ELSEVIER

Comput. Methods Appl. Mech. Engrg. 188 (2000) 727–742

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Shape variable definition with C^0, C^1 and C^2 continuity functions

G. Chiandussi [a], G. Bugeda [b,*], E. Oñate [b]

[a] Department of Mechanical Engineering, Technical University of Turin, Corso Duca degli Abruzzi 24, 10129, Torino, Italy
[b] CIMNE, International Center for Numerical Methods in Engineering, Edificio C1, Campus Norte UPC, Gran Capitàn s/n, 08034, Barcelona, Spain

Received 30 March 1999

Abstract

The present paper proposes a new technique for the definition of the shape design variables in 2D and 3D optimisation problems. It can be applied to the discrete model of the analysed structure or to the original geometry without any previous knowledge of the analytical expression of the CAD defining surfaces. The proposed technique allows the surface continuity to be preserved during the geometry modification process to be defined a priori. This capability allows for the definition of shape variables suitable for every kind of discipline involved in the optimisation process (structural analysis, fluid-dynamic analysis, crash analysis, aerodynamic analysis, etc.). © 2000 Elsevier Science S.A. All rights reserved.

Keywords: Shape variable definition; CAD geometry; Multidisciplinary optimisation; Surface continuity preservation; Finite element method

1. Introduction

The correct definition of shape variables is of the utmost importance in the layout of an optimisation problem. The identification of the most useful geometry changes is crucial if a satisfactory solution is required. The shape variable definition task has to overcome different difficulties like the selection of the admissible geometrical changes and the preservation of several geometrical properties as well as geometrical continuity between fixed areas and areas subject to deformation.

One of the first researchers to point out the complexity and the difficulties in the shape variable definition was Imam. In his work [1] he identifies four main techniques for variable definition: (1) the independent node movement technique, (2) the design element technique, (3) the supercurve technique and (4) the shape superposition technique.

The independent node movement technique is very simple; it uses as shape variables the co-ordinates of the nodes of the discrete model of the structure.

The design element technique has been extensively used. The design element is a macro finite element consisting of one or more finite elements. The isoparametric shape representation, as used for individual elements, is used here to determine the isoparametric co-ordinates of any point on the surface. This technique also allows for the determination of the co-ordinates of the node points inside the design element for mesh generation, which is required every time the shape is changed during optimisation.

* Corresponding author.

0045-7825/00/$ - see front matter
PII: S0045-7825(99)00358-8

The third technique mentioned by Imam is the supercurve one. In structured meshes the boundary surface is fully determined given a few curves laying on it. Only the points on these curves are required to construct the finite element model. In such cases, the parametric representation of curves with a polynomial expression may be used; the coefficients of the polynomial expression serve in this case as the shape variables.

Finally, the shape superposition technique concerns the possibility to superimpose two or more shapes specified in terms of model location of points on a curve or surface in varying proportions. The linear combination of these predefined shapes allows the generation of a variety of shapes, the coefficients of the combination being the shape variables [5]. A similar technique has been used also by Vanderplaats [2] for airfoil optimisation.

In 1984, Braibant and Fleury [3] proposed a new approach for two-dimensional shape variable definition. The shape description is quite similar to the one based on the design element introduced by Imam. The region of the structure to be modified during the optimisation process is also defined by one or more design elements that still contain a part of the mesh. Instead of using the shape functions of a two-dimensional finite element, blending functions typical of Bezier or B-spline curves are used to determine the co-ordinates of any point inside the design element or on its boundaries. Therefore the shape variables are no longer the position of the nodes of an isoparametric two-dimensional element, but the points that control two families of curves whose cartesian product defines the design element. Slope discontinuities at the design element interfaces can be easily managed. Each boundary of the structure is an automatic piecing of spline of degree k. The value of k can be selected by the designer and this formulation guarantees the continuity of the piecing up to the order $k-1$ [4].

The five techniques mentioned above have several limitations. Let us consider first the node movement and the supercurve techniques. The former is practically unusable due to two severe drawbacks: it results in too many design variables and discontinuity of slope in the shape at the element interfaces takes place in the first few iterations of the process. The latter requires the identification of the parametric equations of several curves. Several practical applications can be found in two-dimensional problems concerning, for example, thickness distribution over a shell structure. Moreover, in three-dimensional problems, this technique requires a non-trivial work that is not compensated by the higher order surfaces that can be obtained.

These two techniques are not so common and are usually substituted by the curve superposition and the design element ones. In the former, difficulties arise in the selection and evaluation of the 'master' curves or surfaces whose linear combination will allow identifying the optimal solution. These are usually taken as the first modal shapes of the structure or the shapes resulting by the application of user-defined distributed loads. Moreover, in this case it is difficult to control the surface modification of the structure especially if only local variations are to be considered. The design domain technique is one of the most useful and practical techniques indeed. It avoids the presence of discontinuities at the element interfaces and allows for the reduction of the number of the design variables. The limitations of this technique come from the interpolation functions used to extrapolate the design element nodal movements to all nodes present inside the design element itself. Isoparametric shape functions are used, and because of their origin, they can guarantee only zero continuity between adjacent elements. So, there is not any possibility to choose the continuity degree between two adjacent design elements. Moreover, only the nodes inside the design element are held in consideration for nodal movement extrapolation. If the design element dimensions are not large enough, a discontinuity in the movement distribution could be introduced leading to element distortion and to inaccurate results.

The technique proposed by Braibant and Fleury solves the lack of continuity control at the interface of two adjacent domains by introducing Bezier or B-spline interpolation functions. This approach can be easily adopted for two-dimensional problems, but difficulties arise in its extrapolation to three-dimensional ones. In any case the limitation linked to the movement distribution discontinuity due to the design element concept as introduced by Imam keep on being present.

All the mentioned techniques are characterised by a common problem: they are based on the discrete model of the structure analysed.

The present paper proposes a new shape design variable technique. It allows for defining the shape variables directly on the geometry of the analysed structure without any previous knowledge of the analytical expression of the CAD surfaces. The design element concept has been introduced and modified leading to radical simplifications. In three-dimensional problems, the domains are defined by two-dimensional surfaces whereas, in two-dimensional problems, the domains are defined by one-dimensional curves. Only the 'grouping' capability has been preserved. Also the interpolation functions for nodal movement extrapolation have been modified being defined over a lower degree domain. The proposed technique allows for an a priori definition of the surface continuity to be preserved during the geometry modification process. This capability allows for the definition of shape variables suitable for every kind of discipline involved in the optimisation process in structural analysis, fluid-dynamic analysis, crash analysis, and aerodynamic analysis, among others.

