An Efficient Hybrid Optimization Strategy for Surface Reconstruction

Giulia Bertolino, Marco Montemurro, Nicolas Perry, Franck Pourroy

An Efficient Hybrid Optimisation Strategy for Surface Reconstruction.

Giulia Bertolino[a], Marco Montemurro[a,*], Nicolas Perry[a], Franck Pourroy[b]

[a]*Arts et Métiers Institute of Technology, Université de Bordeaux, CNRS, INRA, Bordeaux INP, HESAM Université, I2M, I2M UMR 5295 F-33405 Talence, France*
[b]*Univ. Grenoble Alpes, CNRS, Grenoble INP*, G-SCOP, 38000 Grenoble, France*

Abstract

An efficient and general surface reconstruction strategy is presented in this study. The proposed approach can deal with both open and closed surfaces of genus greater than or equal to zero and it is able to approximate non-convex sets of target points (TPs). The surface reconstruction strategy is split into two main phases: (a) the mapping phase, which makes use of the shape preserving method (SPM) to get a proper parametrisation of each sub-domain composing the TPs set; (b) the fitting phase, where each patch is fitted by means of a suitable Non-Uniform Rational Basis Spline (NURBS) surface without introducing simplifying hypotheses and/or rules on the parameters tuning the shape of the parametric entity. Indeed, the proposed approach aims stating the surface fitting problem in the most general sense, by integrating the full set of design variables (both integer and continuous) defining the shape of the NURBS surface. To this purpose, a new formulation of the surface fitting problem is proposed: it is stated in the form of a special Constrained Non-Linear Programming Problem (CNLPP) defined over a domain having variable dimension, wherein both the number and the value of the design variables are simultaneously optimised. To deal with this class of CNLPPs, a hybrid optimisation tool has been employed. The optimisation procedure is split in two steps: firstly, an improved genetic algorithm (GA) optimises both the value and the number of design variables by means of a two-level Darwinian strategy allowing the simultaneous evolution of individuals and species; secondly, the solution provided by the GA constitutes the initial guess for the subsequent deterministic optimisation, which aims at improving the accuracy of the fitting surfaces. The effectiveness of the proposed methodology is proven through some meaningful benchmarks taken from the literature.

Keywords: NURBS Surfaces, Surface Fitting, Shape Preserving Method, Mapping, Genetic Algorithms, Optimisation, Reverse Engineering

1. Introduction

Surface reconstruction is widely exploited in different fields for various aims: reverse engineering, geometrical modelling or medical images reconstruction. In the context of surface reconstruction, a geometrical *Computer-Aided Design* (CAD) compatible entity, like a *Non-Uniform Rational Basis*

[*] Corresponding author. Tel.: +33 55 68 45 422, Fax.: +33 54 00 06 964.
Email address: marco.montemurro@ensam.eu; marco.montemurro@u-bordeaux.fr ()

Spline (NURBS) surface, is used to approximate (or to interpolate, depending on the problem at hand) a non-degenerate tessellation representing the boundary of a part. Of course, the Cartesian coordinates of *target points* (TPs), describing the *target surface* (TS) to be reconstructed, must be available: they can be obtained via a 3D scan or from other sampling procedures and they can be easily organised in a *Standard Tessellation Language* (STL) file. Further information are also available in an STL file, e.g. the connectivity map of vertexes and the triangles normals, which can be exploited for decoding the surface from the discrete domain to the continuous one.

Generally speaking, the surface reconstruction strategy is articulated into two different phases: *mapping* and *fitting*. The first step, i.e. the mapping, aims at retrieving a suitable map of the 3D tessellation over a simplified 2D domain (generally a unit square or a unit circle) [1–5]. The mapping of the TS should preserve some fundamental information of the 3D shape of the tessellation to ensure a smooth surface approximation. Once a proper mapping of the TS is obtained, the fitting phase can take place: it aims at determining the optimum value of the variables tuning the shape of the parametric surface (defined over the parametric domain provided by the mapping) approximating the TS. Usually, the fitting phase is stated as a constrained non-linear programming problem (CNLPP) where the distance between the parametric surface and the TS is minimised by satisfying, simultaneously, some constraints, e.g. on the maximum allowable curvature along each parametric direction, continuity conditions, etc.

