv}\,\boldsymbol{j,m}}$.

Following this last process, the number of parametric points constituting polyline $C^{Fj}(t)$ is $\frac{L^{Fj}_{\max}}{ds}+1$. It appears that fewer points are actually required to respect δ_{max} and in order to reduce the number of polyline points, an additional procedure is applied during path creation, with the objective of keeping the minimum number of points that are necessary to respect δ_{max}.

4.5. Validation of triangle creation

4.5.1. New requirements for detecting collisions in the AFM

In the classical AFM scheme, each candidate triangle creation is analysed before being validated to avoid triangle overlaps. As mentioned above, the extension of this analysis in the case of composite surfaces requires dealing with the fact that each mesh segment and mesh triangle generated through the AFM progression on composite geometry is likely to lie on multiple B-Rep faces and is, by the way, likely to be associated with a composite definition across multiple parametric spaces. Consequently, collision tests between new front segments and existing mesh elements are much more difficult to perform than in standard AFM and these tests require setting up specific procedures as described in the following paragraphs.

4.5.2. Computing the image of a segment on a composite surface

Detecting front collisions in the validation of triangle creation in the context of composite surfaces requires computing the curvilinear image of front segments on a *composite surface*. Constructing the curvilinear image of a segment $\boldsymbol{P_iP_{i+1}}$ is achieved by one of the following schemes:

- The first attempt (see section 4.5.3) consists of projecting $\boldsymbol{P_iP_{i+1}}$ in the normal direction on the underlying *Composite surface* $\boldsymbol{\sigma}$. This method is likely to fail when the normal projection is composed of several disconnected components (see Fig. 23).
- If this first attempt fails, the shortest path between $\boldsymbol{P_i}$ and $\boldsymbol{P_{i+1}}$ is evaluated with a method that is detailed in section 4.5.4.

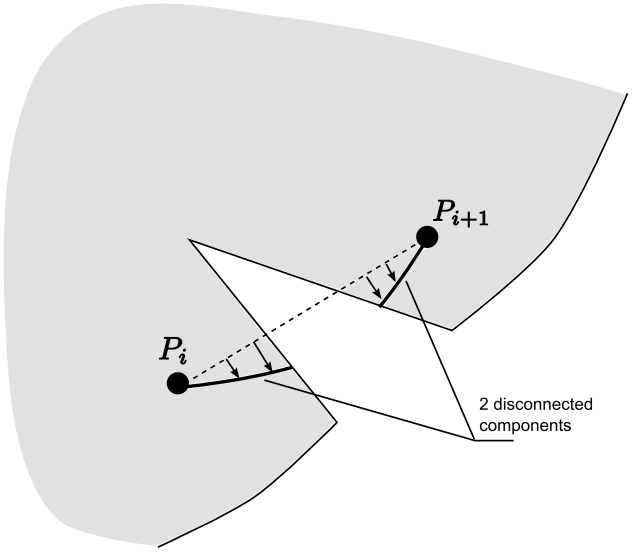


Fig. 23. Example of segments' normal projection composed of 2 disconnected components.

4.5.3. Normal projection representing the image of a segment on a composite surface

As introduced in the previous section, the normal projection method applies when the curvilinear image is composed of one single component, which is true in most of front generation situations. The projection is calculated as a plane-surface intersection, which is similar to the intersection used for creating a trajectory that is perpendicular to the front (as detailed in section 4.4.1). The intersecting plane $\mathbf{\Phi_N}$ is computed as follows:

- Calculate normal vector $\boldsymbol{N}$ by averaging normal vectors at $\boldsymbol{P_i}$ and $\boldsymbol{P_{i+1}}$ (see Fig. 23).
- $\mathbf{\Phi_N}$ is defined as passing through to $\boldsymbol{P_i}$ and $\boldsymbol{P_{i+1}}$ and as collinear to vector $\boldsymbol{N}$.

Then, curvilinear image of front segment $\boldsymbol{P_iP_{i+1}}$ is generated by computing a trajectory, referred to as $\boldsymbol{C}(t')$, at the intersection between $\mathbf{\Phi_N}$ and the c*omposite surface* $\boldsymbol{\sigma}$. This trajectory (from $\boldsymbol{P_i}$ to $\boldsymbol{P_{i+1}}$) is obtained by creating a sequence of curves $\boldsymbol{C}^{Fj}(t)$ that are defined in the parametric space of reference faces, and that are coincident to $\mathbf{\Phi_N}$ with a given tolerance δ_{max}. The first curve $\boldsymbol{C}^{F1}(t)$ starts from $\boldsymbol{P_i}$ and stops either if $\boldsymbol{P_{i+1}}$ is reached, or if the boundary of the current reference face is reached. New curves $\boldsymbol{C}^{Fj}(t)$ are appended to continue the trajectory $\boldsymbol{C}(t')$ until $\boldsymbol{P_{i+1}}$ is reached. (see Fig. 24). Thus, each curve $\boldsymbol{C}^{Fj}(t)$ is constrained by $\mathbf{\Phi_N}$ and $\boldsymbol{F^j}$.