2. The method

The proposed method is able to manage two-dimensional as well as three-dimensional shape variable definition problems. The way to operate is exactly the same in the two situations, the only difference being the degree of the geometrical entities. Whereas in a three-dimensional problem the geometry of the analysed structure is described using two-dimensional surfaces, in two-dimensional problems this is obtained by using one-dimensional curves.

2.1. Surface modification

Let us consider the three-dimensional geometric model of a structure or a fluid domain surrounding an immersed object. The geometrical model is usually defined by several patches linked together. It is possible to group the patches to define one or more macro-patches. This subdivision allows for the identification of the portion of the boundary surface of the structure that will be affected by shape modifications and the one that will be not. The former can be composed of several macro-patches too. Very often, the mathematical definition of these patches is unknown or it is very complex and outside the scope of the numerical analysis. In these cases it is impossible to obtain the parametric equation $r(u, v)$ of the macro-patches, and consequently, to evaluate the parametric co-ordinates describing the location of the points belonging to them.

In order to obtain, at least, the approximate parametric co-ordinates of a point belonging to a macro-patch, a new patch approximating the real one is built by using the information on the co-ordinates of the vertices of the macro-patch. This defines a plane passing through three points if the macro-patch is a triangle, or a curved surface (Coon patch [7,11]) if the macro-patch is a quadrilateral surface. The projection of every point belonging to the macro-patch over the new one along its normal can now be evaluated. The projection location over the approximating patch as well as the approximating patch equation can be transformed from the (x, y, z) reference system to a (u, v) parametric one.

In the proposed approach, the shape variables are defined by using vectors applied to the vertices of the macro-patches previously defined. The co-ordinates (x', y', z') of each point belonging to every macro-patch after shape modification are obtained through the application of the following equation:

$$(x', y', z') = (x, y, z) + \sum_{i=1}^{nd} d_i \cdot \sum_{j=1}^{nv} \mathbf{v}_{ij} \cdot \sum_{k=1}^{np} \mathbf{r}_{jk}(x, y, z), \tag{1}$$

where nd is the total number of design variables, nv the total number of vertices of the macro-patch, np the total number of macro-patches one of whose the vertices is $\mathbf{v}_{ij}$, d_i the value of the ith design variable, $\mathbf{v}_{ij}$ a vector related to the design variable i and applied at the vertex j (vector $\mathbf{v}_{ij}$ is defined in direction and magnitude) and $r_{jk}(x, y, z)$ a function defined over the patch approximating the macro-patch k to interpolate the movement of the vertex $\mathbf{v}_{ij}$.

A set of standard interpolation functions with C^0, C^1 and C^2 continuity have been identified. Every interpolation function is described by using a parametric equation as will be seen in Section 2.2. The knowledge of the parametric equation of the approximating patch allows to relate it to the standard interpolation function. Consequently, it is possible to identify for every point projection on the approximating patch a point on the interpolating function describing the effect of the application of a vertex

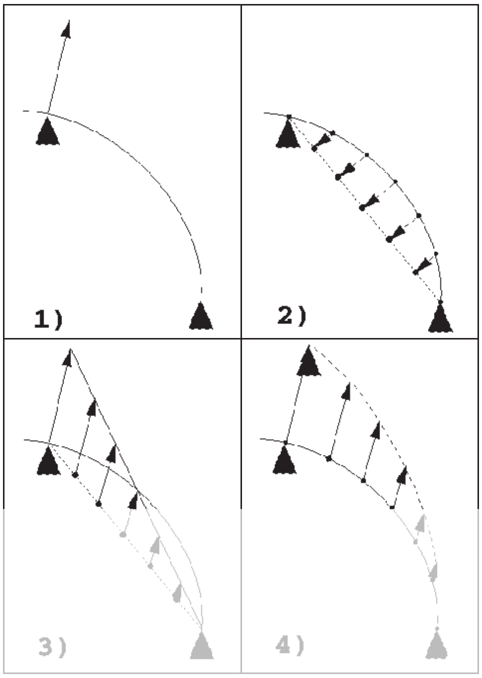


Fig. 1. The four steps of the propagation of the vertex movement on a one-dimensional macro-patch.

movement. The standard interpolation functions are defined considering a unit vertex movement. Consequently, the total effect of a non unitary vertex movement can be obtained by a simple multiplication.

Fig. 1 describes this process by using a two-dimensional example for the sake of simplicity. In this case the macro-patch is a curve and the approximating patch is a straight line connecting the two vertices. The propagation of the vertex movement on the one-dimensional macro-patch can be described by using four pictures: (1) the initial situation (a patch with a vector describing the module and direction of the vertex movement), (2) the projection of the macro-patch points on the approximating patch, (3) the extrapolation of the vertex movement to all points with reference to a C^0 continuity interpolation function, and (4) application of the point movement to the original points by using expression (1).

With this approach the new position of each point of a macro-patch is defined in terms of the original geometry plus the superposition of some curved patches. The original geometrical definition is taken as a starting point in the shape modification process and is not needed any more. The shape modifications are controlled in direction and magnitude by the design variables via the vectors applied at the vertices of the macro-patches.

The design element concept introduced by Imam [1] has been recovered here and used in a slightly different way. The design element of Imam identifies a portion of the structural space subject to shape variation, whereas in the new concept it identifies only a portion of the boundary surface of the structure whose shape is modified by the application of a shape variable.

The information introduced with the definition of the shape variables needs to be transferred from the geometrical model to the finite element model. The surface modifications induced by the design variables are transformed in a displacement field of the nodes of the discrete model belonging to the macro-patches (here the term displacement refers to the movement of a node during the shape modification process). To avoid strong distortion effects, it is necessary to propagate the displacement field defined at the boundary surfaces to all internal nodes of the discrete model. Also, in this situation it is necessary to use an interpolation function. The absence of specific requirements allows for using directly the linear shape functions typical of the finite elements. The movement of the mesh points can now be obtained by computing the displacements of a fictitious linear elastic structure formed by the mesh elements under prescribed boundary conditions [13].

2.2. Boundary movement propagation

The identification of the deformed shape of a macro-patch due to the application of a movement vector to one of its vertices and the transfer of the information so obtained into a displacement field of the nodes of the discrete model belonging to it are performed together.

The co-ordinates (x, y, z) of the nodes of the discrete model lying on the surface of the structure are approximated with the parametric co-ordinates (u, v) of their projection over the approximating patch. These parametric co-ordinates are used to identify the effect of the application of an imposed displacement to any vertex of the macro-patch by keeping as a reference a parametric standard surface with parametric co-ordinates:

$$0 \leqslant u, \quad v \leqslant 1. \tag{2}$$

Every reference surface or, more simply, interpolation function r, is characterised by a unit value at one of the vertices and a null value at all the other ones. Then, for every macro-patch it is possible to identify as many interpolation functions r as the number of the vertices. The sum of the interpolation functions is constant and equal to one. This allows us to define the shape variables leading to the displacement of a portion of the boundary surface parallel to itself.