Different mapping methods can be found in the literature: each one is characterised by advantages and drawbacks, that make it suitable for solving problems of mesh processing, including detailed mapping [6], morphing [7], mesh editing [8], mesh completion [9], mesh compression [10], surface-fitting [11] and shape-analysis [12]. For example, the projection method proposed by Piegl *et al.* [13] can be easily applied to unfolded surfaces, where the projection of points belonging to the surface will lead to a unique solution. Starting from the set of TPs, $\mathrm{OQ}_k = (x_k, y_k, z_k) \in \mathbb{R}^3$, $k = 1, ..., N$, an iterative strategy is used to find the set of (u_{1k}, u_{2k}) parameters involved in the definition of the NURBS surface used in the subsequent fitting phase. The problem is formulated as an unconstrained minimisation of the distance between the TPs and their counterparts evaluated on the parametric surface. The set of (u_{1k}, u_{2k}) parameters is the result of the minimum distance problem. Nevertheless, this method presents a great limitation: it can be applied only to simple surfaces, or at least to surfaces decomposable into unfolded patches. In this second case the mesh should be divided in different patches, but if the topology is particularly complex, the number of patches could be too large and the computational efficiency poor.

A mapping method, overcoming the problem of dealing only with open surfaces, was proposed by Sahand Jamal and Kim [14]. It consists of mapping a genus zero closed TS on the external surface of a sphere, in order to assign a pair of parameters (e.g. latitude and longitude) to each vertex of the tessellation. However, a fundamental condition must be met to assure a proper result. In particular, the Euler equation must be satisfied: $F + V - (E + 2) + 2H = 0$, where F, V, E and H are the numbers of facets, vertices, edges and handles, to filter out toroidal and handled surface.

Gu and Yau [5] developed a mapping method which allows finding the parametrisation of surfaces with non-trivial topologies (i.e. closed surfaces of genus greater than zero). This mapping technique is able to preserve the conformality everywhere, without boundary discontinuity. The method relies on a general homology basis to represent the topology of the surface, which is constituted of a set of curves that can be deformed to any closed curve on the surface. The curves belonging to the homology basis are then used to cut the TS in order to get a topological disk, which is called the fundamental domain.

In [15], Floater developed a general mapping method that has been used (and extended) in other studies [3, 4, 16, 17]. The mapping method proposed by Floater, also called *shape preserving method* (SPM), is a fast procedure, based on a linear system of equations, capable of preserving the shape of the original TS. This leads to optimal results in the surface fitting phase, as shown in [1]. However the SPM, can be applied only to open surfaces of genus zero, but this limitation can be overcome by performing an efficient segmentation of a general surface (open or closed) of genus greater than zero, into patches (satisfying the hypotheses of the SPM). In the context of the SPM, the parameters associated to each internal vertex of the tessellation are obtained as a linear convex combination of those related to its neighbours. In this way the values of (u_{1k}, u_{2k}) related to each TP are averaged on the neighbourhood, assuring the shape preservation in terms of angles and distances, when mapping the TS from $\mathbb{R}^3$ to $\mathbb{R}^2$.

Once a proper mapping of the TS is provided, the surface fitting phase can take place, in the form either of an interpolation or of an approximation problem. This topic has been extensively studied in the literature. The vast majority of works makes use of B-Spline or NURBS entities as fitting surfaces. The variables are the parameters defining the NURBS surface, i.e. control points (CPs) coordinates, weights, blending functions degrees and knot-vectors (KVs) components [13]. For example, in [13], both *interpolation* and *approximation* techniques are proposed, highlighting the limits and the advantages of both strategies. When interpolation is considered, the parametric surface passes exactly through the set of TPs. Of course, in this case the error (in terms of distance between the parametric surface and the TS) is strongly reduced for each TP, whilst a good approximation of the surface shape between TPs is not ensured (this issue is usually defined as *overfitting*). Conversely, the approximation approach aims at capturing the overall shape of the TS, minimising the maximal deviation between the TPs and those belonging to the parametric surface. Among the different problem formulations, the least-squares one, constrained or unconstrained, is the most common formulation. In [13], an efficient algorithm for determining the optimal values of the CPs coordinates of a B-Spline surface is proposed. However, this approach is characterised by a major limitation: the TPs must be arranged in an ordered grid.