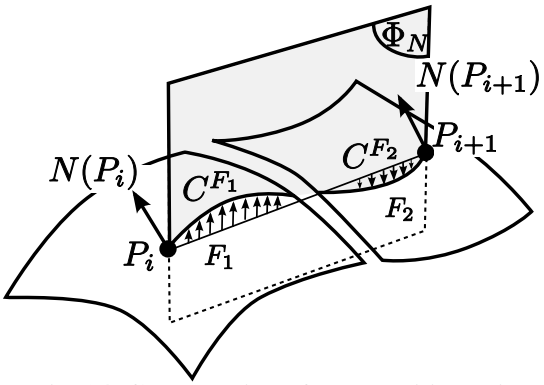


Fig. 24. Construction of the curvilinear image of a segment on its *MC Face* ($\mathbf{F_1}$, $\mathbf{F_2}$)

4.5.4. Shortest path estimation on composite surfaces

Composite surfaces singularities such as sharp edges and thin walls require a specific shortest path evaluation in the following cases:

- When the method presented in section 4.5.3 is affected by surface singularities (see figure Fig. 25), the shortest path estimation provides a valid representation of the segment's image.
- When validating element creation by searching neighboring elements, the shortest path evaluation is a good estimator of the curvilinear distance in zones featuring singularities (see figure Fig. 25).

The shortest path between two points $\boldsymbol{P_B}$ and $\boldsymbol{P_C}$ is computed following these steps:

Step 1: A set of passing points $\boldsymbol{P_k}$ (see Fig. 25) is defined and it is composed with:

- $\boldsymbol{P_B}$ and $\boldsymbol{P_C}$, the beginning and the end of the shortest path,
- $\boldsymbol{P_{ke}}$ defined as points lying on reference edges of the composite surface and located in the neighborhood of linear segment $\boldsymbol{P_B P_C}$, which means such as
  $dist(\boldsymbol{P_B}\, \boldsymbol{P_C}, \boldsymbol{P_{ke}}) < \|P_B P_C\|$
  where $dist(\boldsymbol{P_B}\, \boldsymbol{P_C}, \boldsymbol{P_{ke}})$ is the distance between the linear segment $\boldsymbol{P_B}\, \boldsymbol{P_C}$ and $\boldsymbol{P_{ke}}$;
- $\boldsymbol{P_{kv}}$ defined as coincident points with reference vertices of reference faces and located in the neighborhood of linear segment $\boldsymbol{P_B P_C}$, which means such as
  $dist(\boldsymbol{P_B}\, \boldsymbol{P_C}, \boldsymbol{P_{kv}}) < \|P_B P_C\|$

Step 2: The set of curves $\boldsymbol{C_i}$ connecting two passing points $\boldsymbol{P_k}$ located on a reference face or edge $\boldsymbol{G}$ of the composite surface,

Step 3: The point-curve adjacency graph is constructed as following:

- Each graph node represents a passing point $\boldsymbol{P_k}$,
- Each graph arc represents a curve $\boldsymbol{C_i}$ connecting two passing points,

Step 4: Dijkstra's algorithm [44] is used to compute the shortest path through curves (see Fig. 25):

- In Dijkstra's algorithm the lengths associated with graph arcs are evaluated by calculating the lengths of curves $\boldsymbol{C_i}$.
- The length of the path is the sum of the curves' length attached to arcs.

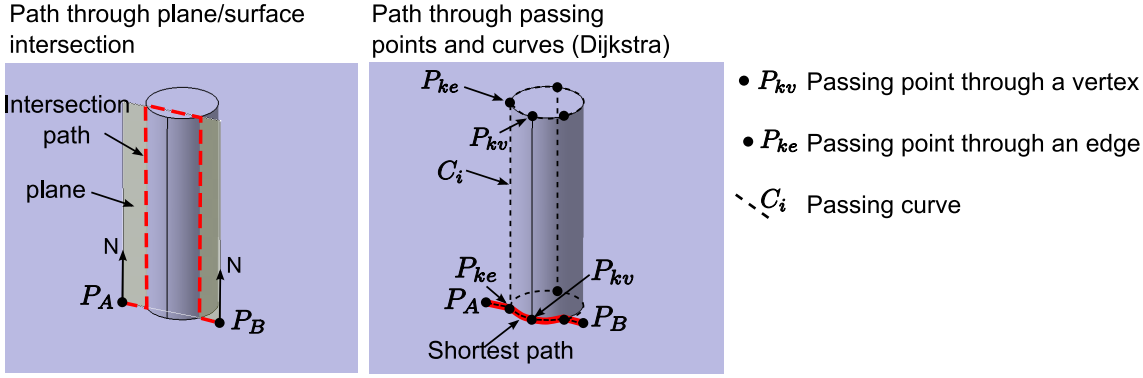


Fig. 25. Shortest path passing through points P_k and curves C_i

4.5.5. Calculating intersection points between segments

As mentioned in the previous section, the validation of triangle creation mainly relies on detecting collisions between mesh segments (the sides of newly created triangles) and front segments. As illustrated in fig. 26 the intersection between two segments is tested by

evaluating the intersection of their curvilinear images in the parametric space of the associated reference faces.

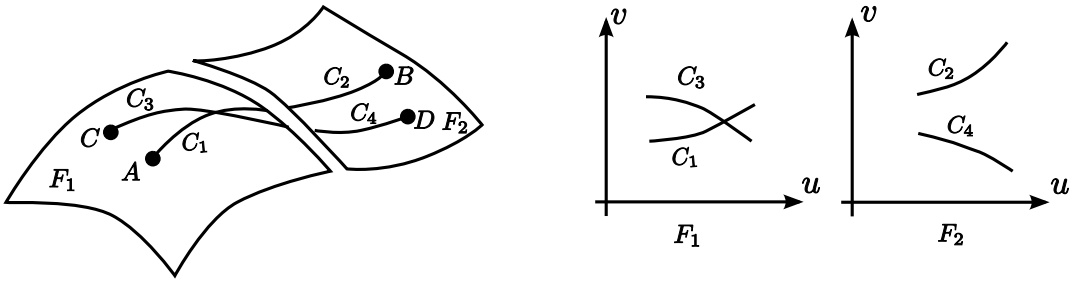


Fig. 26. Segments *AB* and *CD* are intersecting on reference face F_1

4.6. Updating the front and ending the process

Once a triangle creation is validated, the advancing front needs to be updated. The adaptation of front update operations to composite surfaces is straightforward because it is identical to the process used for standard AFM. Also, like in the classical implementation of the AFM, the front progression stops when the front is empty, which means that the whole composite surfaces being triangulated is completely filled with triangles. The application of this adapted version of the AFM to all *MC* faces in the *MCT* model leads to a closed and watertight set of triangles representing the triangulation of the *MCT* model's boundary. Once this triangulation generated, a 3D solid mesh can be derived using any automatic tetrahedral mesh generation scheme.