The interpolation functions r are defined to keep certain continuity properties of the boundary surface of the analysed domain. The C^0 continuity interpolation function is characterised by a null value in correspondence to the sides opposite to the unit value vertex, the C^1 continuity interpolation function has a null first derivative and the C^2 continuity interpolation function has a null second derivative. The smoothness of the boundary surface of the structure requires continuity of the desired degree to be preserved at the boundaries between different macro-patches. This implies 0, 1 and 2 cross-boundary continuity degree respectively for the C^0, C^1 and C^2 continuity interpolation functions. The requirement of unit sum and the consequent requirement of symmetry lead to ask for cross-boundary derivatives continuous and equal to zero.

2.2.1. One-dimensional interpolation functions

The identification of the analytical expressions of the interpolation functions in parametric co-ordinates is quite straightforward in the one-dimensional case. The C^0 interpolation function is simply a first degree polynomial:

$$r(u) = u, \tag{3}$$

the C^1 interpolation function a third degree polynomial:

$$r(u) = 3u^2 - 2u^3 \tag{4}$$

and the C^2 interpolation function a fifth degree polynomial in the unique parametric co-ordinate u (Fig. 2):

$$r(u) = 10u^3 - 15u^4 + 6u^5. \tag{5}$$

They satisfy the requirement of constant unit sum all over the definition domain.

2.2.2. Two-dimensional interpolation functions: quadrilateral surfaces

The identification of the interpolation function in the two-dimensional case is more complex. Let us consider first a quadrangular patch. A curve connecting each pair of adjacent vertices can be identified. The resulting network of curves identifies a rectangular patch, which has as its boundaries two u-curves and two v-curves (Fig. 3). It is assumed that u and v run from 0 to 1 along the relevant boundaries. Then $r(u, v)$ represents the interior of the surface patch, while $r(u, 0)$, $r(1, v)$, $r(u, 1)$ and $r(0, v)$ represent the four known boundary curves. The problem of defining a surface patch, then, is that of finding a suitably well-behaved function $r(u, v)$ which reduces to the correct boundary curve when $u = 0$, $u = 1$, $v = 0$ or $v = 1$. This surface patch can be obtained as boolean sum of two surfaces interpolating the $r(u, 0)$ and $r(u, 1)$ along the v direction and the $r(v, 0)$ and $r(v, 1)$ along the u direction (Coon patch). The resulting surface patch is conveniently expressed in the matrix form by [6,13]:

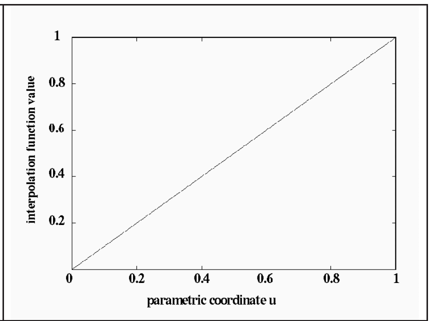
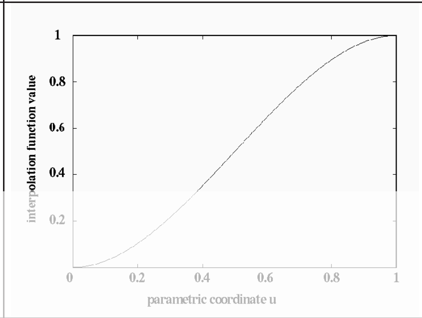
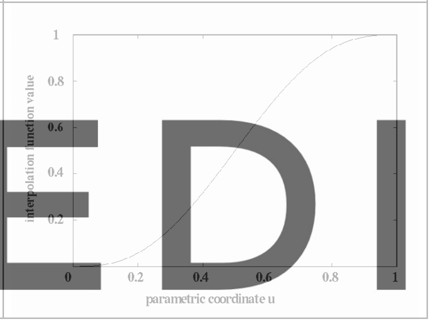
$f(u) = u$	
$f(u) = 3u^2 - 2u^3$	
$f(u) = 10u^3 - 15u^4 + 6u^5$	

Fig. 2. C^0, C^1 and C^2 interpolation functions for the mono-dimensional domain with corresponding analytical expressions.

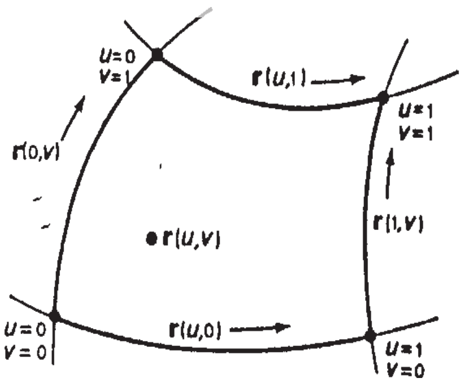


Fig. 3. Coon patch definition through two families of parametric curves.

$$r(u,v) = [1-u \quad u] \cdot \begin{bmatrix} r(0,v) \\ r(1,v) \end{bmatrix} + [r(u,0) \quad r(u,1)] \cdot \begin{bmatrix} 1-v \\ v \end{bmatrix} - [1-u \quad u] \cdot \begin{bmatrix} r(0,0) & r(0,1) \\ r(1,0) & r(1,1) \end{bmatrix} \cdot \begin{bmatrix} 1-v \\ v \end{bmatrix}. \tag{6}$$

