Proceedings of the IMProVe 2011
International conference on Innovative Methods in Product Design
June 15th – 17th, 2011, Venice, Italy

A parametric associative modelling of aeronautical structural concepts under C^0, C^1 or C^2 continuity constraints

V. Dattoma[(a)], M. De Giorgi[(a)], S. Giancane[(a)], P. Manco[(a)], A.E. Morabito[(a)]

[(a)] Dipartimento di Ingegneria dell'Innovazione Università del Salento – Lecce, Italy

Article Information

Keywords:
*design conceptualisation,
aeronautic structures,
primitives,
quad-mapped mesh,
CAD/CAE tool.*

Corresponding author:
*Marta De Giorgi
Tel.: +39(0)832297776
Fax.: +39(0)832297768
e-mail: marta.degiorgi@unisalento.it
Address: via Arnesano, 73100, Lecce, Italy*

Abstract

Purpose:

In this paper, an associative-parametric approach is proposed in order to model the mesh of an aeronautical concept starting from a set of high-level structural primitives. To assure orders of continuity higher than C^0 between adjacent instances, a suitable mathematical description of the structural primitives has been identified. The maintenance of the continuity constraints must be assured when the mesh is modified.

Method:

The Bézier curve and the Coons surface patch, with a suitable degree, are used in order to assure orders of continuity higher than C^0 in the connection points or edges. Appropriate schemes of dependences are identified to assure the automatic propagation of the modifications complying with the continuity constraints.

Result:

The approach here proposed allows the designer to carry out the geometric modelling and the automatic mesh generation within one software environment in a fast and interactive way and complying with the geometric continuity constraints and the one-to-one correspondence between the mesh elements. This represents evidently a large advantage since the structural optimization process is simplified, with a relevant man-hours saving. A lower number of data transfers between different software is, moreover, involved with less problems related to the data corruption. Finally low conceptual value operations, due to manual correction activity of the model, are eliminated.

Discussion & Conclusion:

The methodology here proposed allows the automatic propagation of modifications satisfying the geometric continuity constraints and the one-to-one correspondence between the mesh elements. The approach is implemented into a CAD/CAE tool, called MeshFEM and developed using C++ and Matlab languages and the VTK library for the 3D graphic visualization.

1 Introduction

Early design stage is a critical phase during which several important decisions concerning geometrical shapes, functions, materials and manufacturing processes are taken with a strong impact on the final product and its cost. During this phase it is very important to have efficient modelling, analysis and comparison tools so that several design solutions can be rapidly analysed in order to define the most promising design concept. Unfortunately most of the information generated and handled during the early design stage is qualitative and, consequently, it is hard to manage them with a computer system. Moreover, particularly in the case of a novel product, the input data required for the early design can be not clearly defined for a long period of time so that low-accuracy data are often considered in order to begin the design activity.

In the aeronautical field the early design stage has some peculiarities that differentiate it from other sectors. Firstly it involves multi-disciplinary competences (aero-dynamical, structural, etc). Most of the input data required by the structural design of an aeronautical product often comes from other fields, such as the aerodynamics. This is the case, for example, of the aerodynamic loads acting on the structure. Therefore, especially in the case of the development of novel aeronautical products, some clearly defined interfaces between different competence work centers must be identified in order to allow the exchange of reciprocally required data. In [1] the importance of a common parametric-associative geometry is pointed out in order to overcome these difficulties, which are typical of the aeronautical early design.

Aeronautical products are very complex structures whom high performances and reliability are requested to. Because of this complexity it becomes very problematic to carry out some reliable preliminary evaluations about the validity of the various proposed concepts. The several aspects involved in this validation process point out the necessity of developing some devoted CAD/CAE tools, which are highly interactive, user-friendly and able to quickly give some feedback about the design decisions.

In the literature there are several CAD/CAE tools focused on the parametric geometric modelling and on the relative aerodynamic optimisation of aeronautical concepts [2-5]. In the last years, numerous tools, devoted to preliminary aircraft design and including also structural optimisation module, have been developed. In the following, some examples are reported.

PrADO (Preliminary Aircraft Design and Optimisation) is a comprehensive library of about 500 FORTRAN programs [6]. The geometry of PrADO is not CAD-based and internal structural elements, such as ribs and stringers are not modelled.

MIDAS (Multidisciplinary Interactive Design and Analysis System) is one of the early approaches combining structural analysis and aerodynamics based on graphical CAE tools [7]. However, the geometry generated from this tool is code-based. If a new geometry representation is used, further coding work is necessary.

FIDO (Framework for Interdisciplinary Design Optimisation) is a tool, developed and used by the NASA Langley Research Center, based on the use of NURBS (Non-Uniform Rational B-Spline) [8]. This tool is integrated with a commercial CAD system, not allowing for associative design. Therefore, any geometrical change affecting several parts requires a lot of manual work.

In 2002, TU Munich [9] presented a new approach, based on the use of CATIA v5 and numerous Visual Basic routines, in order to automate the generation of parametric-associative geometry of the components. The used Visual Basic routines are slow and lead to large memory consumption. Moreover, this tool does not allow the automatic generation of the mesh. In 2006, Ledermann et al. [10] improved the use of CATIA v5 developing a tool, called K-operator, that allows to perform an automatic optimization of an aeronautical concept. The procedure consists of several steps: definition of a library of objects, customizing the features and writing a design manual for model generation. This procedure requires coding in C++, Java or Visual Basic. In both these papers, a common parametric-associative geometry is developed as a basis for different domains (structural analysis, CFD simulation).

For some years now, our research group has been developing a parametric-associative modelling approach focused on aircraft structural concepts. Thanks to this approach, the structural optimisation of the aeronautical concept can be carried out in a fast and efficient way. In fig 1 (diagram (a)), a typical methodology used for structural optimisation is shown; generally speaking, it involves a CAD and a FEM tool. A very critical step in this approach is the data transfer (red square in figure) from CAD to FEM environment. An accurate analysis of the imported file is often necessary because of topological errors which can be introduced during this passage. A bad adjacency or a misalignment of two or more geometrical elements of model can cause an incoherent mesh and senseless results. Moreover this approach requires the user to be deeply skilled in using CAD and FEM sw at the same time. Sometimes two different users are dedicated to these activities causing a significant loss of time. With the proposed approach, on the contrary, the geometric modelling and the mesh automatic generation are carried out within one software environment (diagram (b) in fig. 1). *This represents evidently a large advantage since the structural optimization process is simplified, with a relevant man-hours saving. A lower number of data transfers between different software is, moreover, involved with less problems related to the data corruption. Finally poor conceptual value operations, due to manual correction activity of the model, are eliminated.*

In order to develop this approach an early scheme of representation, based on 2D and 3D primitives representing the main structural components of the aeronautical concept, has been proposed by the authors in [11]. Based on these high-level primitives a *quad-mapped mesh* of the concept has been obtained. It consists of quad-elements (or quadrilateral-shaped elements with four nodes) which are located so that a regular pattern, with obvious rows of elements, can be identified [12].

