

---

# Geometry modelling and industrial parameterisation issues

**Vittorio Selmin**

*Alenia Aeronautica S.p.A., Corso Marche, 41. I-10146 Torino*  
*vselmin@aeronautica.alenia.it*

---

*ABSTRACT. The paper addresses issues related to the development of a geometrical modelling system and its connection with shape parameterisation and computational grid update. In particular, it is thought that the use of parameterisations based on the modification of an existing shape instead to exploit the direct representation of the geometry results in an increase of flexibility and accuracy. Selected examples related to aircraft design illustrate the capability of the system.*

*RÉSUMÉ. On décrit un système pour la manipulation de modèles géométriques, ainsi que sa relation avec la paramétrisation de la forme et la modification de maillages de calcul. En particulier, l'utilisation d'une paramétrisation qui se base sur la modification d'une géométrie existante devrait être plus précise et flexible qu'une procédure qui s'appuie sur une représentation directe de l'objet. Des exemples issus de la conception aéronautique illustrent le système.*

*KEYWORDS: geometrical modelling, parameterisation, grid update, optimisation, CAD system, shape modification.*

*MOTS-CLÉS : représentation géométrique, paramétrisation, maillage, optimisation, système CAO, modification de la forme.*

---

DOI:10.3166/REMN.17.127-148 © 2008 Lavoisier, Paris

## 1. Introduction

Shape parametrisation (Samareh, 2001; Castonguay *et al.*, 2007) is perhaps the most critical issue of an optimisation process, that appears to be heavily constrained by the design parameters. An insufficient number of design variables or an inadequate choice of them will lead to poor optimisation results. In addition, design variables might be selected in such a way that they are close to engineers that will use the automatic optimisation system. Indeed, semi-empirical knowledge is often parametrised with respect to some geometrical quantities and it could be advantageous at least to exploit this knowledge through the use of linear inequality constraints. The parametrisation may address the direct representation of the geometry or the modification of an initial shape, the latter resulting at our knowledge more flexible and more accurate.

When the design problem is applied to complex shapes, the shape parametrisation becomes a difficult task. The parametrisation problem will definitively be solved by the implementation of the parametrisation directly within the CAD representation (Daumas *et al.*, 2002; Fudge *et al.*, 2005), as it has been proposed into modern CAD tools, but only if it is completely under the user control. The use of commercial CAD systems will be feasible only if they will allow in the future easy communication with external software, which is yet seldom the case. An alternative might be the use of the so-called CAD-free approach (Mohammadi *et al.*, 2004; Fudge *et al.*, 2005; Daoud *et al.*, 2005).

In addition, important issues are related to the implementation of specific parametrisation techniques requested by the user and to the necessity to control the overall optimisation chain in particular when dealing with MDO problems. It appears to be more attractive to work directly from mathematical models of the geometry extracted from the CAD model, in particular with those used in the grid generation process in order to establish a strong link between geometry modification and grid surface update.

The geometry modelling system which is the object of the paper exploits a data base extracted from the CAD system, that is in addition used for grid generation purpose. The geometrical data base contains the coordinates of corners (curves ends), the mathematical definition of support curves and surfaces, and topological relations that describe the links between the different geometrical entities. Curves and surfaces are defined into the parametric space. In addition, surfaces are characterised with respect to defining control nodes which are linked to the shape parametrisation. Modification of the coordinates of those nodes results in a new definition of the support surfaces. Updated support curves and corners are obtained through geometrical intersections. The geometry grid update maintains the topology of the original grid unchanged and is well suited when combined with a mesh deformation technique for the three-dimensional grid update.

The transfer of information from the modified shape to the computational grid has to be addressed in terms of flexibility, but above all in terms of accuracy. This is the reason why a mesh deformation strategy is in most cases more accurate than to use

remeshing procedures. The mesh deformation is obtained by the resolution of either discretisation of elasticity models, as in (Nielsen *et al.*, 2002), or an adhoc discretized system, relying in many cases on fictive springs controlling edges lengths with lineal springs, (Batina, 1989), or elements deformation with torsional springs, (Degand *et al.*, 2002). What is important is not to obtain a very good grid quality – although this improves the quality of the solver solution – during the overall optimisation process, but to maintain the same quality of that of the initial grid. Otherwise, noise related to non-homogeneous grid quality will be introduced in the optimisation process. This is not only true when associated to deterministic optimisation methods but also for stochastic ones.

## 2. Elements of unstructured grid generation

### 2.1. CAD representation

In three dimensions, the domain to be discretised can be viewed as a region bounded by surfaces which intersect along curves (Figure 1). The portions of these curves and surfaces needed to define the three dimensional domain of interest are called curves and surfaces components, respectively. In addition boundary curves and surfaces are oriented. This is important in the grid generation process as it is used to define the location of the region that has to be discretised. The orientation of a boundary surface is defined by the direction of the inward normal. The orientation of the boundary curves is defined with respect to the boundary faces which contain them. Each boundary curve is common to two boundary surfaces and will have opposite orientations with respect to each of them. The geometrical definition is in general provided through a parametric representation by using a CAD system and forms the so-called geometrical support, which represents the mathematical support of the geometrical elements.

A support curve can be described by a piecewise parametric representation (Figure 2). In this representation, the curve is subdivided into  $n_a$  arcs, and the position vector  $\mathbf{r}$  of a generic point on each arc is expressed as a function of a single real parameter  $u$ , which by convention varies into the interval  $[0, 1]$ :

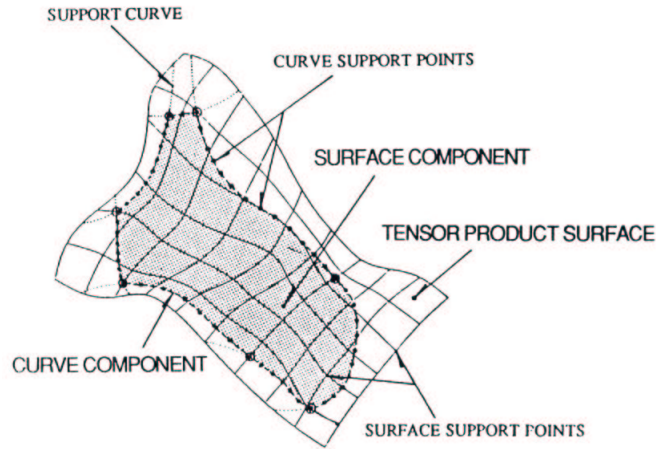
$$\mathbf{r} \equiv [x, y, z] \quad [1]$$