The auxiliary functions u, $(1-u)$, v and $(1-v)$ are called blending functions, because their effect is to blend together four separate boundary curves to give a single well-defined surface. The surface so identified has only positional continuity across patch boundaries. The substitution in (6) of the expression of the boundary curves:

$$r(u,0) = 1-u, \tag{7}$$

$$r(u,1) = 0, \tag{8}$$

$$r(0,v) = 1-v, \tag{9}$$

$$r(1,v) = 0, \tag{10}$$

$$r(0,0) = 1, \tag{11}$$

$$r(1,0) = r(1,1) = r(0,1) = 0 \tag{12}$$

allows us identifying the C^0 continuity interpolation function for quadrilateral patches (Fig. 4)

$$r(u,v) = (1-u)(1-v). \tag{13}$$

Looking for a C^1 continuity interpolation function, gradient continuity needs to be preserved and the patch needs to be defined not only in terms of its boundary curves but also in terms of its cross-boundary slopes. The equation for the patch can be derived in a similar manner to the one previously analysed except that now it is necessary to use generalised Hermite interpolation rather than generalised linear interpolation. The resulting equation is:

$$r(u,v) = [\,\varphi_0(u) \quad \varphi_1(u) \quad \psi_0(u) \quad \psi_1(u)\,] \cdot \begin{bmatrix} r(0,v) \\ r(1,v) \\ r_u(0,v) \\ r_u(1,v) \end{bmatrix} + [\,r(u,0) \quad r(u,1) \quad r_v(u,0) \quad r_v(u,1)\,] \cdot \begin{bmatrix} \varphi_0(v) \\ \varphi_1(v) \\ \psi_0(v) \\ \psi_1(v) \end{bmatrix}$$
$$- [\,\varphi_0(u) \quad \varphi_1(u) \quad \psi_0(u) \quad \psi_1(u)\,] \cdot \begin{bmatrix} r(0.0) & r(0,1) & r_v(0,0) & r_v(0,1) \\ r(1,0) & r(1,1) & r_v(1,0) & r_v(1,1) \\ r_u(0,0) & r_u(0,1) & r_{uv}(0,0) & r_{uv}(0,1) \\ r_u(1,0) & r_u(1,1) & r_{uv}(1,0) & r_{uv}(1,1) \end{bmatrix} \cdot \begin{bmatrix} \varphi_0(v) \\ \varphi_1(v) \\ \psi_0(v) \\ \psi_1(v) \end{bmatrix}. \tag{14}$$

The substitution of the following curve expressions in (14):

$$r(u,0) = 1-3u^2+2u^3, \tag{15}$$

$$r(u,1) = 0, \tag{16}$$

$$r(0,v) = 1-3v^2+2v^3, \tag{17}$$

$$r(1,v) = 0, \tag{18}$$

$$r(0,0) = 1, \tag{19}$$

$$r(1,0) = r(1,1) = r(0,1) = 0 \tag{20}$$

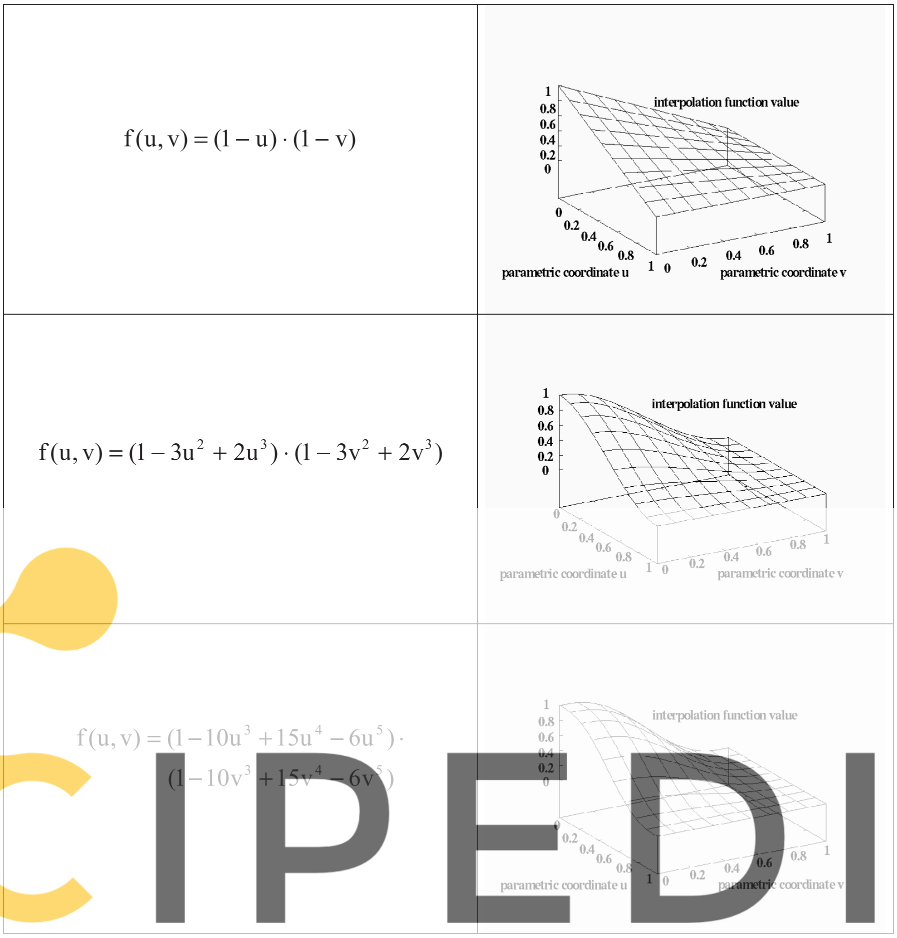


Fig. 4. C^0, C^1 and C^2 interpolation functions for the two dimensional quadrangular domain with corresponding analytical expressions.

and the imposition of null first and cross derivatives leads to the C^1 interpolation function expression (Fig. 4):

$$r(u,v) = (1 - 3u^2 + 2u^3)(1 - 3v^2 + 2v^3). \tag{21}$$

An identical process is followed for the identification of the C^2 continuity interpolation function. The equation of the patch with prescribed cross-boundary second derivatives can be derived in a similar way. Two more blending functions are needed and the square matrix involved will be of order 6×6. The function is shown in Fig. 4 and is given by the equation

$$r(u,v) = (1 - 10u^3 + 15u^4 - 6u^5)(1 - 10v^3 + 15v^4 - 6v^5). \tag{22}$$

The requirement of constant unit sum over the domain is satisfied in the three cases. The interpolation function definition requires verifying the correct sequence of the vertices, independently if clockwise or anticlockwise. This is to avoid the creation of curves connecting opposite vertices, and then, the creation of an interpolation function without meaning.

2.2.3. Two-dimensional interpolation functions: triangular surfaces

The definition of the interpolation functions for a triangular domain is more complex than the one previously explained. Let consider the standard triangle with vertices $V_1(1{,}0)$, $V_2(0{,}1)$ and $V_3(0{,}0)$ as in Fig. 5. Let us define as E_1 the side of the triangle opposite to the vertex V_1, E_2 the side opposite to the vertex

V_2 and E_3 the side opposite to the vertex V_3. Moreover, let define P_1, P_2 and P_3 operators that interpolate the curve representative of each side of the triangle in a direction parallel to the E_1, E_2 and E_3 sides, respectively (Fig. 6).