With a view to giving the designer, since the very early phases of the design process, some information about the structural behaviour of the aeronautic concept, each primitive is parameterised in order to quickly evaluate several conceptual prototypes. If one parameter is changed, the geometric model must adapt to the changes in an automatic and efficient way. To assure this automatic propagation of the modifications in [13] the authors proposed a *hierarchical assembly*, based on well-defined schemes of dependences.

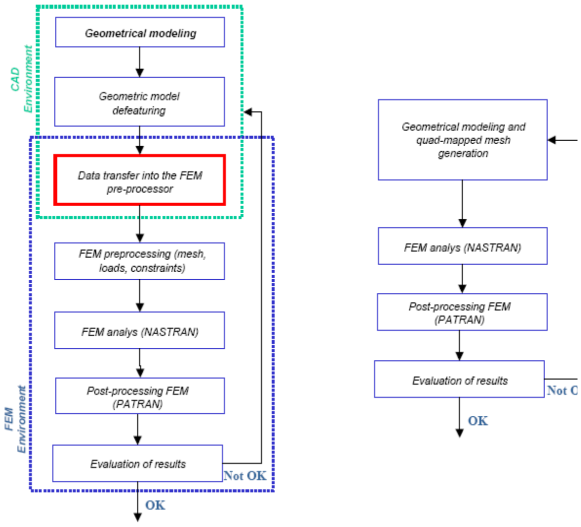
One shortcoming of the modelling approach proposed till that moment has been to not consider orders of continuity higher than C^0 between adjacent instances of the geometric model. The only constraint, which was taken into account, was the one-to-one correspondence between mesh elements. In this paper the mathematical description of the structural primitives has been improved in order to assure the required order of continuity (C^0, C^1 or C^2) between adjacent elements. The schemes of dependences have been enhanced too in order to guarantee that the propagation of modification occurs keeping the pre-established order of continuity.

The approach is implemented into a CAD/CAE tool, called MeshFEM and developed using C++ and Matlab languages and the VTK library for the 3D graphic visualization.

2. *Definition of the structural primitives for aeronautical concepts*

In an aeronautical structure, several elements can be identified; they are: ribs (or frames), stringers, spars and skins (fig. 2). In the approach here proposed, a primitive is defined for each type of structural element, so that its main characteristics are taken into account. Each primitive is parameterised so that several structural configurations can be quickly generated. In this way the designer can interact with geometrically simple primitives which, once instantiated and assembled, can provide a description of the aeronautical concept.

In the following sub-sections a brief description of these conceptual primitives is illustrated. The scheme in fig. 3, reports the primitives together with the input geometric parameters necessary for their definition.


CAD Environment
Geometrical modeling
Geometric model defeaturing
Data transfer into the FEM pre-processor
FEM preprocessing (mesh, loads, constraints)
FEM analys (NASTRAN)
Post-processing FEM (PATRAN)
Evaluation of results
Not OK
OK
FEM Environment
Geometrical modeling and quad-mapped mesh generation
FEM analys (NASTRAN)
Post-processing FEM (PATRAN)
Evaluation of results
Not C
OK

(a)   *(b)*

Fig. 1. Typical (a) and proposed (b) approach used for structural optimisation of aircraft concepts.

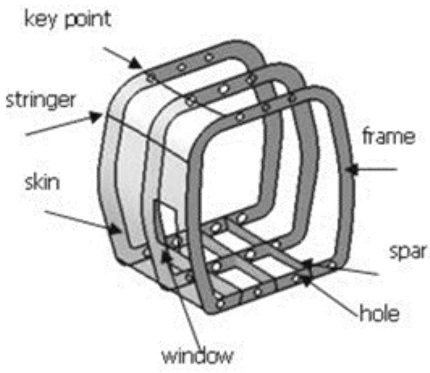


Fig 2. Basic elements of an aeronautical structure.

The rib (or frame)

The rib (or the frame for a helicopter fuselage) has several structural functions:

-) to maintain the correct external aerodynamic profile of the structure;

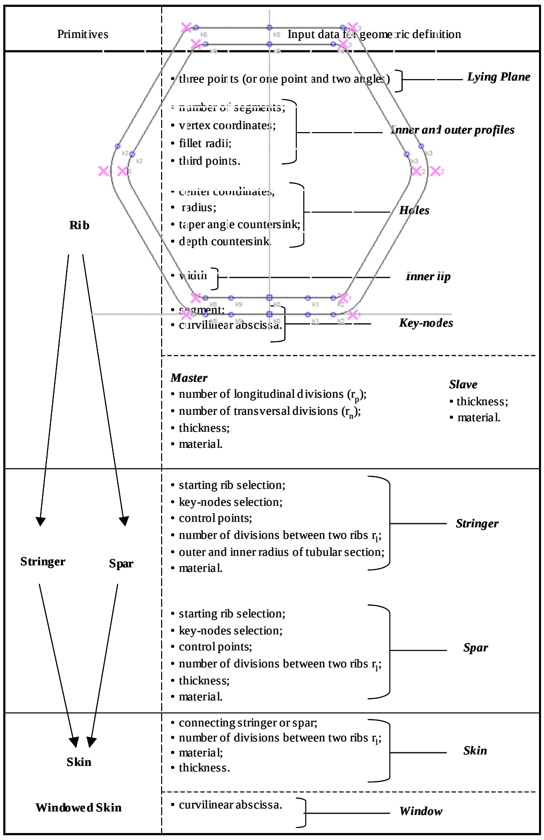
-) to transmit the loads from the skins to the spars.

It is a drilled thin planar panel with an internal profile to lodge the payload and a series of countersunk holes to allow the positioning of all the board instruments.

In order to locate the rib within the 3D modelling scene, a rib plane has to be preventively identified. The rib is, then, sketched specifying the geometry of the inner and the outer profiles. Each of them is defined specifying the vertex coordinates of a polyline consisting of rectilinear or circular segments filleted by circular arcs with C^1 continuity in the connection points. One or more countersunk holes can be introduced. For each of them the centre coordinates, the radius value and the countersinking size parameters must be specified.

Once that the rib has been geometrically defined, some points, referred to as *key-nodes*, are fixed on its profiles. They are defined selecting a rib segment and their position on it by the specification of a curvilinear abscissa value. Since these points will coincide with the nodes of the quad-mapped mesh it is fundamental that both the profiles of each rib have the same number of key-nodes. Moreover, in order to preserve the one-to-one correspondence between mesh elements of adjacent ribs, the profiles of all the frames must have the same number of key-nodes. A correct distribution of key-nodes on the rib profiles allows the mesh quality to be controlled especially in proximity of the critic regions, such as the fillet zone. They are progressively numbered starting from an origin key-node, called K$_0$ in fig. 4. The key-nodes also represent the connection nodes for the stringers and the spars that will be successively introduced between two adjacent ribs.