This function may be represented by a polynomial whose rank may change from arc to arc and which can be expressed as:

$$\mathbf{r} = \mathbf{r}(u) = \mathbf{a}_0 + \mathbf{a}_1 u + \mathbf{a}_2 u^2 + \dots + \mathbf{a}_n u^n \quad [2]$$

where each  $\mathbf{a}_k$  is a vector formed by three coefficients:

$$\mathbf{a}_k = [a_k^x, a_k^y, a_k^z] \quad [3]$$



**Figure 1.** *Surface geometrical support*

which represents the components of  $\mathbf{a}_k$  with respect to a Cartesian reference system  $(x, y, z)$ , whereas  $n$  is the rank of the polynomial. Relation [2] can be rewritten as follows

$$\mathbf{r}(u) = \mathbf{A}\mathbf{U} \quad [4]$$

where

$$\mathbf{A} = [\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] \quad [5]$$

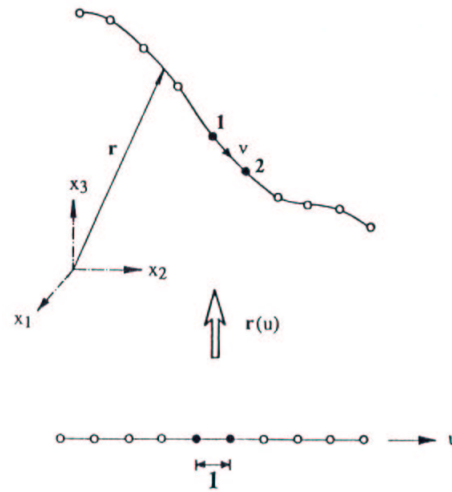
and

$$\mathbf{U} = [1, u, u^2, \dots, u^n]^T. \quad [6]$$

A point on the curve can then be identified by the number of the arc on which it is lying and the value of the parametric coordinate  $u$ , which is usually called local parametric coordinate.

The surfaces are represented by using a similar scheme. They are subdivided into patches which form a regular grid on the so-called parametric plane (Figure 3). On this plane, we can define with respect to each patch the local parametric coordinates  $u$  and  $v$  which vary in the interval  $[0, 1]$ . The position  $\mathbf{r}$  of a node on the surface can then be expressed as a polynomial expansion in  $u$  and  $v$  on each patch:

$$\mathbf{r}(u, v) = \mathbf{a}_{0,0} + \mathbf{a}_{1,0}u + \mathbf{a}_{0,1}v + \mathbf{a}_{2,0}u^2 + \mathbf{a}_{1,1}uv + \dots + \mathbf{a}_{n,m}u^n v^m \quad [7]$$



**Figure 2.** Curve parametric representation

The total rank of the polynomial is  $nm$ , whereas  $n$  and  $m$  represent the rank of the polynomial with respect to variables  $u$  and  $v$ . Here again, the quantities  $\mathbf{a}_{k,l}$  are formed by their components with respect to the Cartesian reference system  $(x, y, z)$ :

$$\mathbf{a}_{k,l} = [a_{k,l}^x, a_{k,l}^y, a_{k,l}^z] \tag{8}$$

Equation [7] can be rewritten in the following form

$$\mathbf{r}(u, v) = \mathbf{UAV} \tag{9}$$

where

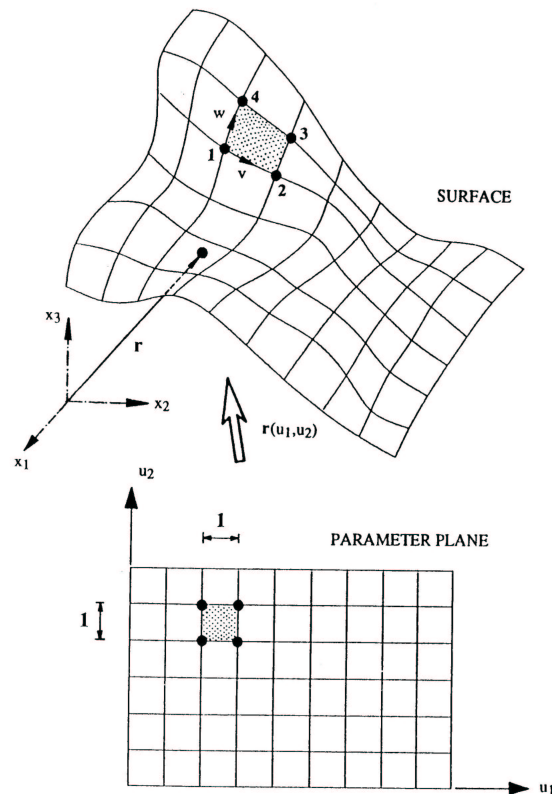
$$\mathbf{A} = [\mathbf{a}_{k,l}] \tag{10}$$

for  $k=(0, \dots, n)$  and  $l=(0, \dots, m)$ . The quantities  $U$  and  $V$  have a shape similar to those of Equation [6].

**2.2. Unstructured grid generation**

The advancing front technique (Peraire *et al.*, 1990) is here described, but other techniques like those based on Delaunay triangulation (Weatherhill, 1990; George, 1991) may be used as alternatives for unstructured mesh generation. The grid generation problem consists of subdividing an arbitrary complex domain into a consistent assembly of elements. The consistency of the generated mesh is guaranteed if the

generated elements cover the entire domain and the intersection between elements occurs only on common points, sides or triangular faces in the three-dimensional case. The final mesh is built in a bottom-up manner. The process starts by discretising each boundary curve. Nodes are placed on the boundary curve components and then contiguous nodes are joined with straight line segments. In a latter stage of the generation process, these segments will become sides of some triangles. The length of these segments must therefore, be consistent with the desired local distribution of mesh size. This operation is repeated for each boundary curve in turn.