In the linear case the P_1 interpolant can be defined as [10]

$$P_1[r(u,v)] = ur(1-v,v) + vr(u,1-u) \tag{23}$$

and the P_2 interpolant as

$$P_2[r(u,v)] = r(0,v) + r(u,0) - r(0,0). \tag{24}$$

The boolean sum of these two interpolants:

$$P_1 \oplus P_2 = P_1 + P_2 - P_1 P_2 \tag{25}$$

gives the expression of a linear patch over the triangle. With the substitution of the expressions:

$$r(u,0) = 1-u, \tag{26}$$

$$r(0,v) = 1-v, \tag{27}$$

$$r(1,0) = r(0,1) = 0, \tag{28}$$

$$r(0,0) = 1, \tag{29}$$

describing the side curves and the actual value of the surface at the vertices, it is possible to obtain the C^0 continuity interpolant for the triangle as (Fig. 7)

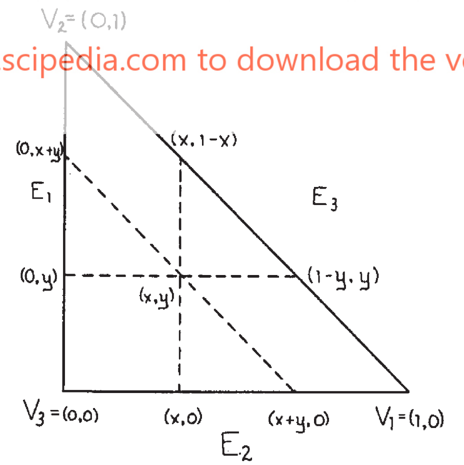


Fig. 5. Vertex and side definition for the triangular reference surface.

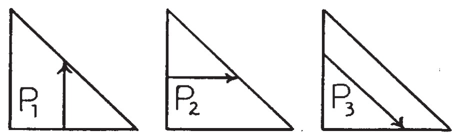


Fig. 6. Interpolant definition for the triangular reference surface.

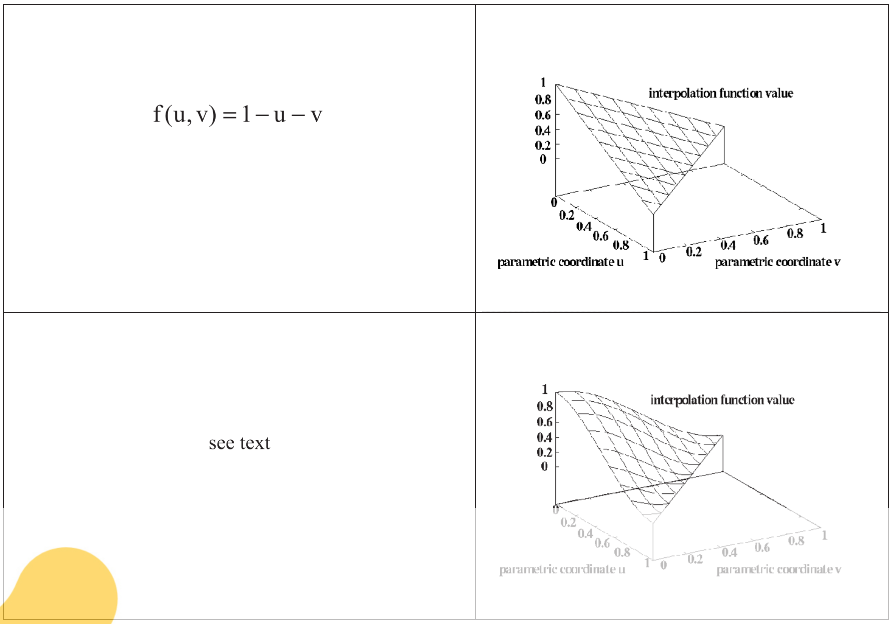


Fig. 7. C^0 and C^1 interpolation functions for the two dimensional triangular domain and corresponding analytical expressions.

$$r(u,v) = 1 - u - v. \tag{30}$$

The approach followed by Gregory [9,12] has been used to identify the interpolation function for the C^1 case. The three interpolants are here defined by

$$P_1[r(u,v)] = \sum_{i=0}^{1} \varphi_i\left(\frac{v}{1-u}\right) \cdot (1-u)^i \cdot r_{0,i}(u,0) + \sum_{i=0}^{1} \psi_i\left(\frac{v}{1-u}\right) \cdot (1-u)^i \cdot r_{0,i}(u,1-u), \tag{31}$$

$$P_2[r(u,v)] = \sum_{i=0}^{1} \varphi_i\left(\frac{u}{1-v}\right) \cdot (1-v)^i \cdot r_{i,0}(0,v) + \sum_{i=0}^{1} \psi_i\left(\frac{u}{1-v}\right) \cdot (1-v)^i \cdot r_{i,0}(1-v,v), \tag{32}$$

$$\begin{aligned} P_3[r(u,v)] = {} & \sum_{i=0}^{1} \varphi_i\left(\frac{u}{u+v}\right) \cdot (u+v)^i \cdot \left[\frac{\partial}{\partial u} - \frac{\partial}{\partial v}\right]^i r(0,u+v) \\ & + \sum_{i=0}^{1} \psi_i\left(\frac{u}{u+v}\right) \cdot (u+v)^i \cdot \left[\frac{\partial}{\partial u} - \frac{\partial}{\partial v}\right]^i r(u+v,0), \end{aligned} \tag{33}$$

where the φ_i and the ψ_i are the Hermite interpolation functions. The expression of the interpolating surface is given by

$$r(u,v) = \alpha(u,v) P_1[r(u,v)] + \beta(u,v) P_2[r(u,v)] + \gamma(u,v) P_3[r(u,v)], \tag{34}$$

where

$$\alpha(u,v) = u^2[3 - 2u + 6v(1-u-v)], \tag{35}$$

$$\beta(u,v) = v^2[3 - 2v + 6u(1-u-v)], \tag{36}$$

$$\alpha(u,v)+\beta(u,v)+\gamma(u,v)=1. \tag{37}$$

The substitution of (15)–(19) into (34) and the imposition of zero first and cross derivatives, lead to the expression of the C^1 continuity interpolation function over a triangle (Fig. 7)

$$\begin{aligned} r(u,v) = {} & \frac{\alpha(u,v)r(u,0)(u+v-1)^2(2v-u+1)}{(1-u)^3} + \frac{\beta(u,v)r(0,v)(u+v-1)^2(2u-v+1)}{(1-v)^3} \\ & + \gamma(u,v)\left[\frac{v^2(3u+v)r(0,v)}{(u+v)^3} + \frac{u^2(3v+u)r(u,0)}{(u+v)^3} - \frac{u^2v^2(-6u+6u^2-6v+6v^2)}{(u+v)^4}\right]. \end{aligned} \tag{38}$$

The sum of the C^0 and C^1 continuity interpolation functions evaluated at each of the three vertices of the triangle is constant and equal to one. Due to the symmetry of the equations the sequence of the vertices has no relevance on the numerical results.