5. Examples and results

The algorithms described in this paper are validated by showing the improvement obtained, on mesh size and shape quality criteria, when using the *MCT* approach if compared to meshes that are directly generated from unprepared CAD models. An analysis of quality improvement shows that mesh quality obtained, after applying the *MCT* preparation process followed by the automatic triangulation process as presented in this paper, is close to optimal. This clearly validates the approach by showing that, whatever the topology of the initial CAD model is, the whole process is successful in avoiding the generation of badly shaped elements or over-refined meshes.

Fig. 27 illustrates the triangle quality distributions obtained for the meshes presented in Fig. 28, Fig. 30, Fig. 31 and Fig. 32. Shape quality is evaluated using a classical criterion (radius of the inscribed circle divided by the maximum edge length) and size quality is evaluated using the geodesic size criterion [5] which quantifies the difference between the actual area of a triangle and its optimal area with regard to the imposed size map.

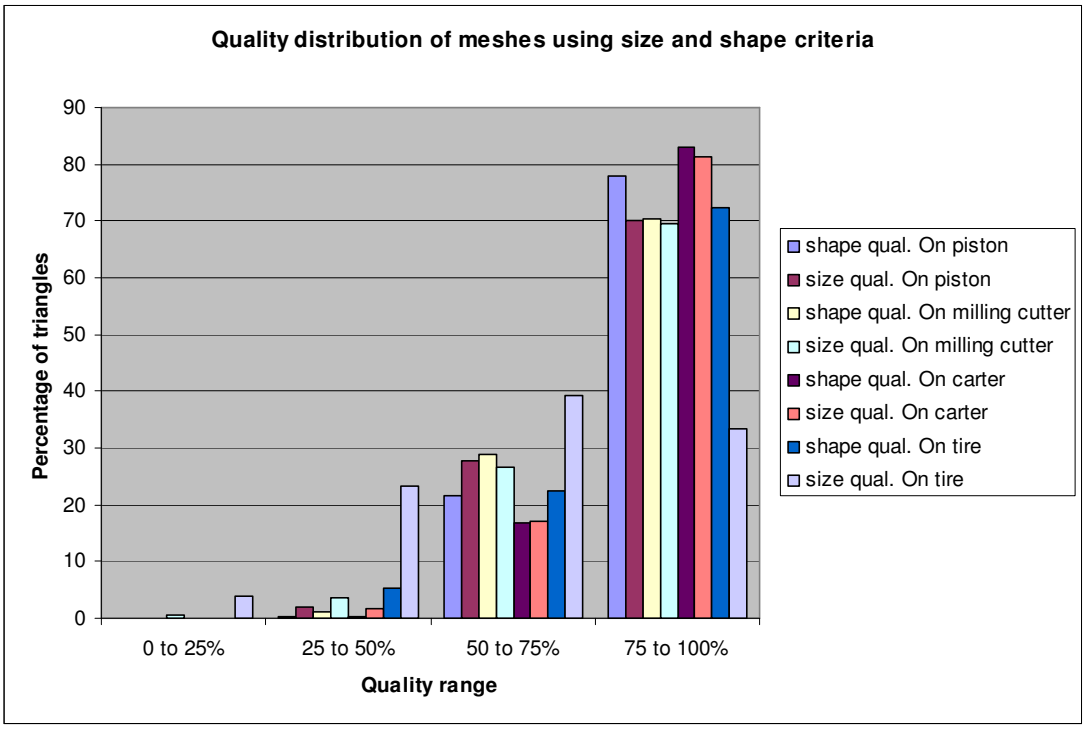


Fig. 27. Shape and size quality distribution of meshes obtained on adapted *MCT* models.

Quarter of a piston : Fig. 28 shows one quarter (due to symmetry) of a piston's CAD model. The original CAD model features many narrow faces and small edges that are irrelevant for mesh generation. On this sample part, the *MCT* simplification process [32] operated 60 edges deletions, 62 vertex deletions, and collapsed one edge to a vertex. By the way, the number of faces has been reduced from 71 to 21, the number of edges from 182 to 76, and the number of vertices from 113 to 63. Size and shape quality distributions are satisfying as illustrated in Fig. 28. Fig. 28 also illustrates that the use of the method presented in this paper on this example completely eliminates poor quality triangles (for both shape and size).