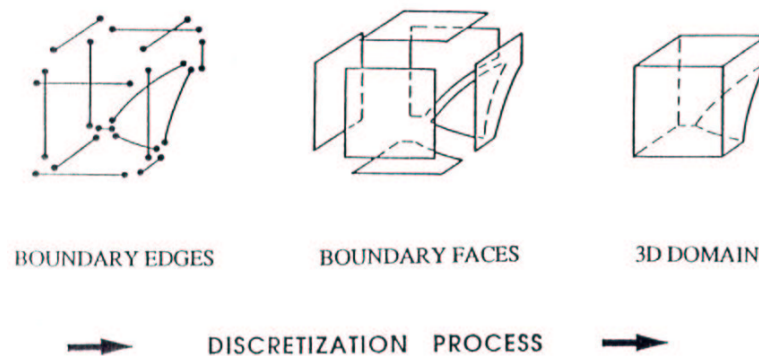


**Figure 3.** *Surface parametric representation*

The next stage consists of generating triangular planar faces. For each two-dimensional region or surface to be discretised, all the edges produced when discretising its boundary curves are assembled into the so-called initial front. The relative orientation of the curve components with respect to the surface must be taken into account in order to give the correct orientation to the sides in the initial front. The front is a dynamic data structure which changes continuously during the generation pro-

cess. At any given time, the front contains the set of all the sides which are currently available to form a triangular face. A side is selected from the front and a triangular element is generated. This may involve creating a new node or simply connecting to an existing one. After the triangle has been generated, the front is updated and the generation proceeds until the front is empty. The size and shape of the generated triangles must be consistent with the local desired size and shape of the final mesh. In the three-dimensional case, these triangles will become faces of the tetrahedra to be generated later.

For the generation of tetrahedra, the advancing front procedure is taken one step further. The front is now made up from the triangular faces which are available to form a tetrahedron. The initial front is obtained by assembling the triangulations of the boundary surfaces. Nodes and elements will be simultaneously created. When forming a new tetrahedron, the three nodes belonging to a triangular face from the front are connected either to an existing node or to a new node. After generating a tetrahedron, the front is updated. The generation procedure is completed when the number of triangles in the front is zero.



**Figure 4.** *Discretisation process*

### 2.3. *Mesh spacing control*

The inclusion of adequate mesh control is a key ingredient in ensuring the generation of a mesh of the desired form. Control over the characteristics is obtained by the specification of the spatial distribution of mesh parameters by means of a background mesh. The background mesh is used for interpolation purposes only and is made up of triangles in two dimensions and tetrahedra in three dimensions. A transformation matrix  $\mathbf{T}$ , which is related to a metric matrix, is defined at the nodes of the background mesh. At any point within an element of the background grid, the transformation  $\mathbf{T}$

is computed by linearly interpolating its components from the element nodal values. The background mesh employed must cover the region to be discretised. The generation process is always carried out in the normalised space. The transformation  $\mathbf{T}$  is repeatedly used to transform regions in the physical space into regions in the normalised space. In this way, the process is greatly simplified, as the desired size for a side, triangle or tetrahedron in this space is always unity. After the element has been generated, the coordinates of the newly created point, if any, are transformed back to the physical space by using the inverse transformation.

The construction of a background grid type control of the spacing is not an easy task when dealing with mesh generation for complex three-dimensional geometries. Alternative approaches to control the grid spacing have been proposed like those based on the use of sources distribution (Weatherhill *et al.*, 1994) or elementary solids (Formaggia, 1991). All these methods use the same information for both the discretisation of the surfaces and the volume. It may be advantageous to take into account two background meshes: one related to the discretisation of the surfaces and the other related to the three-dimensional domain discretisation, the latter being built from the knowledge of the surfaces discretisation provided through the first one (Ghidoni *et al.*, 2006).

### 3. Geometry modelling system

A geometrical data base is extracted from the CAD model. It contains the coordinates of the corners, the mathematical definition of support curves and surfaces, and topological relations that describe the links between the different geometrical entities. The information contained in the data base are in particular used by the surface grid generator that provides in addition to nodes coordinates and elements connectivities the parametric coordinates of the nodes related to the support curves and surfaces description. Those quantities represent the genuine tie between the geometrical mathematical model and the geometry discretisation. By maintaining frozen the parametric coordinates it is possible to update automatically the geometry discretisation when its mathematical description is modified. Note that, except for the original model description, grid consistency is not ensured, but a first guess is obtained.

Each support surface patch (curve arc) may be defined with respect to control nodes. The control nodes are introduced within the model in such a way that, for each patch (arc), Equation [9] (Equation [4]) can be solved with respect to the components of the matrix (vector)  $\mathbf{A}$  assuming that the values of the vectors  $\mathbf{r}$ ,  $\mathbf{U}$  and  $\mathbf{V}$  have been previously defined. As an example, a polynomial of rank  $n$  may be described by using  $n + 1$  control points. Thus, any modification of the components of the vector  $\mathbf{r}$  of the control points, assuming that the related values of vectors  $\mathbf{U}$  and  $\mathbf{V}$  are frozen, will result in a new set of values for the components of  $\mathbf{A}$  and consequently in a new mathematical representation of the surface patch (curve arc). Within an optimisation process, the control nodes will be directly associated to the parametrisation. A



modification of support surfaces means redefinition of support curves (intersection of surfaces) and of corners (intersection of support curves).

### 3.1. Surface intersection

For each control point  $i$  on the support curve, its parametric coordinates on the curve defining surfaces  $j$  (surfaces whose intersection produces the curve,  $j = 1, 2$ ) are computed as the value of the parameters that minimizes the distance between the node and the surface. This is equivalent to solve

$$[\mathbf{r}_i - \mathbf{r}_j(u, v)] \cdot \left[ \frac{\partial \mathbf{r}_j}{\partial u}, \frac{\partial \mathbf{r}_j}{\partial v} \right] = \mathbf{0} \quad [11]$$

which corresponds to find the foot of the normal at surface  $j$  dropped from node  $i$ . This is done by using the initial geometrical model description.

Starting from these initial values, it remains to solve the system of equations

$$\mathbf{r}_1(u_1, v_1) - \mathbf{r}_2(u_2, v_2^0) = \mathbf{0} \quad [12]$$

with respect to the parametric coordinates  $u_1, v_1$  and  $u_2$ , the value of  $v_2^0$  being frozen. The new coordinates of the curve control point are computed from Equation [9] and used in order to update the support curve definition.