Conversely, when TPs are the result of a scan acquisition, they cannot be easily arranged in an ordered grid (i.e. scattered data points), hence a more general fitting strategy is needed. In [18, 19] two approaches are proposed to deal with this problem. In [18], the surface fitting problem is formulated as an unconstrained minimisation problem. The functional to be minimised is defined as a suitable balance between the deformation energy S of the surface and distance between the fitting surface and the TS. The two terms involved in this functional have different effects, competing to achieve a surface which fits the TPs and which is as smooth as possible. The proposed algorithm finds iteratively the best approximating surface, but the user has to define some parameters of the B-Spline entity like the number of KVs components and the degrees of the basis functions along each parametric direction. Conversely, in the work of Mao *et al.* [19], after a curve fitting of each rows of TPs, a re-sampling of the curves is proposed, with the aim of producing a structured grid of TPs, in order to use the approach proposed in [13].

A different approach is proposed in [20], which exploits the SPM during the mapping phase, and a B-Spline surface for the fitting phase. After setting the degrees and the KVs components (by means of empiric rules), a suitable functional, taking into account for the distance between the B-Spline surface and the TS and for the so-called *thin-plate spline energy* (TPSE) of the surface, is minimised.

As it can be inferred from this non-exhaustive state of the art, the approaches available in the literature are based on preliminary hypotheses and rules to set some parameters involved in the definition of the NURBS entity. Therefore, a general fitting strategy able to optimise at the same time both the number and the values of the NURBS constitutive parameters, i.e. degrees, number of CPs, CPs coordinates, weights and values of non-trivial KV components, is still missing.

In order to overcome this limitation, a general and original surface reconstruction method (SRM) is presented in this work. The proposed SRM is articulated in two phases, i.e. mapping and fitting. The first phase is based on the SPM [15] (extended to the case of closed surface of genus greater than zero) to retrieve a proper mapping of the TS. The second phase focuses on the surface fitting problem, which is formulated as an *unconventional* CNLPP. In particular, the proposed formulation aims at integrating all the parameters defining the shape of the NURBS surface in the design variables vector, in order to state the problem in the most general way. Nevertheless, this fact implies two consequences of paramount importance, constituting just as many difficulties in solving the related CNLPP. Firstly, when the number of CPs and the degrees of the basis functions are included among the unknowns, the overall number of design variables for the problem at hand is not set a-priori, hence, the resulting CNLPP is defined over a search space of *variable dimension*. Secondly, the related resolution strategy must be able to handle design variables of different nature and to optimise, at the same time, the dimension of the design domain and the value of each constitutive parameter of the NURBS surface.

In order to overcome the two aforementioned issues, the surface fitting phase is split in two optimisation steps. Firstly, the ERASMUS (EvolutionaRy Algorithm for optimiSation of ModUlar Systems) code optimises both the value and the number of design variables by means of a two-level Darwinian strategy, allowing the simultaneous evolution of individuals and species [21]. Secondly, the optimum solution provided by ERASMUS constitutes the initial guess for the local deterministic optimisation, which aims at improving the accuracy of the fitting surface.

Starting from benchmarks dealing with genus zero open surfaces, a general strategy (involving segmentation, sub-domain mapping and C1 connectivity conditions) is proposed for genus G surfaces (open and closed). The effectiveness of the proposed SRM is tested on some meaningful real-world engineering problems taken from the literature [20, 22].

The paper is structured as follows. Section 2 presents the theoretical background of both the NURBS surfaces and the SPM. In Section 3, the surface fitting problem formulation and the related numerical strategy are introduced for the case of genus zero open surfaces. In Section 4, the methodology is extended to the general case of genus G surfaces (open and closed). The results of the proposed method are discussed in Section 5, while Section 6 ends the paper with some concluding remarks and prospects.

Notation. Upper-case bold letters are used to indicate matrices, while lower-case bold letters indicate vectors, which are to be intended as column ones. Conversely, a position vector linking point A with point B will be indicated as AB (without using bold letters).