The C^2 continuity interpolation function is still missing and this will be the object of future research work.

2.3. Displacement propagation inside the discrete model

The interpolation functions previously defined allow for the propagation of the displacements defined by the vectors over the macro-patches. The process takes into account only the boundaries of the discrete model of the analysed domain. To reduce the distortion effects that would take place in the areas near the boundaries of the model during the updating process operated by the optimisation code, it is necessary to propagate these boundary modifications to the internal nodes of the discrete model. These nodes will belong to the real structure if a structural problem is considered, or to the fluid domain if a fluid-dynamic problem is analysed.

The propagation of the boundary displacements can be accomplished by considering the mesh as a fictitious linear elastic structure. By solving a linear structural problem with the displacements of the moving surfaces as imposed displacements, it is possible to obtain the displacement of the nodes of the entire mesh. By using this procedure, the displacements of the boundaries of the structure are extrapolated to the entire mesh using the isoparametric shape functions of the finite elements. Unfortunately, the solution of the structural problem by considering an isotropic and homogeneous linear material does not provide a uniform displacement distribution as should be required to minimise the element distortion during the mesh updating process. In fact, the elements near the moving surfaces are constrained to modify their shape much more than the ones far from these surfaces.

Finite elements are used in this process just as an interpolation method and, consequently, there is not any interest about the stress distribution obtained depending on the specific material used for the analysis. Different mechanical properties can therefore be selected and assigned to each mesh element. Assigning materials with higher mechanical characteristics to the elements near the moving surfaces and materials with lower mechanical characteristics to the elements far from these surfaces, it is possible to distribute the mesh deformation more uniformly. Different laws of variation of the mechanical characteristics can be adopted following geometrical or more physical criteria. For example, following a pure geometrical criterion, the Young modulus of the mesh elements can be selected depending on the minimum distance of the element itself from the nearest moving surface. Differently, using a more physical approach, the Young modulus can be selected depending on data concerning the element strain vector or the element strain energy [8].

The best results have been obtained using a distribution law of the material depending on the strain energy [13]. A preliminary linear analysis with an isotropic and homogeneous material allows for the evaluation of the strain field for the entire discrete model. Once the strain components ε_1, ε_2 and ε_3 along the principal strain reference system have been evaluated, it is possible to evaluate the strain energy of each element as

$$E_d = \bar{E}\left[\left(\varepsilon_1^2+\varepsilon_2^2+\varepsilon_3^2\right) - 2\nu(\varepsilon_1\varepsilon_2+\varepsilon_2\varepsilon_3+\varepsilon_3\varepsilon_1)\right] \tag{39}$$

and to compare it with the strain energy evaluated considering a constant strain field $\bar{\varepsilon}$:

$$E_d = 3\bar{\varepsilon}^2 E(1-2v). \tag{40}$$

The comparison leads to the following expression for the new Young modulus to be assigned to each element for the final analysis:

$$E = \frac{\bar{E}}{\bar{\varepsilon}^2}\frac{\left(\varepsilon_1^2+\varepsilon_2^2+\varepsilon_3^2\right)-2v(\varepsilon_1\varepsilon_2+\varepsilon_2\varepsilon_3+\varepsilon_3\varepsilon_1)}{3(1-2v)}, \tag{41}$$

where v can be arbitrarily selected and $\bar{\varepsilon}$ has been chosen to be equal to 1.0E−06.

If a fluid-dynamic problem is analysed it is necessary to be aware that the surfaces that limit the fluid domain remain fixed in the solution of the fictitious structural problem.

3. Implementation of the method

The proposed method operates by the attribution of well-defined properties to several geometrical entities like points, lines or surfaces. So, the geometrical definition of the structure to be analysed is needed only. There is no problem indeed to use the same method for displacement propagation starting from a discrete model. In this case several facilities as the transfer of the geometrical properties to the discrete model are lost preserving the main characteristics of the method.

The proposed method operates in two steps. The first step is completely controlled by the user. He has to identify the most correct and useful design variables to be defined taking into account the specific kind of problem to be analysed. The variable identification is performed by the definition of several geometrical properties directly linked to the geometrical entities (points, lines, surfaces) of the geometrical model. It is necessary to identify: the surfaces influenced by every shape variables (curves in two-dimensional problems) gathered together in several macro-patches, the vertices of each macro-patch, the displacement vectors representative of the shape variable applied to the above mentioned vertices and the continuity degree to be preserved in the propagation of the vertices displacement.

In the second step, all the geometrical information previously defined are transferred to the discrete model and elaborated. A set of instructions is generated by linking every shape variable to the displacement of the nodes of the discrete model. This information is used by the optimisation code to update the discrete model at every iteration depending on the values of the design variables. This second phase is completely independent and does not require any intervention from the user.

The proposed method has been implemented in a specific computer program that takes advantage of the capabilities offered by the pre-post-processor system GiD developed at the International Centre of Numerical Methods in Engineering (CIMNE, Barcelona) [14]. The result of the application of the program is the creation of two files for post processing purposes and one file in standard NASTRAN format containing the relationships between the design variables and the nodal displacements of the discrete model.

4. Examples

Two examples will be shown next. The first example concerns the shape variable definition in an airfoil fluid-dynamic optimisation problem. The airfoil is defined by several points connected by spline curves. The fluid domain has been defined as a rectangular ‘box’ around the airfoil itself.

There exist different possibilities for the definition of the shape variables: for example, local shape variables changing only a portion of the geometry of the profile can be used or global ones taking in

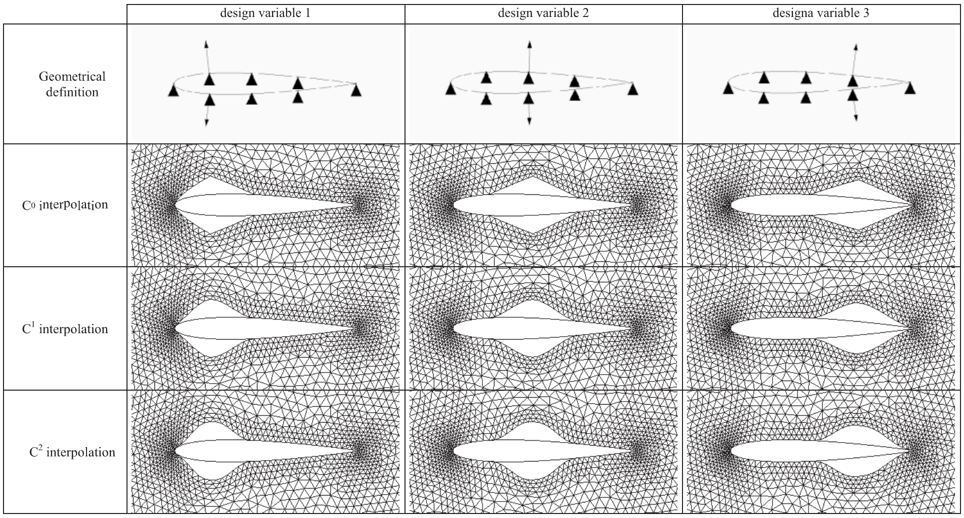


Fig. 8. Local shape variables for the airfoil problem.