Fig 3. Primitives and input geometric parameters necessary for their definition


Primitives

Input data for geometric definition

Rib

- three points (or one point and two angles) — *Lying Plane*
- number of segments;
- vertex coordinates;
- fillet radii;
- third points. — *Inner and outer profiles*
- center coordinates;
- radius;
- taper angle countersink;
- depth countersink. — *Holes*
- width — *Inner lip*
- segment;
- curvilinear abscissa. — *Key-nodes*

Master
- number of longitudinal divisions (r_p);
- number of transversal divisions (r_n);
- thickness;
- material.

Slave
- thickness;
- material.

Stringer

Spar

- starting rib selection;
- key-nodes selection;
- control points;
- number of divisions between two ribs r_l;
- outer and inner radius of tubular section;
- material. — *Stringer*

- starting rib selection;
- key-nodes selection;
- control points;
- number of divisions between two ribs r_l;
- thickness;
- material. — *Spar*

Skin

- connecting stringer or spar;
- number of divisions between two ribs r_l;
- material;
- thickness. — *Skin*

Windowed Skin

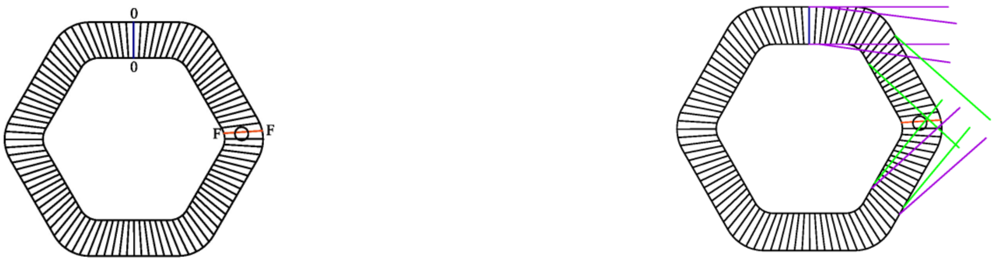
- curvilinear abscissa. — *Window*

Fig 4. An example of rib with numbered key-nodes.

The definition of the rib ends with the specification of the number of divisions r_p for the outer (or the inner) profile and the number of divisions r_n along the direction normal to the profile. The value of these resolution parameters can be explicitly set only if the rib is selected as master. All the others ribs, referred to as slave ribs, inherit the r_p and the r_n values from the master one. In any time, however, it is possible to convert one of the slave ribs into the master.

The quad-mesh generation of a rib needs the definition of the outer and the inner profile, the presence of a progressively numbered key-nodes (K_n) on profiles, the choice of a master rib and of the resolution along the profiles (r_p); it is subdivided into several phases:

1) the outer profile of the master rib is uniformly subdivided into r_p segments distributing the r_p nodes according to the simple following formula (1):



$$I = \frac{l_i}{perimeter\ of\ profile}\, r_p \qquad (1)$$

$$[subdivisions\, between\ \ keynodes(i\, and\ i+1)] = \begin{cases} 1 & if\ \ I \le 1 \\ \mathrm{int}(I) if\ \ I > 1 \end{cases}$$

where l_i is the length of the part of profile between i_{th} and $(i+1)_{th}$ key-node. These r_p nodes identify a partition of the outer profile, referred to as *master partition*.

2) the inner profile of the master rib and all the other profiles of the remaining slave ribs are subdivided according the master partition.

3) for each couple of corresponding nodes (i.e. belonging to inner and outer profiles of a rib) r_n-1 nodes are created by a linear interpolation. Each node can be identified by a couple of indices (i,j). The i-value identifies the node position along a profile; the j-value defines the node position between the two profiles. The quad-elements are built connecting, by straight lines, the (i,j) node with the nodes (i-1,,j), (i+1,j), (i,,j-1) and (i,,j+1) (fig. 5).

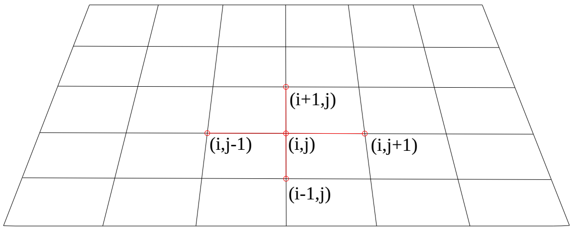


Fig 5. Elements definition algorithm.

The presence of one or more holes into a rib has important consequences on its structural behaviour and a simple algorithm is here proposed in order to get a quad-mapped mesh.

Starting from the mesh of the un-drilled rib a domain around the hole is established; here the mesh is locally modified leaving the remaining part as unchanged. This domain must be a quadrilater whose two edges are on the inner and the outer profiles of the rib and the others are two transversal segments of the un-drilled meshed rib. This domain must be as regular as possible. For this reason its boundary transverse segments must have, as much as possible, the same distance from the centre of the hole. In order to define this domain, the transversal segment (connecting the corresponding nodes identified on the inner and outer profiles of the rib), which is the closest to the centre of the hole, is firstly identified. In fig. 6a this segment is called as F-F, while the segment connecting the origin key-nodes (previously defined on the two profiles) is denoted by O-O. For each transversal segment, a couple of perpendicular straight lines from its ends is traced (fig. 6b). The segments, for which the related perpendiculars intersect F-F, are potential candidates for bounding the domain around the hole (Figure 6c). For each of these segments the following distance value is calculated:

$$\mathrm{d} = \frac{(\mathrm{PF})_{\mathrm{inner}} + (\mathrm{PF})_{\mathrm{outer}}}{2} \qquad (2)$$

where $(PF)_{inner}$ and $(PF)_{outer}$ are the distances between the points P (ends of the generic segment which is a potential candidate for bounding the domain around the hole) and F on the inner and the outer profile respectively.

If lF is the length of F-F, only the segments for which d is higher or equal than lF/2 are kept. Among them the segment to be considered as domain boundary is that one characterized by the minimum value for d (fig. 6d). In proximity of a hole there can be the transversal connecting edge of a spar (whose geometric modelling is illustrated in section 2.3), as shown in fig. 6e where this edge is called S-S. In this case the transversal edge of the spar automatically represents a boundary segment for the domain. If in proximity of a hole another hole is present, the boundary is moved as much as possible to avoid any possible intersection between two domains (detail in fig. 7). If the distance between the centre of the hole and the edge domain is less than 1.2x(hole radius), this hole is eliminated. In these two last cases the domain can lose its original regularity (fig. 6e).

The final result is a domain that can be subdivided in four sub-domains (fig. 6f). These sub-domains can be meshed with a quad-mesh. It is important to have a suitable number of diagonal subdivisions, s_d, to get a regular mesh; this value is chosen as follows:

$$s_d = \mathrm{int}\left(\frac{length\ \ of\ \ perimeter\ of\ domain}{number\ of\ elements\ on\ perimeter} \right) \qquad (3)$$

A hole can present an inner and out-of-plane lip all around its perimeter with an angle that the user can properly set. A single row of elements will constitute the mesh of this part.

Finally, a thickness value and a material are assigned to define the properties associated with the rib.

Figure 7 shows a rib resulting from this procedure and two detail views in the neighbourhood of as many couples of holes.