### 3.2. Curve intersection

The corners coordinates are defined by computing the parametric coordinates of the corner defining curves  $j$  (surfaces whose intersection produces the curve) taken two by two as the value of the parameters that minimizes the distance between the two curves which is equivalent to solve

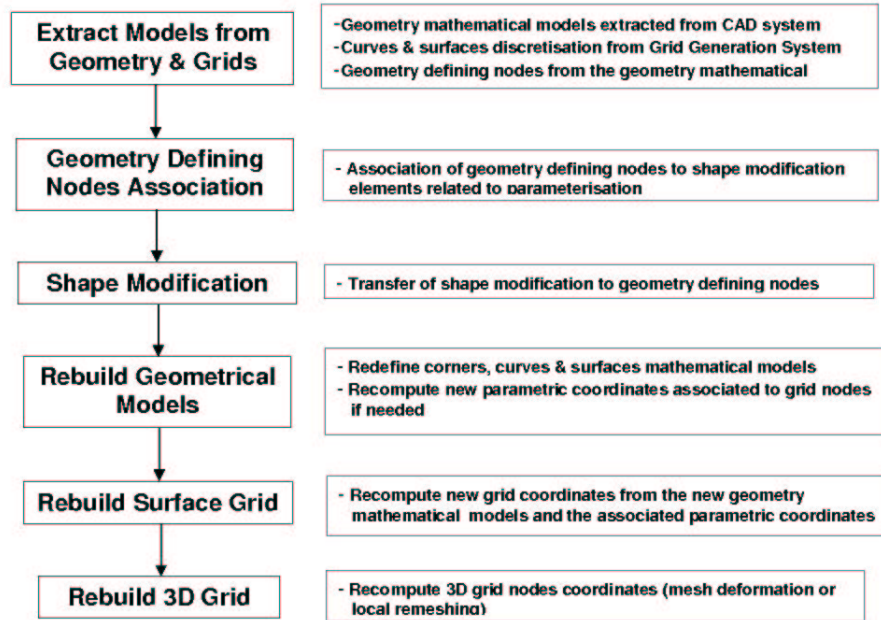
$$[\mathbf{r}_1(u_1) - \mathbf{r}_2(u_2)] \cdot \frac{\partial \mathbf{r}_j}{\partial u} = \mathbf{0}; j = 1, 2 \quad [13]$$

with respect to the parametric coordinates  $u_1$  and  $u_2$ . The new corner coordinates are computed from Equation [4].

### 3.3. Surface grid update

The geometrical model description having been updated, the new position of the grid nodes on the geometry curves and surfaces can be computed from the knowledge of their parametric coordinates. The use of the parametric coordinates associated to the original model description do not ensured grid consistency, but can be used as a first guess for a mesh optimisation process which can be applied either in the parametric

or in the physical space. The above described geometry grid update maintains the topology of the original grid unchanged and is well suited when combined with a mesh deformation technique for three-dimensional grid update. The overall process is summarised in Figure 5.



**Figure 5.** Mesh update procedure

#### 4. Mesh deformation algorithm

The modification of the grid on the body needs to be transferred to the tridimensional mesh. This can be provided by using a mesh deformation strategy which, as previously mentioned, has to be carefully designed in order to maintain the same grid quality of that of the initial grid during the optimisation process.

In the following,  $\mathbf{x}_i$  will indicate the position vector of node  $i$  and  $\mathcal{E}_i$  the set of elements belonging to the patch  $\mathcal{P}_i$  of elements surrounding a given internal node  $i$ .  $\mathcal{K}_i$  is instead the set formed by the nodes on the boundary of  $\mathcal{P}_i$ .

#### 4.1. Spring analogy based algorithm

To move the mesh, it is first assumed that each node  $i$  is connected to each adjacent node  $j$  by a fictitious spring under the force  $\mathbf{F}_{ij}$  defined by

$$\mathbf{F}_{ij} = K_{ij}(\mathbf{x}_j - \mathbf{x}_i) \quad [14]$$

where  $K_{ij}$  is the spring constant which in general will depend on some local grid features and on the flow properties for adaptation. The resulting mesh is for each node  $i$  the solution of the equilibrium system:

$$\sum_{j \in \mathcal{K}_i} \mathbf{F}_{ij} = \mathbf{0} \quad [15]$$

which is also equivalent to minimize the energy of the overall spring system

$$\frac{1}{2} \sum_{j \in \mathcal{K}_i} K_{ij} (\mathbf{x}_j - \mathbf{x}_i)^2 \quad [16]$$

Equation [15] may be rewritten as

$$\sum_{j \in \mathcal{K}_i} K_{ij} (\Delta \mathbf{x}_j - \Delta \mathbf{x}_i) = - \sum_{j \in \mathcal{K}_i} K_{ij} (\mathbf{x}_j^0 - \mathbf{x}_i^0) \quad [17]$$

where  $\Delta \mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_i^0$  represents the deviation of the actual position  $\mathbf{x}_i$  from its initial value  $\mathbf{x}_i^0$ .

In order to modify the grid in such a way to maintain basic characteristics of the original one, additional terms are introduced which modify the spring energy relation [16] as follows

$$\frac{1}{2} \sum_{j \in \mathcal{K}_i} K_{ij} [(\mathbf{x}_j - \mathbf{x}_i) - \mathbf{R}_{ij}(\mathbf{x}_j^0 - \mathbf{x}_i^0)]^2 \quad [18]$$

where  $\mathbf{R}_{ij}$  is a mesh adaptation transformation matrix. The equivalent equilibrium system equation takes the form

$$\sum_{j \in \mathcal{K}_i} K_{ij} (\mathbf{x}_j - \mathbf{x}_i) = \sum_{j \in \mathcal{K}_i} K_{ij} \mathbf{R}_{ij} (\mathbf{x}_j^0 - \mathbf{x}_i^0) \quad [19]$$

Note that in the case the following conditions occur

$$\mathbf{R} = \mathbf{I} \quad , \quad \mathbf{x}_j - \mathbf{x}_i = \mathbf{x}_j^0 - \mathbf{x}_i^0 \quad [20]$$

where  $\mathbf{I}$  denotes the unity matrix, Equation [19] is identically satisfied and the grid nodes do not move. Equation [19] is equivalent to

$$\sum_{j \in \mathcal{K}_i} K_{ij} (\Delta \mathbf{x}_j - \Delta \mathbf{x}_i) = \sum_{j \in \mathcal{K}_i} K_{ij} (\mathbf{R}_{ij} - \mathbf{I}) (\mathbf{x}_j^0 - \mathbf{x}_i^0) \quad [21]$$

where the unknowns are now expressed in terms of the deviation between the actual and initial values of the position vector  $\mathbf{x}_i$ .

For the solution of Equation [21], the following iterative algorithm may be adopted

$$\Delta \mathbf{x}_i^{l+1} = \Delta \mathbf{x}_i^l + \mathbf{R} \mathbf{H}_i^l \quad [22]$$

where

$$\mathbf{R} \mathbf{H}_i^l = \sum_{j \in \mathcal{K}_i} K_{ij} (\Delta \mathbf{x}_j^l - \Delta \mathbf{x}_i^l) - \sum_{j \in \mathcal{K}_i} K_{ij} (\mathbf{R}_{ij} - \mathbf{I}) (\mathbf{x}_j^0 - \mathbf{x}_i^0) \quad [23]$$

and  $l$  indicates the iteration numbering. The iterative process is stopped when the quantity

$$\delta \mathbf{x}_i^{l+1} = \Delta \mathbf{x}_i^{l+1} - \Delta \mathbf{x}_i^l = \mathbf{x}_i^{l+1} - \mathbf{x}_i^l \quad [24]$$

is less than a predefined small quantity.