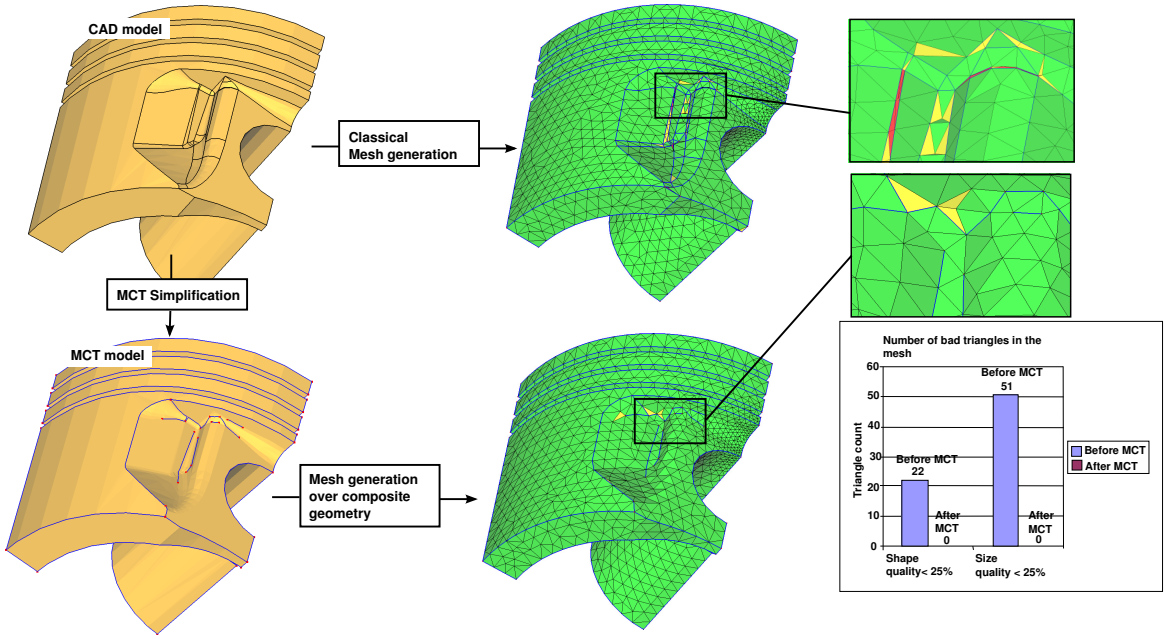


Fig. 28. Quarter of a piston: comparison between the meshes generated on the *MCT* model by composite surface meshing and on the initial (without preparation) CAD model.

In Fig. 28, a constant FE size is imposed while in Fig. 29 the FE size required varies. In this last figure, element colours refer to the imposed element size (instead of element quality) as indicated in the colour legend. This illustrates that the method proposed in this paper can handle both constant and varying FE maps of size.

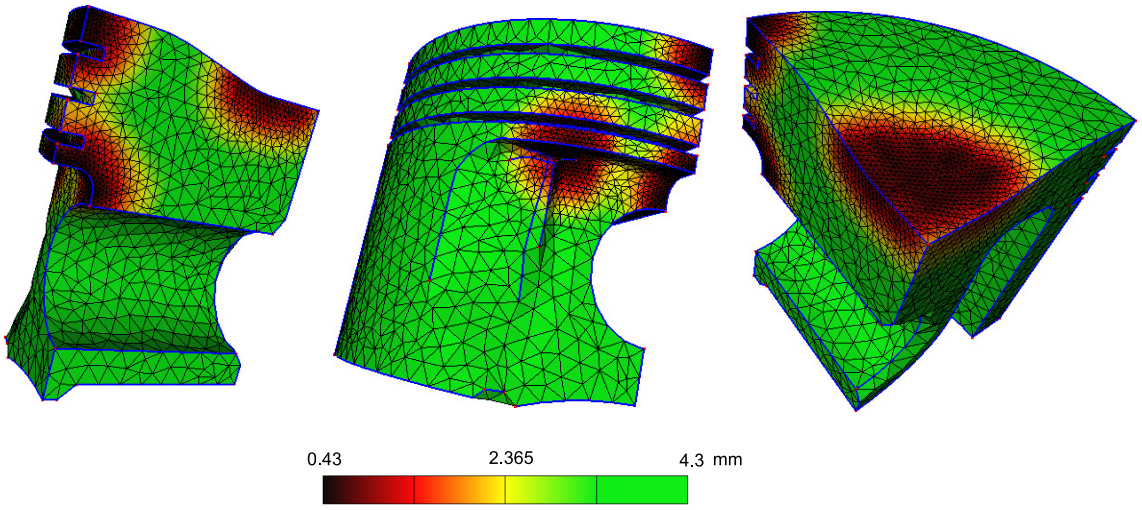


Fig. 29. Quarter of a piston: mesh generated on the *MCT* model by composite surface meshing with a non-constant FE map of size.

Milling cutter: results presented in Fig. 30 represent the initial CAD model aside a *MCT* model automatically generated using our adaptation process. The B-Rep entities representing the cutter size label and other irrelevant edges have been removed. Fig. 30 shows that shape and size quality distributions are both significantly improved with the use of the approach presented in this paper. In this case 7 triangles with a shape quality inferior to 25 % remain in the final triangulation. It is important to outline that, for the size map imposed, the badly shaped triangles remaining cannot be eliminated. This is due to the fact that the geometry

itself still over constrains mesh generation, even once the *MCT* preparation process is applied. In this case, the elimination of the few badly shaped triangles remaining could only be achieved using a local mesh refinement.