Fig 6. Definition of a hole domain.

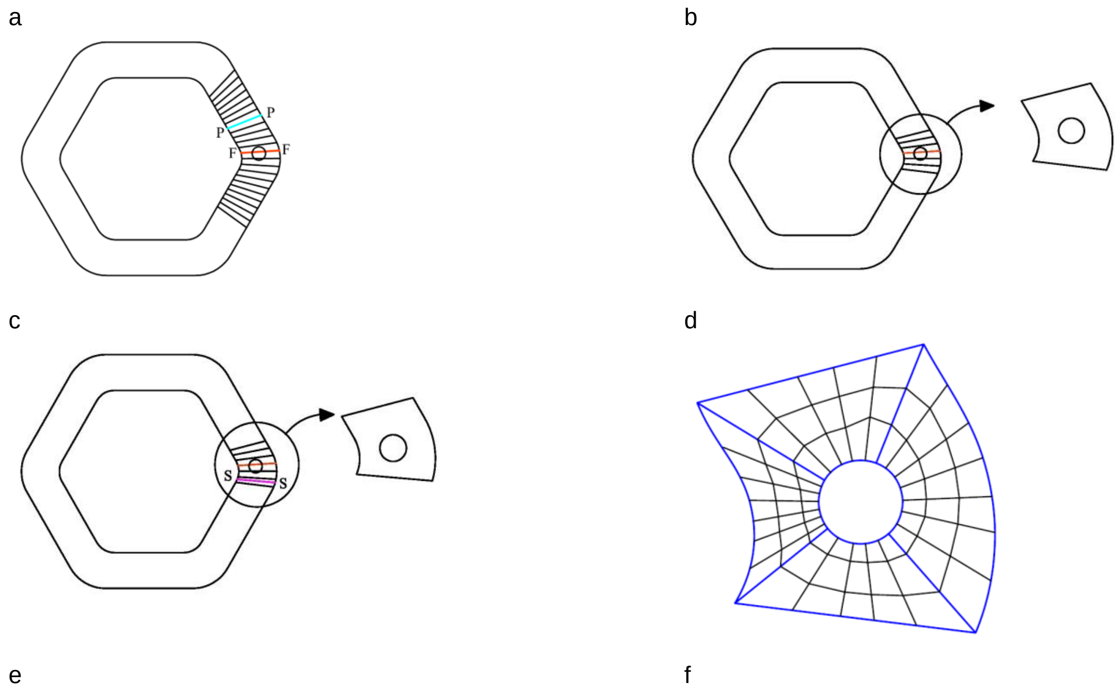


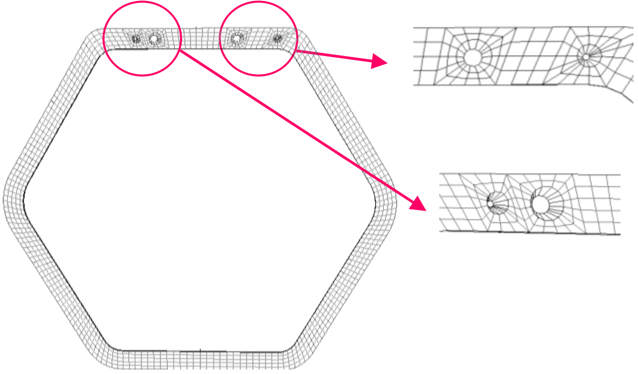
Fig 7. Rib and detail of holes.

2.2 The stringer

From the structural point of view, the stringer is a beam which fundamentally reacts to the bending stress. The stringers together with the spars connect two consecutive ribs and represent the elements where the external coating lies.

The stringer geometry is defined within a 3D modelling scene. It is represented by a 3D curve connecting two adjacent ribs. For each stringer a couple of connection nodes must be specified: each of these points is fixed on the external profile of a rib and it must coincide with one of the key-nodes. The stringer is modelled by a Bézier curve. This curve interpolates the first and the last point (i.e. the keynodes) and approximates all the others. The 5-degree Bèzier curve is automatically selected for a stringer which must satisfy C^2 continuity at both the extremities. In this case, in fact, the maintenance of C^2 continuity involves the automatic positioning of all the control points of the curve since this order of continuity fixes the position of the first (and last) three control points.

In all the other cases a 4-degree Bézier formulation is sufficient to adjust interactively the stringer shape guaranteeing the maintenance of the required continuity order. For example in fig. 8 the stringers A and B are modelled by two 4-degree Bézier sections; in the connection point P the C^2 continuity is preserved.



The stringers are subdivided using beam elements: the number of division is established setting the r_l value between two adjacent ribs. All the stringers connecting two adjacent ribs, in fact, are characterized by the same number of subdivisions. This number is equal to r_l.

The stringer is characterised by a hollow circle-shaped transverse section. This section geometry allows the designer to modify the bending and the torsional stiffness values, acting on the inner and the outer radii values. A material must be specified for each stringer.

Fig. 9 shows two adjacent ribs connected by a curved stringer .

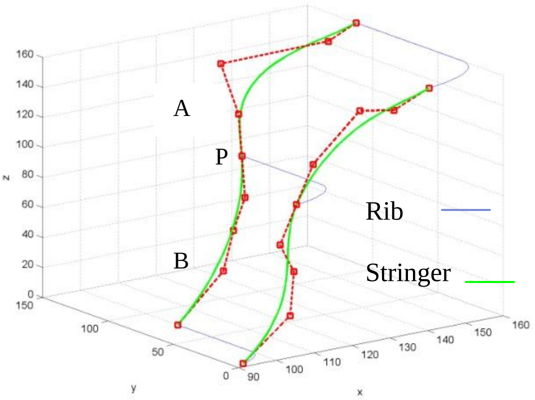
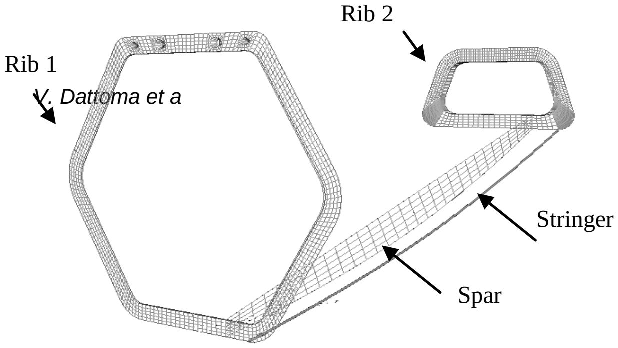


Fig. 8 Stringers modelled as Bézier curves

Fig. 9 Two ribs connected by a curved stringer and a curved spar.

2.3 The spar

The spars give the main stiffness to absorb the bending load applied to the aeronautical structure.



Generally speaking, they can be outlined as four edges-panels crossing two adjacent ribs.