#### 4.2. Mesh deformation due to surface grid modification

This deals in general with fluid-structure interaction and shape optimisation applications for which the modification of the surface grid has to be transferred in the overall computational domain. On the basis of the finite element approximation theory related to Lagrangian simplex elements, let denote by  $\mathbf{S}^{e,s}$  the transformation matrix (parametric to physical space) of the surface element on the body skin related to the final position vector  $\mathbf{x}$  and by  $\overline{\mathbf{S}}^{e,s}$  that related to the initial position vector  $\mathbf{x}^0$ . The transformation matrix  $\mathbf{R}$  is defined as

$$\mathbf{R} = \mathbf{S}^{e,s} \left( \overline{\mathbf{S}}^{e,s} \right)^{-1} \quad [25]$$

The matrix  $\mathbf{R}$  is first computed on the skin boundary and then propagated through the entire domain by solving the equation

$$\sum_{j \in \mathcal{K}_i} K_{ij} (\mathbf{R}_j - \mathbf{R}_i) = \mathbf{0} \quad [26]$$

while maintaining its value on the skin fixed. The initial value of the matrix  $\mathbf{R}$  is set to be the unity matrix within the overall domain.

#### 4.3. Node movement on grid boundaries

In the case nodes are allowed to move on boundaries, the node movement algorithm has to be adapted in such a way that those nodes remain on the CAD defining surfaces. The boundary nodes can be divided in three categories:

- 1) corners whose coordinates have to be frozen,
- 2) nodes that lie on curves, and
- 3) nodes that lie on surfaces.

In order to obtain the new coordinates on the boundaries, the parametric coordinates deviation associated to the CAD elements is computed by projecting the physical coordinates displacement along the tangent to the boundary and then the physical coordinates are obtained, previous the computation of the new parametric coordinates, by using Equations [2] or [7] for nodes that have to be located on curves or surfaces, respectively.

The curves are parametrised through a single parameter  $u$ . Let denote by  $u_i^l$  the parametric coordinate corresponding to the physical coordinate  $\mathbf{r}_i^l$  of the node  $i$  at iteration  $l$ . By using the relation

$$\Delta \mathbf{r} = \frac{\partial \mathbf{r}}{\partial u} \Delta u \quad [27]$$

the displacement in the parametric space at iteration  $l+1$  is computed according to the expression

$$\Delta u_i^{l+1} = \Delta u_i^l + \left( \frac{\partial \mathbf{r}}{\partial u} \Big|_i^l \cdot \mathbf{R}\mathbf{H}_i^l \right) / \left( \frac{\partial \mathbf{r}}{\partial u} \Big|_i^l \cdot \frac{\partial \mathbf{r}}{\partial u} \Big|_i^l \right) \quad [28]$$

where

$$\frac{\partial \mathbf{r}}{\partial u} \Big|_i^l \equiv \frac{\partial \mathbf{r}}{\partial u} \Big|_{u_i^l} \quad [29]$$

On the surfaces, we can define the parametric coordinates  $u$  and  $v$ . Let denote by  $u_i^l$  and  $v_i^l$  the parametric coordinates corresponding to the physical coordinates  $\mathbf{r}_i^l$  of the node  $i$  at iteration  $l$ . In a similar way that for curves, starting from the relation

$$\Delta \mathbf{r} = \frac{\partial \mathbf{r}}{\partial u} \Delta u + \frac{\partial \mathbf{r}}{\partial v} \Delta v \quad [30]$$

the displacement in the parametric space at iteration  $l+1$  is computed according to the expression

$$\Delta \begin{pmatrix} u \\ v \end{pmatrix}_i^{l+1} = \Delta \begin{pmatrix} u \\ v \end{pmatrix}_i^l + (\mathbf{G} \Big|_i^l)^{-1} \left[ \frac{\partial \mathbf{r}}{\partial u} \Big|_i^l, \frac{\partial \mathbf{r}}{\partial v} \Big|_i^l \right]^T \cdot \mathbf{R}\mathbf{H}_i^l \quad [31]$$

where

$$\frac{\partial \mathbf{r}}{\partial u} \Big|_i^l \equiv \frac{\partial \mathbf{r}}{\partial u} \Big|_{(u_i^l, v_i^l)}, \quad \frac{\partial \mathbf{r}}{\partial v} \Big|_i^l \equiv \frac{\partial \mathbf{r}}{\partial v} \Big|_{(u_i^l, v_i^l)} \quad [32]$$

and

$$\mathbf{G} = \begin{bmatrix} \frac{\partial \mathbf{r}}{\partial u} & \frac{\partial \mathbf{r}}{\partial v} \end{bmatrix}^T \begin{bmatrix} \frac{\partial \mathbf{r}}{\partial u} & \frac{\partial \mathbf{r}}{\partial v} \end{bmatrix} \quad [33]$$

is the first fundamental matrix of the surface.

## 5. Parametrisation

The parametrisation may address the direct representation of the geometry or the modification of an initial shape, the latter resulting at our knowledge more flexible and more accurate. The initial shape is recovered when the perturbation is equal to zero. Four levels of parametrisation are presently implemented in the system, that are characterised by the following attributes.

### 5.1. Node

It corresponds to a rotation around an axis complemented by a 3D translation. The set of design variables are formed by the angle of rotation and the displacement associated to the translation. The corresponding geometry modification for a node  $i$  on the geometry can be described according to

$$\Delta \mathbf{r}_i = \mathbf{x}_c + \cos \theta (\mathbf{r}_i^0 - \mathbf{x}_c) - \sin \theta [(\mathbf{r}_i^0 - \mathbf{x}_c) \wedge \mathbf{b}] + \Delta \mathbf{r}_t \quad [34]$$

In the previous relation,  $\mathbf{r}_i^0$  is the position of node  $i$  in the initial grid,  $\theta$  is the angle of rotation,  $\mathbf{b}$  is the unit vector along the rotation axis,  $\Delta \mathbf{r}_t$  is the displacement vector related to the imposed translation and  $\mathbf{x}_c$  is a point on the axis defined as

$$\mathbf{x}_c = \mathbf{x}_0 + [(\mathbf{r}_i^0 - \mathbf{x}_0) \cdot \mathbf{b}] \mathbf{b} \quad [35]$$