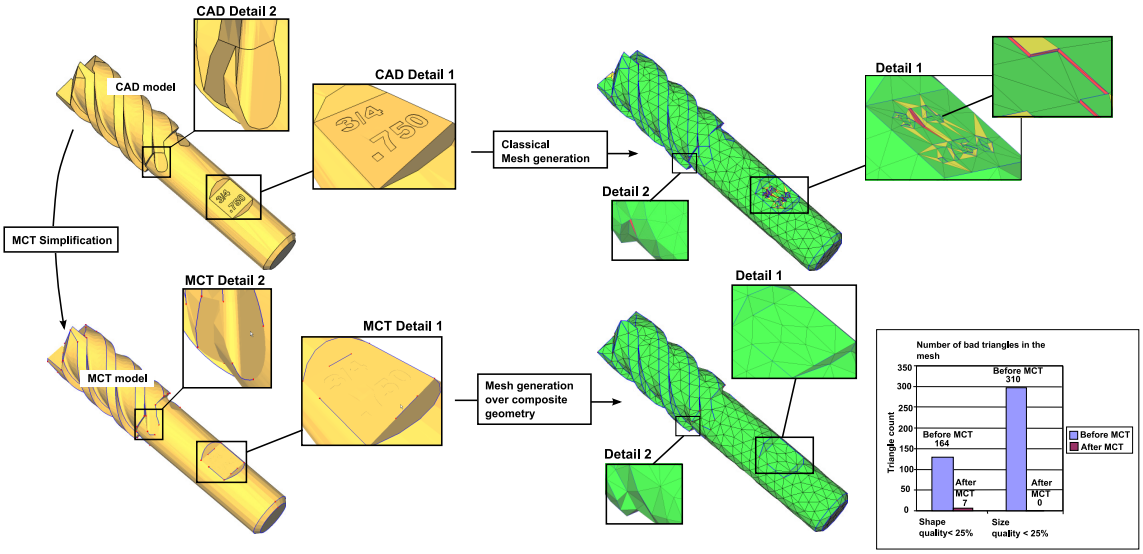


Fig. 30. Milling cutter: comparison between the meshes generated on the *MCT* model by composite surface meshing and on the initial (without preparation) CAD model.

Carter: Fig. 31 illustrates a more complex CAD model (a carter). The *MCT* simplification process operated 540 edge deletions, 497 vertex deletions, and collapsed 7 edges. In this case, the number of faces has been reduced from 520 to 71, the number of edges from 1256 to 301, and the number of vertices from 785 to 279.

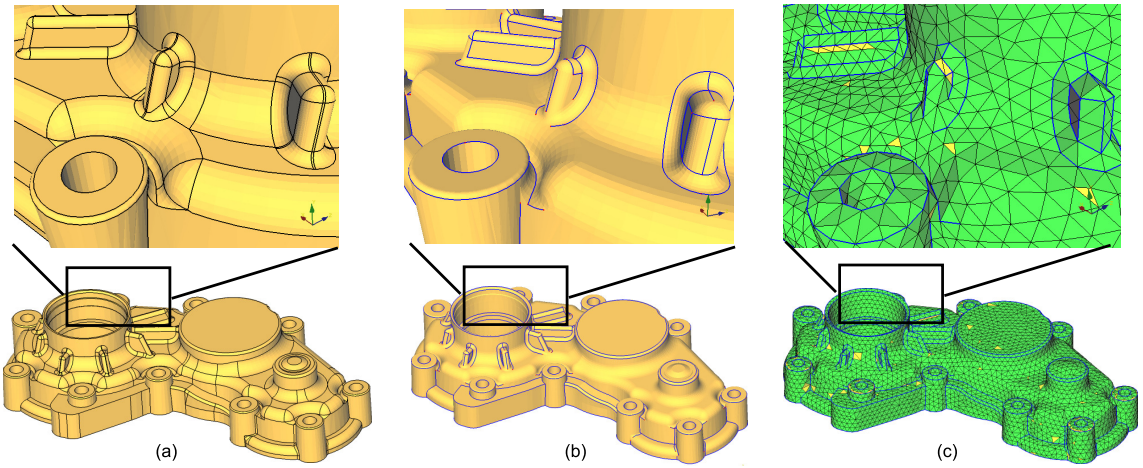


Fig. 31. Carter: (a) initial topology (b) topology obtained after applying the *MCT* process (c) mesh generated on the *MCT* model by composite surface meshing.

Tire: This last example is issued from a CAD model generated for simulating the heating process of a tire section (courtesy Michelin™). The *MCT* simplification process operated 185 edge deletions, 283 vertex deletions, and collapsed 1 edge. The number of faces has been reduced from 183 to 11, the number of edges from 569 to 134, and the number of vertices from 388 to 136.

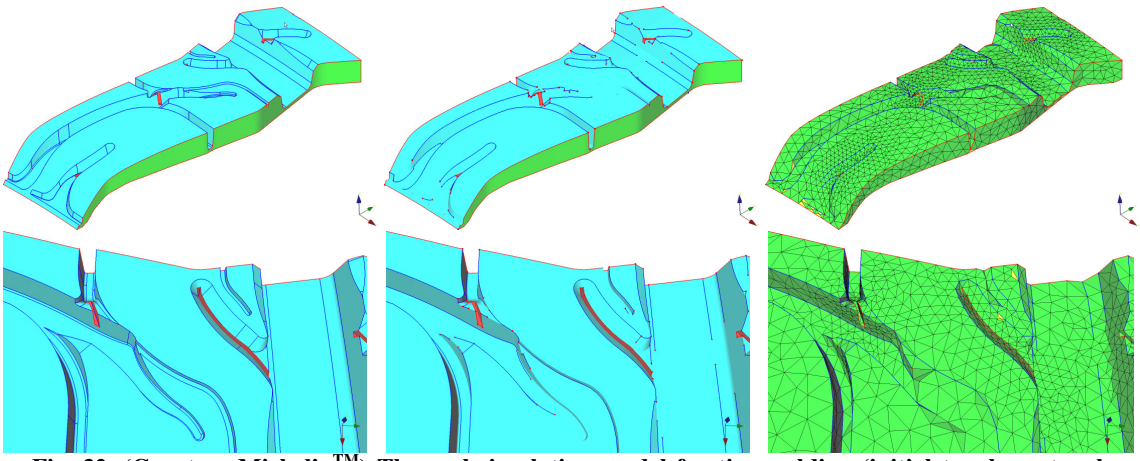


Fig. 32. (Courtesy Michelin™) Thermal simulation model for tire molding (initial topology, topology adapted with the *MCT* process and final mesh)