For each crossed rib two vertices have to be specified, one on the outer profile and the other on the inner one. Each of them must coincide with one of the key-points previously fixed on the rib. These two key-points are, then, connected by a linear segment. The other two opposite edges, following the longitudinal direction of the aeronautical structure, are modelled as shown in sub-section 2.2 for the stringers. Thanks to the correspondence between the key-nodes on adjacent ribs, it is sufficient to select the starting key-node to define a spar: the remaining three key-nodes are, in fact, automatically identified.

Starting from these four edges a quad-mesh is generated by an algorithm based on degree-5 Coons surface patch and setting the number of division r_l between two adjacent ribs. This algorithm guarantees the C^2 continuiy between adjacent spars along the common transverse edge if a C^2 connection exists between the related longitudinal (internal or external) profiles. All the spars connecting two adjacent ribs are characterized by the same number of subdivisions along the longitudinal direction. This number is equal to r_l and it has the same value of the stringer resolution.

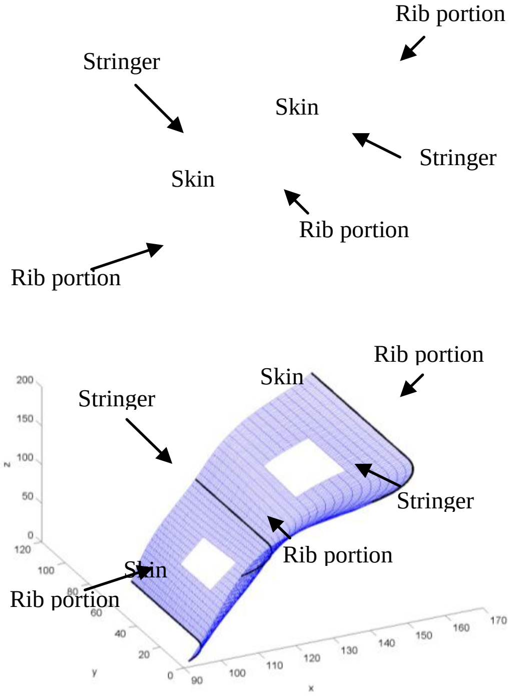
Finally, a thickness value and a material must be assigned to define completely the spar. Fig. 9 shows a curved spar between two ribs.

2.4 The skin

The skin is a 3D thin panel with the main role of covering the structure constituted by the previously described elements and allows an efficient transmission and distribution of loads.

The skin can be modelled specifying the set of four edges delimiting the panel. Two opposite edges of this loop are circular or rectilinear segments belonging to two subsequent ribs. The others are 3D curves representing stringers or the external longitudinal profile of spars. Also in this case an algorithm based on degree-5 Coons surface patch is implemented to guarantee the smooth connection between adjacent skins, if a C^2 connection exists between adjacent edges. In order to define a skin, it is sufficient to select the first connector element, starting from it the others three edges are automatically identified.

As shown in fig. 10, the C^2 continuity between adjacent stringers implies the C^2 continuity of the two skins along the common edge.



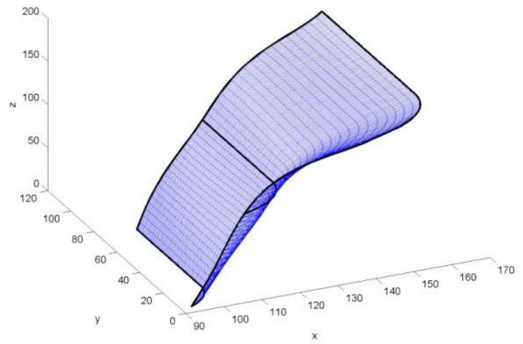


Fig. 10 Two adjacent skins with C2 continuity

Some windows can be included into the skin. A simplified method for modelling them is here proposed. It consists in identifying the four vertices, representing the projection of the window corners on the skin. These vertices are defined by specifying, for a couple of adjacent edges delimiting the skin, the values of curvilinear abscissa. Two windowed skins are shown in fig. 11.

As regards the element type, the mesh is made of shell elements in the case of ribs, spars and skins, and of beam elements for stringers.

Fig. 11. Two windowed skins.

3. Definition of dependences for the propagation of modifications

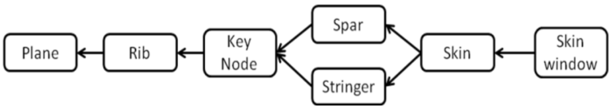
During the preliminary design of an aeronautical structure, several characteristics of the aircraft concept can be iteratively modified.

Generally, the propagation of a modification does not interest necessarily the whole structure. For this reason, the proposed approach aims at avoiding the total regeneration of the model when a modification occurs, reducing time-consuming operations. The adopted strategy consists in propagating the modifications only where strictly necessary, according to well-defined schemes of dependence, as explained later on. In particular two kinds of dependence are here considered: the “dependence between definitions” and the “dependence between meshed elements”. They are detailed in the two following sub-sections.

3.1 Dependence between definitions

The removal of some elements can cause a loss of information necessary for the definition of other parts of structure. To correctly manage this dependence, a hierarchy among classes of elements is defined. Every instance of a class depends directly on the definition of one or two instances of the previous class (as reported in fig. 12). When an instance is created or modified the user must specify its mathematical description giving values to the parameters and identifying the other instances necessary to its definition (*bearing instances*). The removal of an instance, which is bearing for others, makes these lasts unstable (i.e. not well-defined). If an instance results to be unstable, then it is deleted. This removal could result as critical for the definition of other parts that, consequently, are deleted. The check of the unstable instances can be efficiently performed on the basis of the scheme of dependences in fig. 12. Since the definition of an instance for a class depends only on the previous ones and never on the following ones the checking process can be carried out in a single step from left to right (fig. 12).

Fig. 12 Scheme of dependences between definitions



3.2 Dependence between meshed elements

The strategy here adopted allows to propagate the modifications only where strictly necessary. In order to implement it, a second type of dependence, called “dependence between meshed elements”, has been considered. The graph, shown in fig. 13, allows to correctly manage this dependence.

Generally, the process of mesh generation of each element requires different information on the basis of the element type. These data can be derived from the mathematical description of the element, from global parameters (for example the number of divisions r_p or r_n of the master rib), from the definition of connected elements or from the mesh of these last ones.

The black blocks in fig. 13 pertain to global properties (such as, for example, the subdivision of master profile) while the orange ones refer to single element property. The first column on the left side of the graph includes both black and orange blocks. Since the data for the definition of these blocks are directly updated by the user and not generated from other data no arrows start from them. On the contrary, the blocks in the remaining columns represent types of data that are always automatically regenerated starting from information contained in the blocks where the arrows point. This information can be portions of element mesh or intermediate results of mesh generation (such as partition and polygonal curves).

For each instance of the classes in the graph, the value of a given property can depend on one or more properties of the connected elements. For this reason, the instance *checks* the properties of the connected elements in order to detect some possible variations. If a modification is detected, the instance updates the value of its own properties. If the updated properties are monitored by other instances, then these last ones update their properties consequently. As a result the propagation of the modifications starts. Since the graph does not include any cyclic dependences between the elements properties (except the cycles indicated by red arrows in fig. 13 and explained later on) the propagation crosses the graph as a wave from left to right.