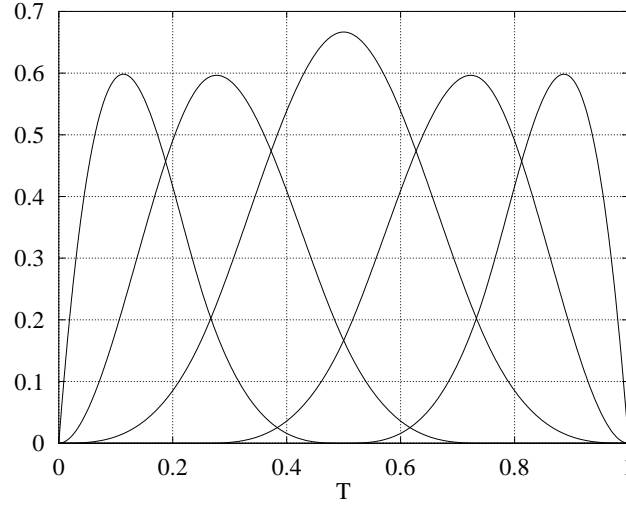
where  $\mathbf{x}_0$  is a fixed point along the axis.

### 5.2. Line

It corresponds to a perturbation function which acts along a line and is built as a linear combination of basis functions. The perturbation is added to the initial geometry along the normal vector of the surfaces linked to the line definition or transferred along selected coordinates axis. The design variables are the coefficients of the expansion. The perturbation function can be described according to

$$q_i(u) = \sum_i^n \Delta q_i H_i^n(u) \quad [36]$$

where  $u$  represents the parametric coordinate along the line and  $H_i^n$  are general shape modification functions. The design variables are the parameters  $\Delta q_i$ . Various shape functions, including Hicks-Henne functions (Hicks *et al.*, 1978), Bernstein polynomials and B-spline functions (Farin, 1989), may be used. Among those, the approach based on B-splines is preferred whereas the functions have a more compact support and allow to modify only locally the shape (Figure 6).



**Figure 6.** Example of B-spline functions

### 5.3. Surface

It corresponds to a perturbation function which is computed as a tensor product of linear combination of 1D basis functions. The perturbation is added to the initial geometry along the normal vector of the surfaces linked to the parametrisation. The design variables are the coefficients of the expansions.

$$q_s(u, v) = q_l^u(u) q_l^v(v) \quad [37]$$

where  $q_l^u$  and  $q_l^v$  corresponds to perturbation functions along the  $u$  and  $v$  coordinate axis, respectively. As alternative, we might take the following representation

$$q_s(u, v) = \sum_i^n \sum_j^m \Delta q_{ij} H_i^n(u) H_j^m(v). \quad [38]$$

### 5.4. Master geometry

A master represents a fine discretisation of a basic component. The deviation between the actual master and that associated to the initial geometry provides a geometrical perturbation to be transferred to the surfaces linked to the parametrisation. The design variables are the parameters that describe the master. An example of master parametrisation is given in the next section.

## 6. Wing parametrisation

The wing parametrisation described in this section has been designed on the basis of the following consideration:

- 1) design variables might be selected in such a way they are close to engineers that will use the automatic optimisation system,
- 2) semi-empirical knowledge is often parametrised with respect to geometrical quantities and it could be advantageous at least to exploit this knowledge through the implementation of linear inequality constraints.

### 6.1. Wing sections parametrisation

The present parametrisation is based on the splitting of an airfoil normalised by its chord in its thickness (symmetric part, Figure 7) and camberline (antisymmetric part or mean line) distributions (Figure 8). The thickness distribution is approximated by two B-splines whose curve defining polygons ( $\mathbf{P}_i [i = 0, \dots, 4]$  and  $\mathbf{P}_i [i = 4, \dots, 8]$ , respectively) are built in order to satisfy the following conditions:

- 1)  $\mathbf{P}_0 = (x_0, z_0) = (0, 0)$  and  $\mathbf{P}_8 = (x_8, z_8) = (1, 0)$ ;
- 2)  $\mathbf{P}_4 = (x_{tmax}, tmax/2)$  where  $x_{tmax}$  and  $tmax$  are the position of the maximum thickness and its value, respectively;
- 3)  $\mathbf{P}_3$  and  $\mathbf{P}_5$  are fixed by prescribing the continuity of the two B-splines at node 4, but also the continuity of their slope (equal to zero) and of their curvature, denoted by  $crv4$ ;
- 4) the abscissa of nodes 1 and 2 are prescribed by the knowledge of the slope at node 0 ( $\infty$  when  $x_1 \rightarrow 0$ ) and of the curvature radius at the leading edge, denoted by  $r_0$ . Their ordinates, denoted by  $z_1$  and  $z_2$ , respectively, are left free;
- 5)  $\mathbf{P}_7$  is prescribed by the knowledge of the trailing edge angle ( $\phi_{te}$ ) and of the distance between nodes 7 and 8, denoted by  $d_{78}$ ;
- 6) the co-ordinates of node 6 are left free.

In brief, the thickness distribution is approximated by B-spline representations whose parameters are:

$$r_0, z_1, z_2, x_{tmax}, tmax, crv_4, x_6, z_6, \phi_{te}, d_{78}. \quad [39]$$

Within the optimisation system, the geometry modification is performed by using perturbations of the initial shape and differences between initial and actual approximation of the thickness distribution are computed.

In the case of the camberline, a B-spline curve perturbation of its initial shape is directly used. The parameters are the co-ordinates  $\mathbf{Q}_i$  of the polygon defining knots of the spline.