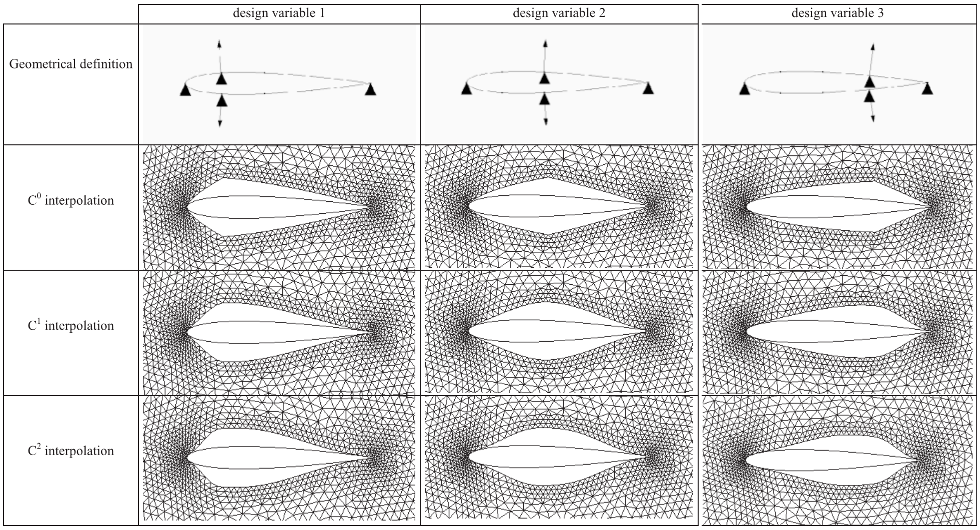


Fig. 9. Global shape variables for the airfoil problem.

consideration all the profile can be preferred. Three shape variables have been defined using the two different approaches. The local design variables obtained by the application of the three available interpolation functions are shown in Fig. 8. The global design variables obtained by the application of the three available interpolation functions are shown in Fig. 9; different macro-patches have been used here to define the three design variables. In the two figures, the triangles define the limits of the macro-patches and the vectors the design variable. The three interpolation facilities have been used to allow for a comparison between the different results that can be obtained. Mesh modification is quite uniform and elements near the moving surfaces maintain quite the same dimension and shape as the original ones. This procedure allows for reducing element distortion effect due to surface modification during the optimisation process.

A three dimensional example concerning the bulb of a ship hull is presented too. Symmetric shape modifications due to seven shape variables are shown in Figs. 10 and 11. The geometry of the bulb and

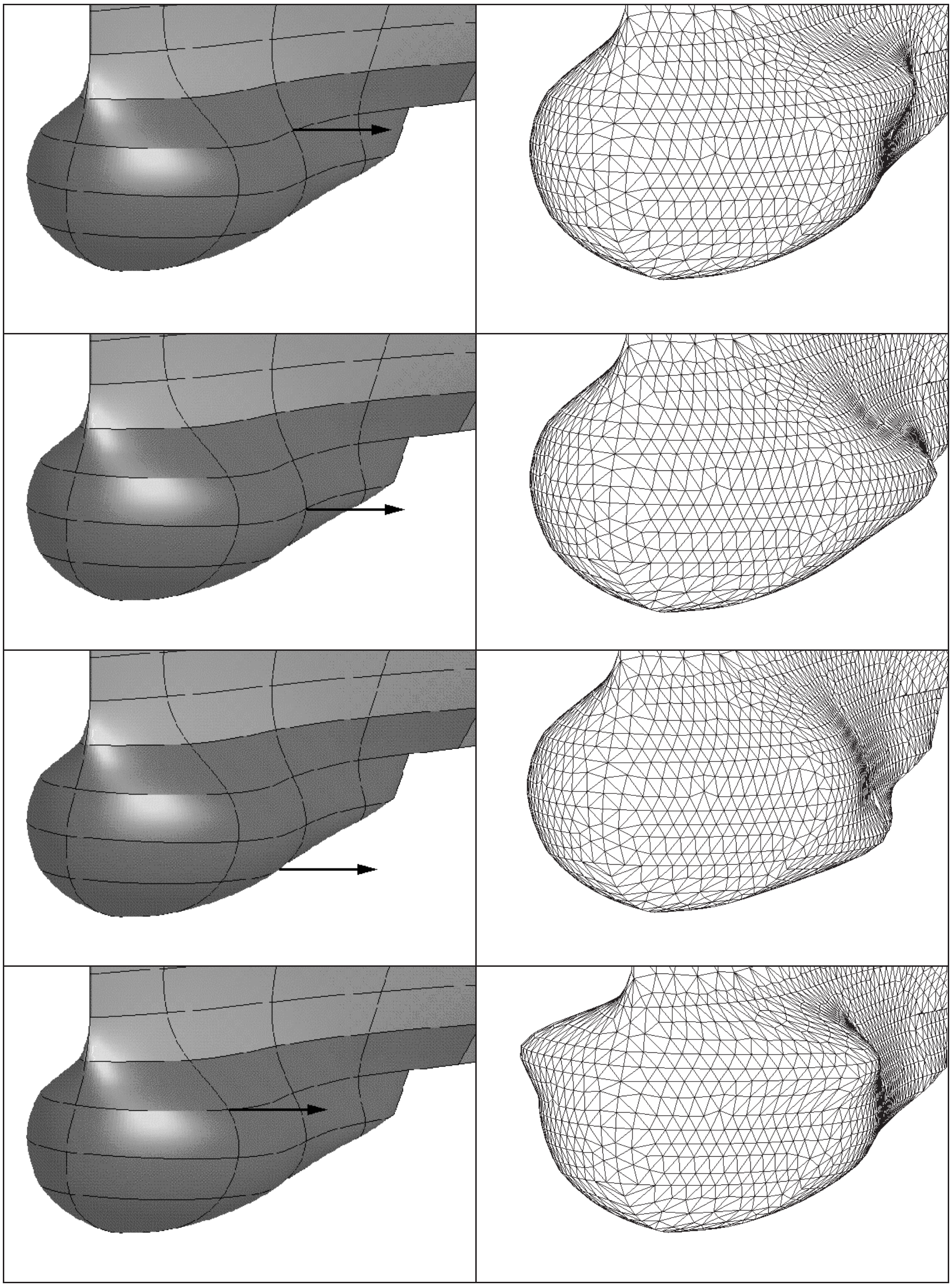


Fig. 10. Symmetric surface modification due to seven shape variables defined over the bulb of a ship hull.