To clarify this concept, an example is now illustrated. Let A and B be two stringers between portions of two ribs sections and let C be the relative skin (fig. 14).

To generate its mesh, the skin C uses the property "Stringer (polygonal curve)" of A and B. The stringers A and B generate the relative polygonal curves starting from their analytical description "Stringer (curve, key node)". If the user modifies the analytical shape of the stringer A (for example translating a control point), the polygon of this stringer (i.e. the property "Stringer (polygonal curve)") obtained from the previous analytical description becomes “not valid” and then it is regenerated. Likewise, the mesh of the skin C, generated on basis of the original polygonal curve of stringer A, results “not valid” and then regenerated.

This mechanism of propagation must occur in accordance with the imposed order of continuity between adjacent elements. To guarantee the maintenance of the required order of continuity, one element (the *slave element*) modifies the value of its first and second derivatives in the connection point on the basis of the other element (the *master element*). This means that, if the user modifies the shape of the master stringer, the slave stringer must be automatically regenerated to satisfy the new value of the derivative.

The slave and master roles can be assigned to the elements in each connection point or edge. This mechanism is valid also for configurations where two spars or one stringer and one spar are connected since the profiles of a spar are modelled like the stringer. This is the reason for the presence of the red cycles within the graph in fig. 13.

As default the continuity between adjacent elements is of C^0 order and the control points of A, B and C are explicitly defined by the user. It is possible to change the order of continuity in one or both the connection points P and Q. In order to have a C^2 continuity in P, the user must assign the role of master to one of the stringers, for example the stringer A. Consequently the control points of the stringer B change their position ignoring the initial definition. The analytical description of the stringer changes (i.e. the property "Stringer (curve)") and, in order to propagate the modification, the polygon B will be regenerated.

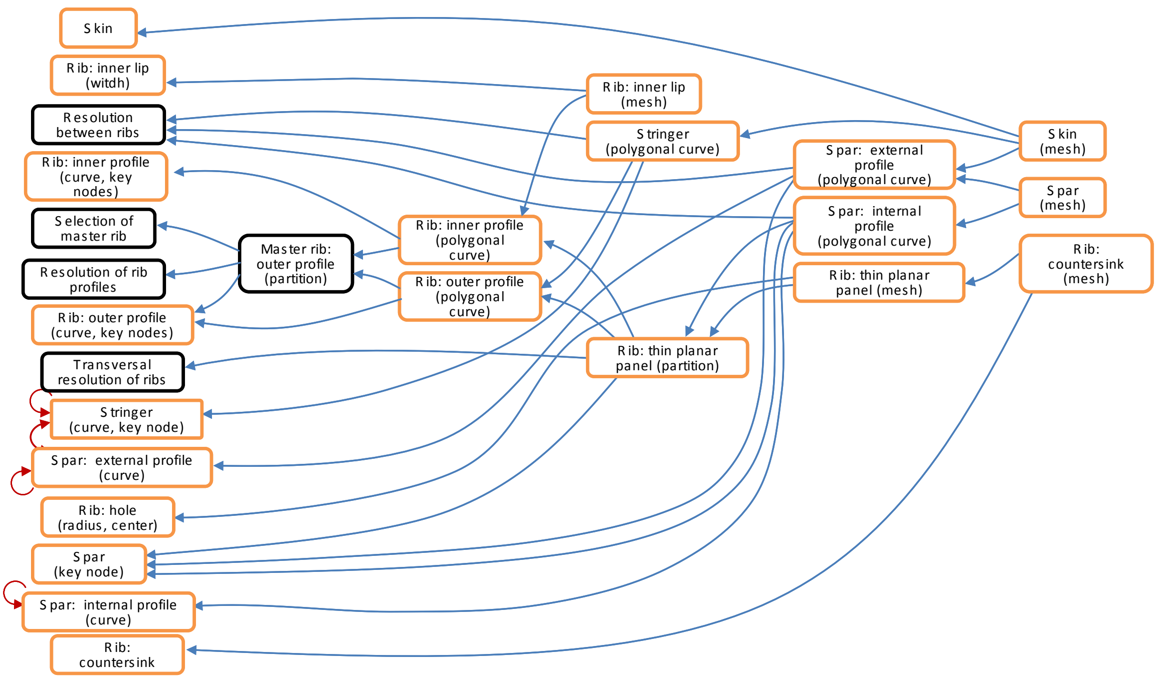

Skin
Rib: inner lip (witdh)
Resolution between ribs
Rib: inner profile (curve, key nodes)
Selection of master rib
Resolution of rib profiles
Rib: outer profile (curve, key nodes)
Transversal resolution of ribs
Stringer (curve, key node)
Spar: external profile (curve)
Rib: hole (radius, center)
Spar (key node)
Spar: internal profile (curve)
Rib: countersink
Master rib: outer profile (partition)
Rib: inner profile (polygonal curve)
Rib: outer profile (polygonal curve)
Rib: inner lip (mesh)
Stringer (polygonal curve)
Rib: thin planar panel (partition)
Spar: external profile (polygonal curve)
Spar: internal profile (polygonal curve)
Rib: thin planar panel (mesh)
Skin (mesh)
Spar (mesh)
Rib: countersink (mesh)

Fig. 13 Graph of the dependences between mesh.

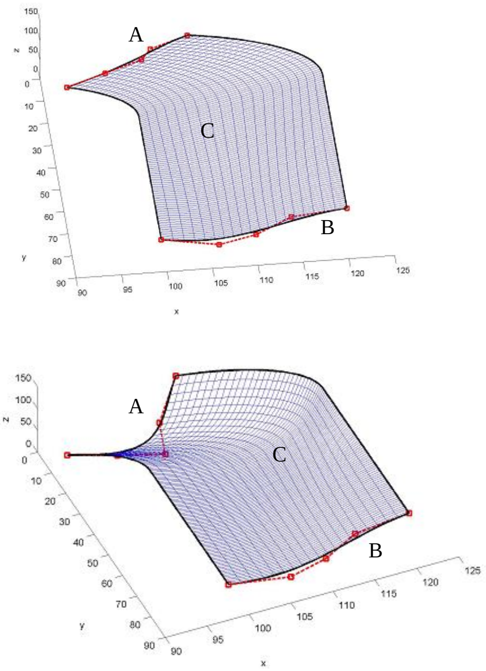


Fig. 14 Example of propagation of modifications

The stringer C is not involved in the propagation of the modifications since the continuity in Q is of C^0 order.

If the user requires the C^2 continuity in the point Q selecting C as the master in the couple {B, C} the control points of the slave stringer (B) change their position and the polygonal curve of B is regenerated. This operation does not affect the stringer A since it is the master stringer in the couple {A, B}. From now on the shape of the stringer B will depend on that of two master stringers A and C. If A or C's shape changes, also B must change.