5.1. Discussion with regard to closed surfaces and stretched geometries

One of the most interesting aspects of the work presented in this paper is that using composite parametric definitions avoids using a global parameterization of composite surfaces. This is an important improvement because it overcomes the main limitations of approaches based on mapping composite surfaces into a single parametric domain.

Thus, as illustrated in Fig. 33, the AFM adaptation presented in this paper is:

- *not restricted to specific boundary topologies*: closed composite surfaces that are homeomorphic to a sphere or to a n-torus can be meshed as well as open composite surfaces.
- *not limited by stretched geometries and steep metrics variations*: highly stretched surfaces, such as the one representing a glove inspired shape, are handled easily and efficiently and it results in very good quality elements, even in high curvature zones. In fact, with the approach presented here, these high curvature zones are handled as easily as when using the AFM on single surfaces.

6. Conclusion

This paper presents an extension of the advancing front method to surfaces composed of multiple parametric faces, avoiding the construction of a global parameterization and by the way, overcoming the limitations of this type of approaches. This extension is intended to be used in the scope of a *MCT* adaptation procedure aimed at improving and automating the preparation of FEA models from CAD models. Unlike previous re-parameterization based schemes, the proposed method has no limitations concerning the type and topology of composite surfaces involved.

However, the approach proposed shows some limitations when applied to over-featured CAD models. For example, location of the candidate point when calculated according to Fig. 20 can be inadequate in some specific configurations (in the case of a twisted intersection curve for example). In general, the process presented in this paper requires prior feature

removal when applied to such CAD models. We have set up automatic feature removal tools that are not presented in this paper but a failure in this feature-removal preparation step often cause the presence of small features disturbing the convergence of advancing front mesh generation. The ideal feature-removal algorithm would transform the initial model into a model that conforms to the specified size map. Further work could overcome this weakness by improving the robustness of feature-removal and advancing front mesh generation processes.

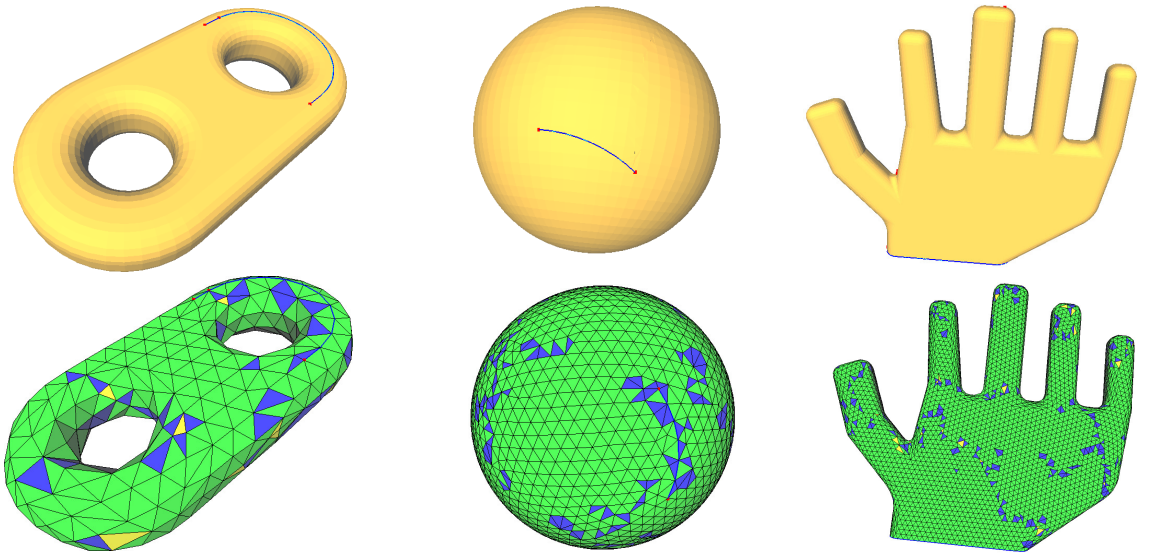


Fig. 33. Models after topology adaptation and resulting triangulations, obtained on closed composite surfaces. For each example, the remaining *MC edge* (in blue for each CAD model) has been used for the advancing front initialization. For these 3 types of configurations, approaches based on a mapping into a unique parameterization would fail.

CPU time is also a matter of potential improvement. Indeed, processing geometries like those presented in this paper usually takes more than one minute to be fulfilled. The classical AFM is indeed known as being quite slower than Delaunay based methods and the extension proposed in this paper is slower that the classical AFM for obvious reasons (computing trajectories over multiple surface patches is iterative). However, the overall quantitative benefits of using this method are also obvious. Indeed, it significantly reduces CPU time dedicated to the FEA itself and in some cases it allows meshing geometries that could not even be meshed without adapting boundary topology. Moreover, at this point of our research, no specific effort has been put on optimizing CPU time, neither for algorithms themselves nor for data structures used, which makes that potential gains are likely to be made. To reduce CPU time, a possible enhancement would be parrallelizing the method by simultaneously progressing on several *MC Edges* when generating the initial front and then simultaneously progressing on several *MC Faces* with our adapted AFM.