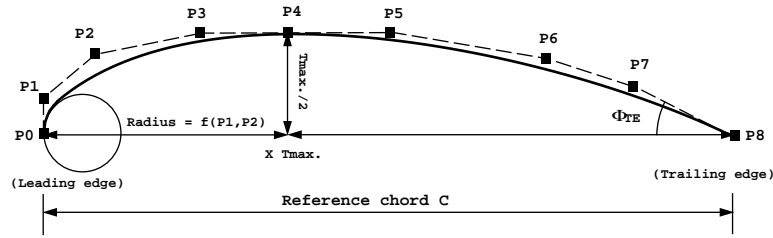


Figure 7. Thickness distribution parameters

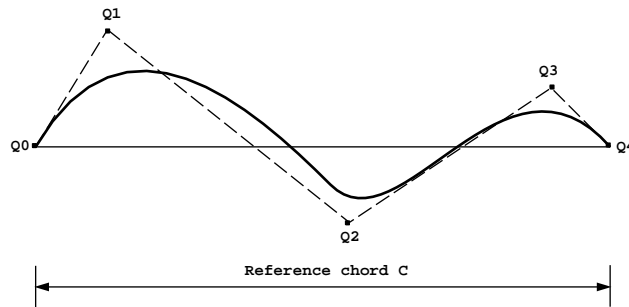


Figure 8. Camber modification parameters

In short, the node position  $\mathbf{q}(t)$  on the wing section shape can be modified by using the following formula

$$\mathbf{q}(t) \equiv (x(t), z(t)) = \mathbf{q}^*(t) + \sum_i^n (\mathbf{P}_i - \mathbf{P}_i^*) B_i^n(t) + \sum_j^m \mathbf{Q}_j B_j^m(t) \quad [40]$$

where the symbol \* refers to the original shape,  $t \in [0, 1]$  is the parametric abscissa and  $B_i^n$  are the B-spline functions of order  $n$ .

### 6.2. Wings parametrisation

The wing is in general defined by:

1) platform defining parameters, such as wing area ( $S$ ), wing aspect ratio ( $AR$ ), wing sweep angle ( $Sw$ ) and the wing taper ratio ( $TR$ ) at some spanwise stations. Those parameters are not in general all independent;

2) global wing section shape defining parameters, such as the twist ( $Tw$ ) and the maximum thickness ( $Tmax$ ) of key wing sections;

3) local wing section shape defining parameters, that define the local shape of key wing sections. They may be represented by relation [40].

Each key wing section is defined with respect to a local frame, centred at the section leading edge and having the x-axis coincident with the section chord. Normalised coordinates are introduced by scaling each length with the chord length, which means that  $x \in [0, 1]$ . Those sections without a design variables specified value are modified from the design variables associated to neighbouring sections by using a linear relationship. It is recommended to apply the linear interpolation to the deviations between initial and modified geometry and not directly to the latter.

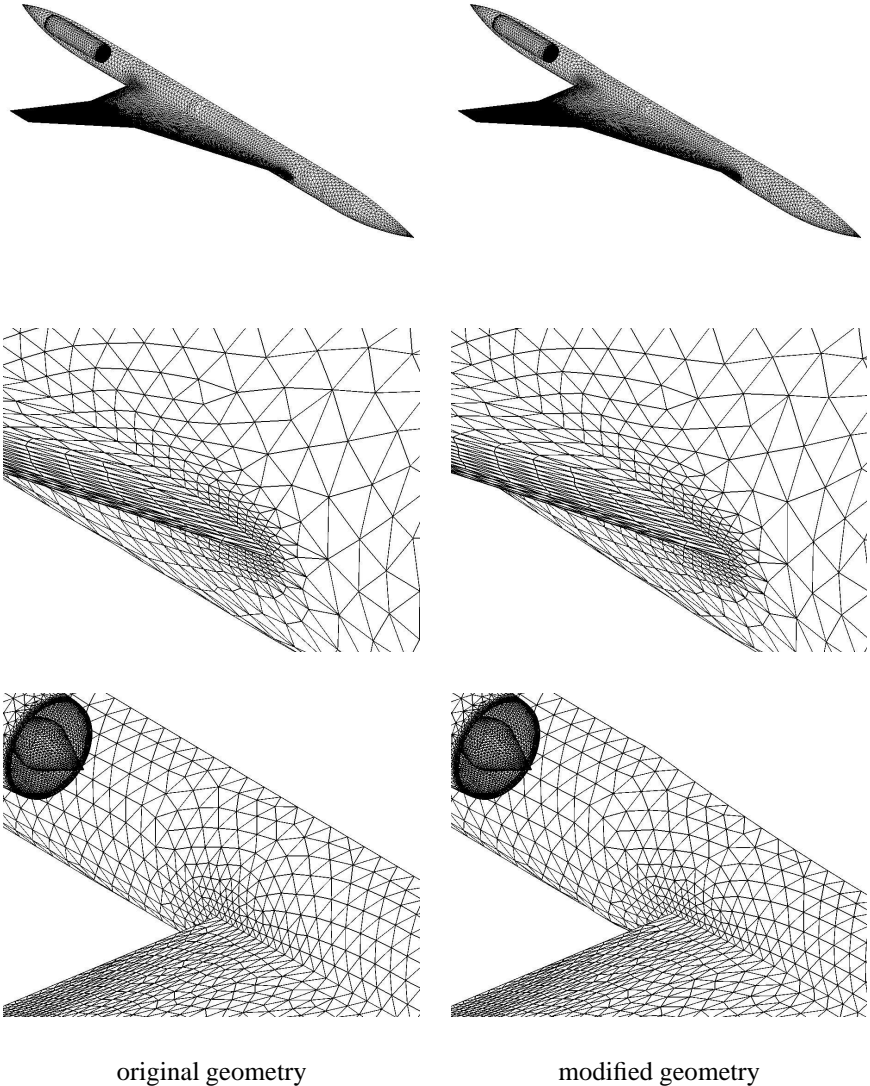
Starting from the knowledge of the values of the wing defining parameters (modified or frozen values), a model generator will provide a new representation of the wing shape, that corresponds to a fine geometry discretisation, named master discretisation. By maintaining the same number of discretisation nodes, the shape modification may be easily computed as the deviation of the node coordinates between the actual master and the initial one.

The technique described in this section can be applied to any wing-like geometry, such as wings, horizontal and vertical tails and pylons.

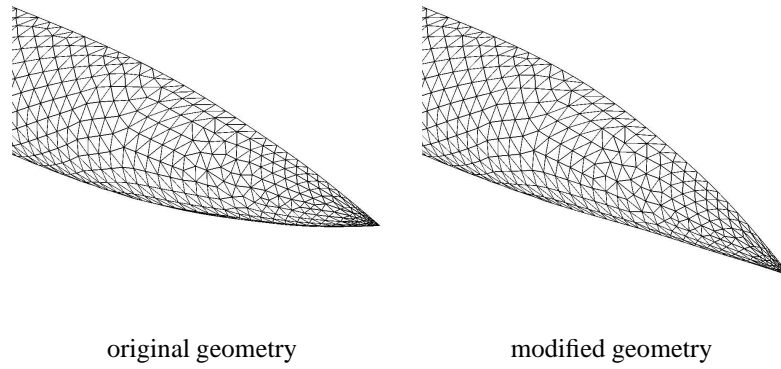
## 7. Conclusions

A geometry modelling technique has been described. Instead to use the parametrisation capability of modern CAD systems, we prefer to manipulate a mathematical geometry model extracted from them. This choice is dictated by the necessity to genuinely control the overall optimisation chain and to be able to implement within the system specific parametrisation technique based on the user experience. In particular, it is thought that the use of parametrisations based on the modification of an existing shape instead to exploit the direct representation of the geometry results in an increase of flexibility and accuracy. As the geometrical model is the same that the one used for grid generation purposes, it allows to easily transfer the geometry modification to the computational grid. Particular attention has to be paid to the overall accuracy of the grid update procedure.