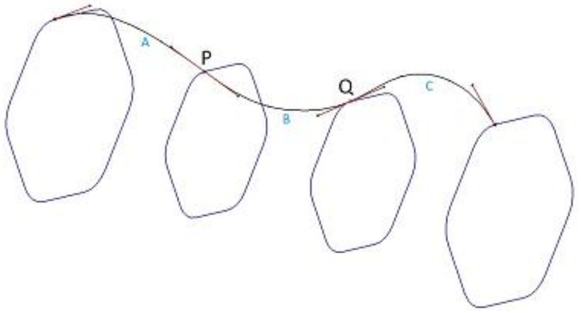


Fig. 15 Example of propagation of modification in the case of C^1 or C^2 continuity.

If the role of A and B is exchanged, the control points of B near the point P became again those previously specified by the user, while the stringer A changes in order to maintain the C^2 continuity in P. Now, C is the master stringer in the couple {B, C} and B is the master in the couple {A, B}: a modification of the stringer C involves a modification in the stringer B and then a modification in the stringer A. A and B polygonal curves change in order to maintain the continuity in P and Q.

If the user removes the stringer B, then the stringers A and C loose their roles respect to the removed stringer and the control points of A, that was a slave stringer for B, recover their initial position defined by the user.

4. CASE-STUDY

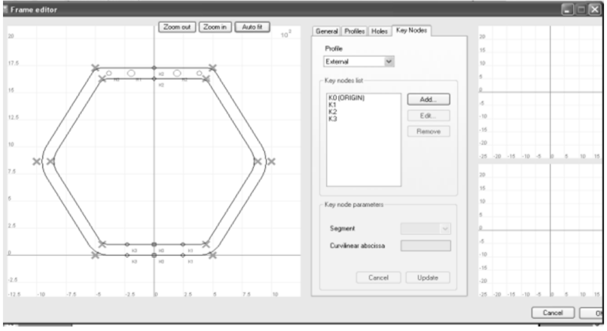
The approach here proposed has been implemented into a CAD/CAE tool, called MeshFEM and developed using C++ and Matlab languages and the VTK library for the 3D graphic visualization.

In this section a case study, represented by a tail of helicopter fuselage is modelled using the MeshFEM tool. It is composed by: three frames, two spars, several stringers and several skins. In the following, the procedure to perform the parametric modelling of the structure is presented.

-) Frame geometric definition.

For each frame a plane must be identified assigning the spatial coordinates of three points. A frame is geometrically defined specifying the inner and outer profiles, the holes and the key-points.

In fig. 16 the sketch plane where the user defines the frame geometry is shown.



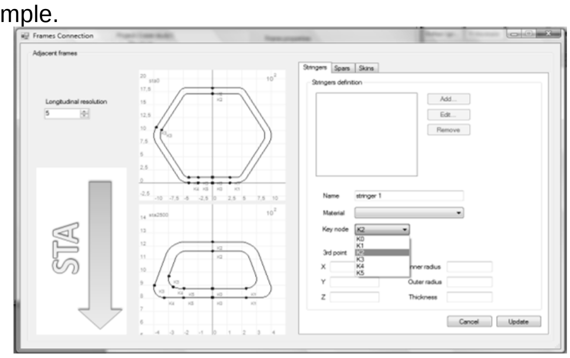
Profile: External

Key nodes list:
K0 (ORIGIN)
K1
K2
K3

Fig. 16 Definition of a rib with holes and key-nodes.

-) Stringer and spar geometric definition

The stringers and the spars are geometrically identified starting from the key-points previously fixed on the ribs. In fig. 17 is reported the definition of a stringer as example.



Longitudinal resolution: 5

Name: stringer 1

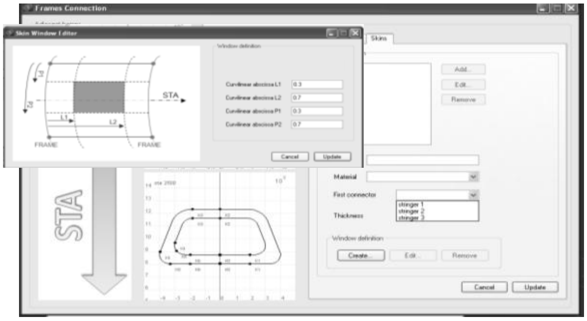
Key node: K2

Fig. 17 Definition of a stringer.

-) Skin geometric definition;

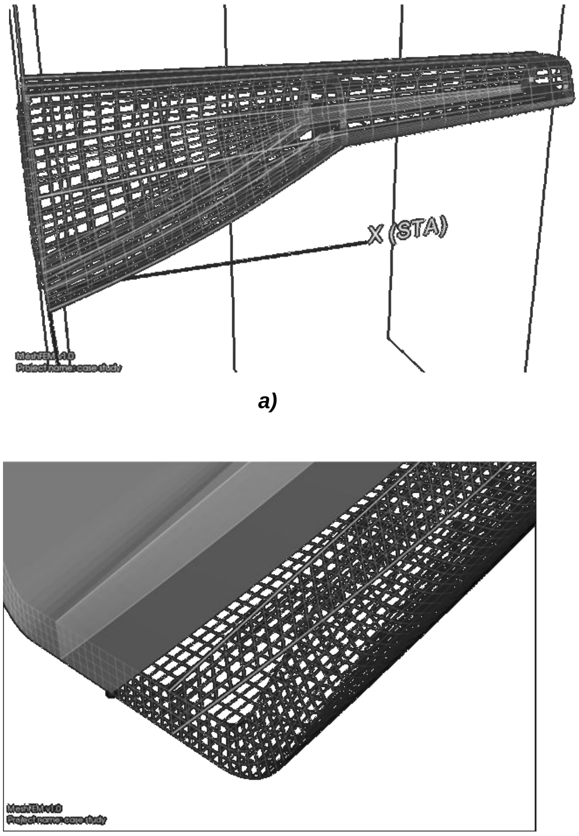
In fig. 18 the windowed skins are geometrically modelled starting from the edges selected on the frames and the curve segments between two frames belonging to the stringers.

Finally, fig. 19a shows the entire model of the helicopter fuselage displayed in a 3D visualization. A portion of the quad-meshed structure is represented in fig. 19b, where it is possible to observe the continuity of the mesh along the borders shared by several primitives.



Curvilinear abscissa L1: 0.3
Curvilinear abscissa L2: 0.7
Curvilinear abscissa P1: 0.3
Curvilinear abscissa P2: 0.7

Fig. 18 Definition of a skin with a window.

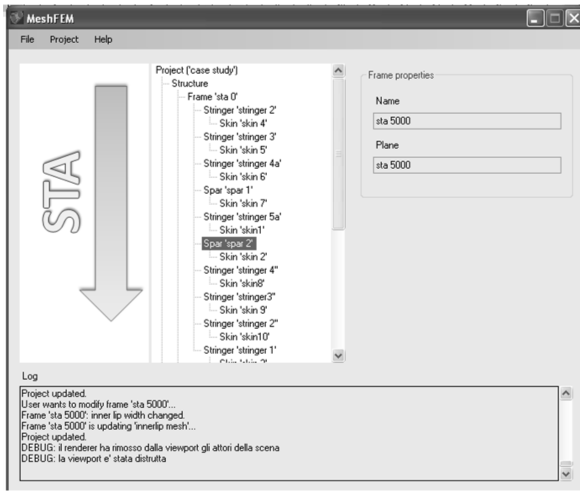


a)

b)

Fig. 19 The entire model of the helicopter fuselage in 3D scene (a) and a portion of the quad-meshed structure (b).