its partition in macro-patches can be observed on the left of each figure together with the vector representative of the design variable. Tangent continuity is preserved over all the surface of the structure by means of the C^1 interpolation function used to extrapolate the vertices displacements. The mesh is not shown being impossible to identify the singular elements and to keep trace of their shape modifications.

Again, the quality of the modified shape and the corresponding mesh for each design variable are, practically, as good as the original one.

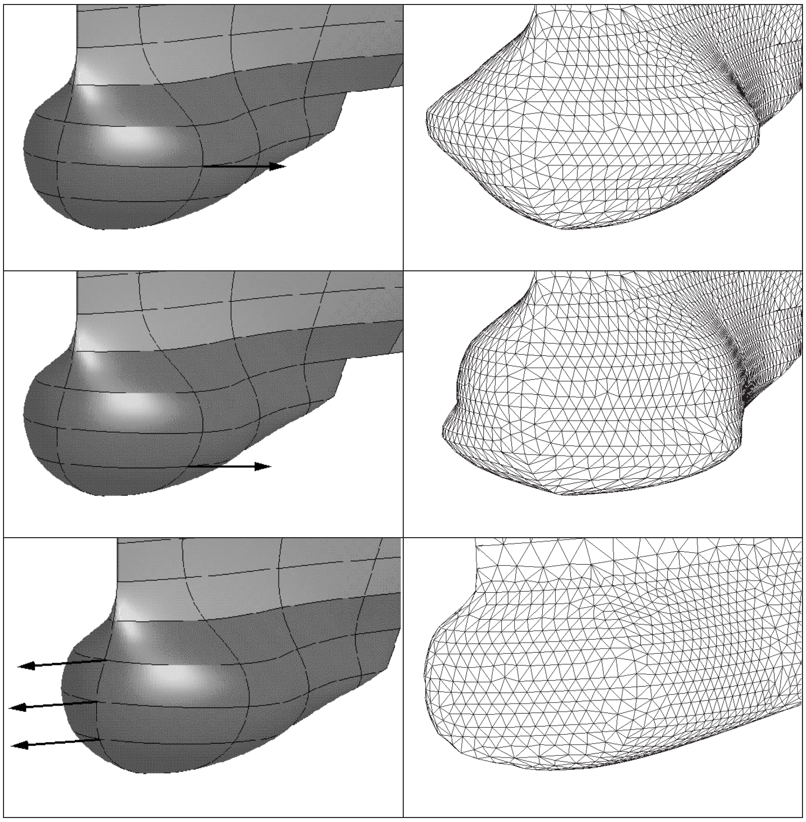


Fig. 11. Continuation of Fig. 10.

In a complete optimisation process, all the design variables are used all together, and the shape of each design is defined through the superposition of the shape modification corresponding to each design variable using expression (1).

5. Conclusions

A new method for shape design variable definition has been proposed. It overcomes some of the pitfalls of the methods formerly proposed by several authors. It is able to manage two-dimensional as well as three-dimensional problems allowing the user to work directly on the geometrical model of the problem and simplifying the variable definition phase of an optimisation problem layout. C^0, C^1 or C^2 continuity properties of the boundary surface of the analysed structure can be preserved leading to the definition of shape variables suitable for optimisation problems in structural mechanics as well as in fluid-dynamic or aerodynamic analyses. Boundary surface displacements are propagated over the internal nodes of the mesh by keeping the distortion of the elements to a minimum. This allows to reduce the error introduced during the optimisation process due to the presence of highly distorted elements.

Acknowledgements

This work has been supported by the EU Marie Curie Grant BRMA-CT97-5761 held by the first author. The authors want to acknowledge the support of E. N. BAZAN (Madrid) and of the International Centre of Numerical Methods in Engineering (CIMNE, Barcelona).

References

[1] M.H. Imam, Three dimensional shape optimization, Int. J. Numer. Meth. Engrg. 18 (1982) 661–677.
[2] G. Vanderplaats, Approximation concepts for numerical airfoil optimization, NASA Technical Report no. 1370, 1979.
[3] V. Braibant, C. Fleury, Shape optimal design using B-splines, Comput. Meth. Appl. Mech. Engrg. 44 (1984) 247–267.
[4] V. Braibant, C. Fleury, An approximation-concepts approach to shape optimal design, Comput. Meth. Appl. Mech. Engrg. 53 (1985) 119–148.
[5] S. Zhang, A.P. Belegundu, A systematic approach for generating velocity fields in shape optimization, Struct. Optim. 5 (1992) 84–94.
[6] G. Farin, Curves and Surfaces for Computer Aided Geometric Design, Academic Press, New York, 1993.
[7] R.E. Barnhill, R.F. Riesenfeld, Computer Aided Geometric Design, Academic Press, New York, 1974.
[8] G. Chiandussi, G. Bugeda, E. Oñate, A simple method for automatic update of finite element meshes, CIMNE Publication no. 147, January 1999.
[9] J.A. Gregory, Smooth interpolation without twist constraints, in: R.E. Barnhill, R.F. Riesenfeld (Eds.), Computer Aided Geometric Design, Academic Press, New York, 1974.
[10] R.E Barnhill, Smooth interpolation over triangles, in: R.E. Barnhill, R.F. Riesenfeld (Eds.), Computer Aided Geometric Design, Academic Press, New York, 1974.
[11] S.A. Coons, Surface patches and B-spline curves, in: R.E. Barnhill, R.F. Riesenfeld (Eds.), Computer Aided Geometric Design, Academic Press, New York, 1974.
[12] R.E Barnhill, J.A. Gregory, Polynomial interpolation to boundary data on triangles, Math. Comput. 29 (131) (1975) 726–735.
[13] G. Chiandussi, G. Bugeda, E. Oñate, Shape variable definition with C^0, C^1 and C^2 continuity functions, CIMNE Publication no. 134, July 1998.
[14] GID User Manual, CIMNE, Barcelona.