Along with improving feature removal procedures and CPU time, the future potential directions in this research include:

Extension to high order elements, quadrangles and anisotropy: quadratic triangles (T6) can be easily generated on the exact geometry by inserting a middle node on each segment's image curve. This method should include a quality criterion to avoid squeezed elements in highly curved zones. The segment-swapping and node moving optimization steps will require quality

criteria that are adapted to curvilinear mesh elements. An adaptation of our method could also be foreseen towards generating quadrangular meshes. The MCT adaptation and AFM mesh generation could also be extended for anisotropic mesh generation. For instance, the criteria used for shape detail identification could be extended to compare the width of shape details with the expected FE size in the direction of element generation, as well as the size criteria used for the MCT adaptation.

Extension to surface models with overlaps and gaps: at the intersection between two B-Rep faces there is a very small numerical inconsistency that cannot be avoided because there is a tiny difference between the actual intersection curve, the 3D parametric definition of the intersection curve in the B-Rep and the images of this intersection curves lying on both intersecting surfaces. This numerical inconsistency is handled in all our developments involving B-Reps and by the way, if the B-Rep is consistent it is necessarily watertight. The other possible context is a model with two connecting trimmed patches featuring an actual gap or overlap at the interface (due to CAD format conversion issues for example). This type of configuration (which cannot be considered as part of a consistent B-Rep) is not handled in our work and handling it would require significant adaptations of the method presented. However, this could be an interesting extension.

Extension to mixed-dimensional models: extending feature-removal and *MCT* preparation algorithms to 3D geometric models mixing curves (meshed with beam elements), surfaces (meshed with shell elements), and solids (meshed with solid elements).

Extension to assembly models: even if the method proposed in this paper is not primarily intended for being applied on assembly models, it appears that it actually is one potential significant enhancement. Overall, applying our method to assembly models could be achieved by applying the following steps:

- Detect faces that are in contact between B-Rep models associated with the assembly components.
- Similar to what is done in [45], pre-process B-Rep models in the assembly with an imprinting operator [46] for splitting B-Rep faces that are in contact.
- Generate the MCT for each B-Rep in the assembly: specific processing should be applied so that vertices and edges lying on faces that are in contact are preserved.
- Meshing each B-Rep in the assembly while taking into account faces that are in contact: along with the previous operators, this guarantees that meshes are conformal at interfaces between parts in contact.

Acknowledgements

This study was carried out as part of a project supported by research funding from Québec Nature and Technology Research Fund and by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Halpern, M., *Industrial requirements and practices in finite element meshing: a survey of trends*, in *Proceedings of the 6th International Meshing Roundtable*. 1997: Park City.

2. Borouchaki, H., et al., *Surface mesh enhancement with geometric singularities identification.* Computer Methods in Applied Mechanics and Engineering, 2005. **194**(48-49): p. 4885-4894.
3. Sheng, X. and B.E. Hirsh, *Triangulation of trimmed surfaces in parametric space.* Computer Aided Design, 1992. **24**(8): p. 437-444.
4. Bechet, E., J.-C. Cuilliere, and F. Trochu, *Generation of a finite element MESH from stereolithography (STL) files.* Computer Aided Design, 2002. **34**: p. 1-17.
5. Cuilliere, J.-C., *An adaptive method for the automatic triangulation of 3D parametric surfaces.* Computer Aided Design, 1998. **30**: p. 139-149.
6. Frey, P.J. and H. Bourouchaki, *Surface meshing using a geometric error estimate.* International Journal for Numerical Methods in Engineering, 2003. **58**: p. 227-245.
7. Frey, P.J. and P.-L. George, *Mesh generation : Application to finite elements*, ed. Wiley. 2008.
8. Guan, Z., et al., *An extended advancing front technique for closed surfaces mesh generation.* International Journal for Numerical Methods in Engineering, 2008. **74**(4): p. 642-667.
9. Laug, P. and H. Borouchaki, *BLSURF-mesh generator for composite parametric surfaces-user's manual.* INRIA Technical Report.
10. Lo, S.H. and T.S. Lau, *Mesh generation over curved surfaces with explicit control on discretization error.* Engineering Computations (Swansea, Wales), 1998. **15**(2-3): p. 357-373.
11. Laug, P. and H. Borouchaki, *Curve linearization and discretization for meshing composite parametric surfaces.* Communications in Numerical Methods in Engineering, 2004. **20**(11): p. 869-876.
12. Beall, M.W., J. Walsh, and M.S. Shephard, *Accessing CAD geometry for mesh generation*, in *Proceedings of 12th International Meshing Roundtable*. 2003: Santa Fe.
13. Dey, S., M.S. Shephard, and M.K. Georges, *Elimination of the adverse effects of small model features by the local modification of automatically generated meshes.* Engineering with Computers, 1997. **13**(3): p. 134-152.
14. Fine, L., L. Remondini, and J.C. Leon, *Automated generation of FEA models through idealization operators.* International Journal for Numerical Methods in Engineering, 2000. **49**(1-2): p. 83-108.
15. Shephard, M.S., M.W. Beall, and R.M. O' Bara, *Revisiting the elimination of the adverse effects of small model features in automatically generated meshes*, in *Proceedigs of the 7th International Meshing Roundtable*. 1998: Dearborn. p. 119-132.
16. Gao, S., et al., *Feature suppression based CAD mesh model simplification.* CAD Computer Aided Design, 2010. **42**(12): p. 1178-1188.
17. Quadros, W.R. and S.J. Owen, *Defeaturing CAD models using a geometry-based size field and facet-based reduction operators.* Engineering with Computers, 2012. **28**(3): p. 211-224.
18. Laug, P., *Some aspects of parametric surface meshing.* Finite Elements in Analysis and Design, 2010. **46**(1-2): p. 216-226.
19. Owen, S.J. and S. Saigal, *Surface mesh sizing control.* International Journal for Numerical Methods in Engineering, 2000. **47**(1-3): p. 497-511.
20. Pippa, S. and G. Caligiana, *GradH-Correction: Guaranteed sizing gradation in multi-patch parametric surface meshing.* International Journal for Numerical Methods in Engineering, 2005. **62**(4): p. 495-515.
21. Venkataraman, S. and M. Sohoni, *Blend recognition algorithm and applications*, in *Proceedings of the 6th ACM symposium on Solid modeling and applications*. 2001. p. 99-108.
22. Venkataraman, S., M. Sohoni, and R. Rajadhyaksha, *Removal of blends from Boundary Representation models*, in *Proceedings of the 7th ACM symposium on Solid modeling and applications*. 2002. p. 83-94.
23. Venkataraman, S. and M. Sohoni, *Reconstruction of feature volumes and feature suppression*, in *Proceedings of the 7th ACM symposium on Solid modeling and applications*. 2002. p. 60-71.
24. Lee, K.Y., et al., *A small feature suppression/unsuppression system for preparing b-rep models for analysis*, in *Proceedings of the 2005 ACM symposium on Solid and physical modeling*. 2005: New York.
25. Clark, B.W., *Removing small features with real CAD operations*, in *Proceedings of the 16th International Meshing Roundtable*. 2007: Seattle.
26. Thakur, A., A.G. Banerjee, and S.K. Gupta, *A survey of CAD model simplification techniques for physics-based simulation applications.* CAD Computer Aided Design, 2009. **41**(2): p. 65-80.
27. Sheffer, A., *Model simplification for meshing using face clustering.* Computer Aided Design, 2001. **33**: p. 925-934.
28. Sheffer, A., et al., *Virtual topology operators for meshing.* International Journal of Computational Geometry and Applications, 2000. **10**(3): p. 309-331.
29. Inoue, K., et al., *Clustering a large number of faces for 2-Dimensional mesh generation*, in *Proceedings of the 8th International Meshing Roundtable*. 1999: South Lake Tahoe.
30. Wei, P., C. Wenliang, and B. Yidong. *Mesh generation based on virtual geometry*. in *Proceedings of the 10th International Conference on Numerical Methods in Industrial Forming Processes*. 2010.