The graphical interface of the tool includes a logical tree where each primitive, defined by the user, appears as node. Thanks to this logical tree and to the parametric geometric definition of the primitives, it is very easy for the designer to modify, remove or add new primitives at any time (fig. 20). This graphical interface avoids the user the input of wrong values and narrows the selection of the bearing instances to the only existent elements.



```
Project ('case study')
  Structure
    Frame 'sta 0'
      Stringer 'stringer 2'
        Skin 'skin 4'
      Stringer 'stringer 3'
        Skin 'skin 5'
      Stringer 'stringer 4a'
        Skin 'skin 6'
      Spar 'spar 1'
        Skin 'skin 7'
      Stringer 'stringer 5a'
        Skin 'skin1'
      Spar 'spar 2'
        Skin 'skin 2'
      Stringer 'stringer 4''
        Skin 'skin8'
      Stringer 'stringer3''
        Skin 'skin 9'
      Stringer 'stringer 2''
        Skin 'skin10'
      Stringer 'stringer 1'
```

Name: sta 5000

Plane: sta 5000

```
Project updated.
User wants to modify frame 'sta 5000'...
Frame 'sta 5000': inner lip width changed.
Frame 'sta 5000' is updating 'innerlip mesh'...
Project updated.
DEBUG: il renderer ha rimosso dalla viewport gli attori della scena
DEBUG: la viewport e' stata distrutta
```

Fig. 20 The logical tree with all the primitives

Once the aeronautical structure is modelled, the interactive selection within the 3D scene of the nodes, where load and boundary conditions are applied, is carried out. The CAD/CAE tool allows writing a data file (with .bdf format) as input for NASTRAN solver.

Figure 21 shows an example of propagation of the modifications; in comparison with the fig. 19a) here the control points of the upper stringer (in the left section of the fuselage) are no more collinear,

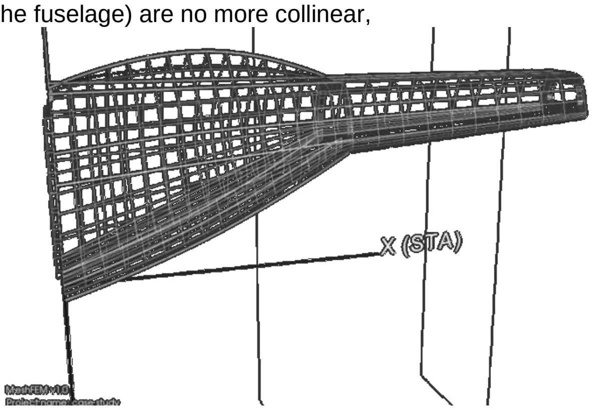


Fig. 21 An example of propagation of modifications.

5 Conclusions

For some years now our research group has been developing a parametric-associative modelling approach focused on aircraft structural concepts. This approach allows the designer to carry out the geometric modeling and the automatic mesh generation within one software environment. This represents evidently a large advantage since the structural optimization process is simplified. Moreover the data translation from a format into the other is avoided, together with poor conceptual value operations, due to manual correction activity of the model, In this paper the the mathematical description of the structural primitives has been improved in order to include higher order of continuity (C^1 and C^2) between adjacent instances. Based on the scheme of representation here proposed, a quad-mapped mesh is obtained. A suitable algorithm has been implemented in order to optimize the mesh quality in the neighborhood of the holes.

The automatic propagation of modifications is efficiently implemented according to well-defined schemes of dependence thanks to which the modifications involve only the interested instances. These schemes assure that the propagation of modification occurs keeping the pre-established order of continuity.

This approach has been implemented into a CAD/CAE tool, called MeshFEM, using C++ and Matlab languages and the VTK library for the 3D graphic visualization. This tool, opportunely interfaced with a FEM solver, allows the designer to obtain, since the very early phases of the design process, some crucial structural information based on which a conceptually valuable design solution can be selected.

References

[1] C. Ledermann, C. Hanske, J. Wenzel, P. Ermanni, R. Kelm *Associative parametric CAE methods in the aircraft pre-design*. Aerosp. Sci. and Technol 9: (2005) 641–651.

[2] X. Xie, Ch. Haberland A new numerical design tool for concept evaluation of propeller aircraft, Aircr. Des. 2: (1999) 147-165.

[3] M. Nemec, M. J. Aftosmis, T. H. Pulliam *CAD-Based Aerodynamic Design of Complex Configurations Using a Cartesian Method*, Proceedings of Aerospace Sciences Meeting AIAA 2004–0113 42nd.

[4] M. Athanasopoulos et al. *Parametric design of aircraft geometry using partial differential equations*, Adv Eng Softw*,* doi:10.1016/j.advengsoft.2008.08.001, (2008).

[5] S. Sarakinos, I. Valakos, I. Nikolos *A software tool for generic parameterized aircraft design*. Adv Eng Softw, 38 (1), (2007) 39–49.

[6] C.M. Österheld, W. Heinze, P. Horst *Preliminary design of a blended wing body configuration using the design tool PrADO*, in: Proceedings of the CEAS Conference on Multidisciplinary Aircraft Design and Optimisation, Köln, June 25–26, 2001.

[7] X. Luo, H. Rajagopalan, R. Grandhi *MIDAS: Multidisciplinary Interactive Design and Analysis System – integration of ASTROS and I-DEAS*, Proceedings of 37th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, Salt Lake City, UT, 1996.

[8] J.C. Townsend, J.A. Samaresh, R.P. Weston, W.E. Zorumski *Integration of a CAD system into an MDO framework*, Technical Report NASA/TM-1998-207672, NASA – Langley Research Center, Hampton, VA, 1998.

[9] T. Richter, M. Mechler, D. Schmitt *Integrated parametric aircraft design*, Technical Report, Institute of Aeronautical Engineering, Technical University Munich, 85747 Garching, 2002.

[10] C. Ledermann, P. Ermanni, R. Kelm *Dynamic CAD objects for structural optimization in preliminary aircraft design*, Aerospace Science and Technology 10: (2006) 601–610.

[11] Cocciolo M., De Giorgi M., Giancane S., Morabito A., Nigri M., *Representation of conceptual design solutions in aeronautic field*, XVII ADM-XXI INGEGRAF International Congress, Lugo, Spain, 9-12 Giugno 2009

[12] Documentation for Ansys 11.0 Chapter 7 – Generating the Mesh.

[13] V. Dattoma, M. De Giorgi, S. Giancane, P. Manco, A.E. Morabito, A parametric-associative Modelling of aeronautical concepts for structural optimization, The Tenth International Conference on Computational Structures Technology, Valencia, Spain, 14-17 September 2010.