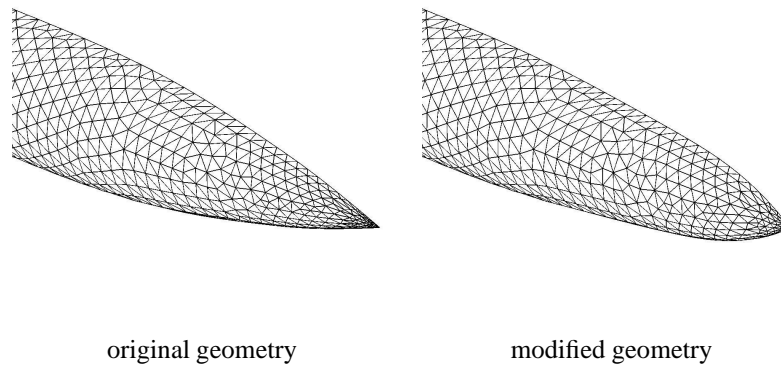
Figures 9 to 12 illustrate the capability of the geometrical modelling system. Fuselage sections reduction that implies recomputation of the wing/body intersection and several front fuselage modifications are presented.



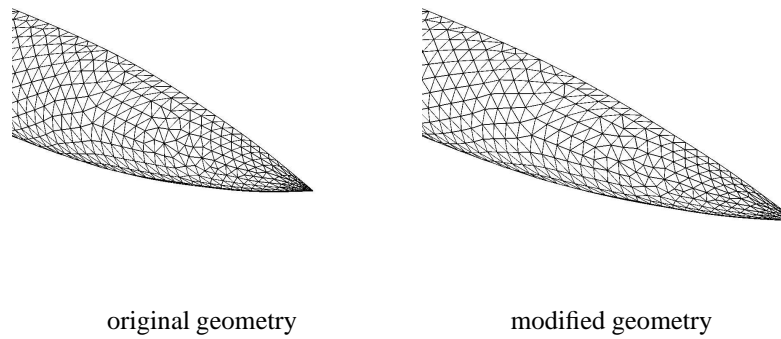
**Figure 9.** *Fuselage section modification*



**Figure 10.** *Front fuselage deflection*



**Figure 11.** *Front fuselage modification*



**Figure 12.** *Front fuselage stretching*

## 8. References

- Batina J., "Unsteady Euler airfoil solutions using unstructured dynamic meshes", *AIAA paper 89-0115, 27th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 1989.
- Castonguay P., Nadarajah S., "Effect of Shape parametrization on aerodynamic shape optimisation", *AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2007.
- Daoud F., Firl M., Bletzinger K., "Filter Techniques in Shape Optimisation with CAD-Free Parametrisation", *6th World Congress of Structural and Multidisciplinary Optimisation*, Rio de Janeiro, 30 May-03 June, Brazil, 2005.
- Daumas L., Dinh Q., Kleinvelde S., Rogé G., "CAD-Based Wing-Body Optimisation", *CEAS Aerospace Aerodynamics Research Conference*, Cambridge, 2002.
- Degand C., Farhat C., "A three-dimensional torsional spring analogy method for unstructured dynamic meshes", *Computers and Structures*, vol. 80, p. 305-316, 2002.
- Désidéri J.-A., "Hierarchical Optimum Shape Algorithm using Embedded Bézier Parametrizations", *Numerical Methods for Scientific Computing Variational Problems and Applications*, CIMNE, Barcelona, Y. Kuznetsov et al eds., 2003.
- Désidéri J.-A., Janka A., "Multilevel Shape Parameterization for Aerodynamic Optimisation - Application to Drag and Noise Reduction of Transonic/Supersonic Business Jet", *ECCOMAS 2004 Conference*, Jyväskylä, Finland, P. Neittammaki et al eds., 2004.
- Farin G., *Curves and Surfaces for Computed Aided Design*, Academic Press, 1989.
- Formaggia L., "An Unstructured Mesh Generation Algorithm for Three Dimensional Aeronautical Configurations", *Numerical Grid Generation in Computational Fluid and Related Fields*, North Holland, A.S. Arcilla et al eds., 1991.
- Fudge D., Zingg D., Haimes R., "A CAD-Free and a CAD-Based Geometry Control Based System for Aerodynamic Shape Optimisation", *AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2005.
- George P.-L., *Automatic Mesh Generation, Application to Finite Element Methods*, Wiley, New York, 1991.
- Ghidoni A., Pelizzari E., Rebay S., Selmin V., "3D Anisotropic Unstructured Grid Generation", *Int. J. Numer. Methods Fluids*, vol. 51, n° 9-10, p. 1097-1115, 2006.
- Hicks R., Henne P., "Wing Design by Numerical Optimisation", *Journal of Aircraft*, vol. 15, n° 7, p. 407-413, 1978.
- Mohammadi B., Pironneau O., "Shape Optimisation in Fluid Mechanics", *Annual Review of Fluid Mechanics*, vol. 36, p. 255-279, 2004.
- Nielsen E., Anderson W., "Recent improvements in aerodynamic design optimization on unstructured meshes", *AIAA Journal*, vol. 40, n° 6, p. 1155-1163, 2002.
- Peraire J., Morgan K., Peiro J., *Unstructured Mesh Method for CFD*, Von Karman Institute Lecture Series 1990-06, 1990.
- Samareh J., "A Survey of Shape Parametrisation Techniques", *AIAA Journal*, vol. 39, n° 5, p. 877-884, 2001.
- Samareh J., "Aerodynamic Shape Optimisation based on free-form deformation", *AIAA-2004-4630*, 2004.
- Weatherhill N., *Grid Generation*, Von Karman Institute Lecture Series 1990-06, 1990.

Weatherhill N., Marchant M., Hassan O., Marcum D., “ Grid Adaptation using a Distribution of Sources Applied to Inviscid Compressible Flow Simulations”, *Int. J. Numer. Methods Fluids*, vol. 19, p. 739-764, 1994.