31. Tautges, T. *Automatic Detail Reduction for Mesh Generation Applications*. in *Proceedings of the 10th International Meshing Roundtable*. 2001.
32. Foucault, G., et al., *Adaptation of CAD model topology for Finite Element Analysis.* Computer Aided Design, 2008. **40**(2): p. 176-196.
33. Marcum, D.L. and J.A. Gaither, *Unstructured surface grid generation using global mapping and physical space approximation*, in *Proceedings of the 8th International Meshing Roundtable*. 1999: South Lake Tahoe.
34. Noel, F., *Global parameterization of a topological surface defined as a collection of trimmed bi-parametric patches: Application to automatic mesh construction.* International Journal for Numerical Methods in Engineering, 2002. **54**(7): p. 965-986.
35. Wu, B. and S. Wang, *Automatic triangulation over three-dimensional parametric surfaces based on advancing front method.* Finite Elements in Analysis and Design, 2005. **41**(9-10): p. 892-910.
36. Lee, C.K., *Automatic metric advancing front triangulation over curved surfaces.* Engineering Computations (Swansea, Wales), 2000. **17**(1): p. 48-74.
37. Lee, C.K., *Automatic metric 3D surface mesh generation using subdivision surface geometrical model. Part 2: Mesh generation algorithm and examples.* International Journal for Numerical Methods in Engineering, 2003. **56**(11): p. 1615-1646.
38. Tristano, J.R., J.O. Steven, and A.C. Scott, *Advancing front surface mesh generation in parametric space using riemannian surface definition*, in *International Meshing Roundtable*. 1998.
39. Lohner, R. and P. Parikh, *Generation of three-dimentional grids by the advancing front method.* International Journal for Numerical Methods in Fluids, 1988: p. 2135-2159.
40. Rassineux, A., *Generation and optimization of tetrahedral meshes by advancing front technique.* International Journal for Numerical Methods in Engineering, 1998. **41**(4): p. 651-674.
41. Moller, P. and P. Hansbo, *On advancing front mesh generation in three dimensions.* International Journal for Numerical Methods in Engineering, 1995. **38**: p. 3551-3569.
42. Foucault, G., et al., *An extension of the advancing front method to composite geometry*, in *Proceedings of the 16 th International Meshing Roundtable*. 2007.
43. Francois, V. and J.-C. Cuilliere, *3D Automatic remeshing applied to model modification.* Computer Aided Design, 2000. **32**(7): p. 433-444.
44. Dijkstra, E.W., *A note on two problems in connexion with graphs.* Numerische Mathematik, 1959. **1**(1): p. 269-271.
45. Qian, J. and Y. Zhang, *Automatic unstructured all-hexahedral mesh generation from B-Reps for non-manifold CAD assemblies.* Engineering with Computers, 2011: p. 1-15.
46. Clark, B.W., B. Hanks, and C. Ernst, *Conformal assembly meshing with tolerant imprinting*, in *Proceedings of the 17th International Meshing Roundtable*. 2008: Pittsburgh.