2. Theoretical Background

2.1. NURBS Surfaces

In this section a brief reminder on B-Spline and NURBS surfaces is given. B-Spline and NURBS entities constitute a generalisation of the well-known Bézier's curves and surfaces [13]. Nowadays, NURBS entities (curves, surfaces and hyper-surfaces) are used in different fields: topology optimisation [22–26], shape optimisation [27, 28], anisotropy field optimisation for variable stiffness composites [29, 30] and surrogate models generation [31–33].

In this work a NURBS surface is used to approximate the set of TPs embedded in the surface tessellation. The parametric explicit form of a NURBS surface is:

$$\mathbf{s}(u_1, u_2) \coloneqq \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} R_{i_1,i_2}(u_1, u_2)\mathbf{X}_{i_1,i_2}, \tag{1}$$

where $\mathbf{s}^{\mathrm{T}}(u_1, u_2) \coloneqq \{x(u_1, u_2), y(u_1, u_2), z(u_1, u_2)\}$ are the Cartesian coordinates of a point belonging to the surface, whilst $R_{i_1,i_2}(u_1, u_2)$ is the generic rational basis function having the form

$$R_{i_1,i_2}(u_1, u_2) = \frac{N_{i_1,p_1}(u_1)N_{i_2,p_2}(u_2)w_{i_1,i_2}}{\sum_{k=0}^{n_1}\sum_{l=0}^{n_2} N_{k,p_1}(u_1)N_{l,p_2}(u_2)w_{k,l}}. \tag{2}$$

In Eqs. (1) and (2), u_1, u_2 are the dimensionless parameters (also called parametric coordinates) defined in the range $[0, 1]$, $N_{i_1,p_1}(u_1)$ and $N_{i_2,p_2}(u_2)$ are the basis functions, recursively defined by means of Bernstein polynomials [13], p_1, p_2 are the degrees, w_{i_1,i_2} are the weights and $\mathbf{X}^{\mathrm{T}}_{i_1,i_2} = (x_{i_1,i_2}, y_{i_1,i_2}, z_{i_1,i_2})$ is the array containing the Cartesian coordinates of the CPs. The $n_{\mathrm{CP}} = (n_1 + 1) \times (n_2 + 1)$ CPs form the so-called *control net.* The blending functions $N_{i_j,p_j}(u_j)$, with $j = 1, 2$, are recursively computed as

$$N_{i_k,0}(u_k) \coloneqq \begin{cases} 1, \text{ if } v^{(k)}_{i_k} \le u_k < v^{(k)}_{i_k+1}, \\ 0, \text{ otherwise}, \end{cases} \tag{3}$$

$$N_{i_k,q}(u_k) \coloneqq \tfrac{u_k - v^{(k)}_{i_k}}{v^{(k)}_{i_k+q} - v^{(k)}_{i_k}} N_{i_k,q-1}(u_k) + \tfrac{v^{(k)}_{i_k+q+1} - u_k}{v^{(k)}_{i_k+q+1} - v^{(k)}_{i_k+1}} N_{i_k+1,q-1}(u_k), q = 1, ..., p_k, \tag{4}$$

where $v^{(j)}_{i_j}$ is the i_j-th component of the following non-periodic, non-uniform *knot vector*:

$$\mathbf{v}^{(j)\mathrm{T}} = \{\underbrace{0, \dots, 0}_{p_j+1}, v^{(j)}_{p_j+1}, \dots, v^{(j)}_{m_j-p_j-1}, \underbrace{1, \dots, 1}_{p_j+1}\}. \tag{5}$$

It is noteworthy that the size of the knot vector is $m_j + 1$, with

$$m_j = n_j + p_j + 1. \tag{6}$$

The knot vector is a non-decreasing sequence of real numbers that can be interpreted as a discrete collection of values of the related dimensionless parameter u_j. Therefore, the knot vectors components split the surface in patches. The components of $\mathbf{v}^{(j)}$ are called *knots* and each knot can have a multiplicity λ.

Among the properties characterising blending functions, one of the most important is the *partition of unit property*, i.e.

$$\sum_{i_j=0}^{n_j} N_{i_j,p_j}(u_j) = 1, \ \forall u_j \in [0, 1]\,. \tag{7}$$

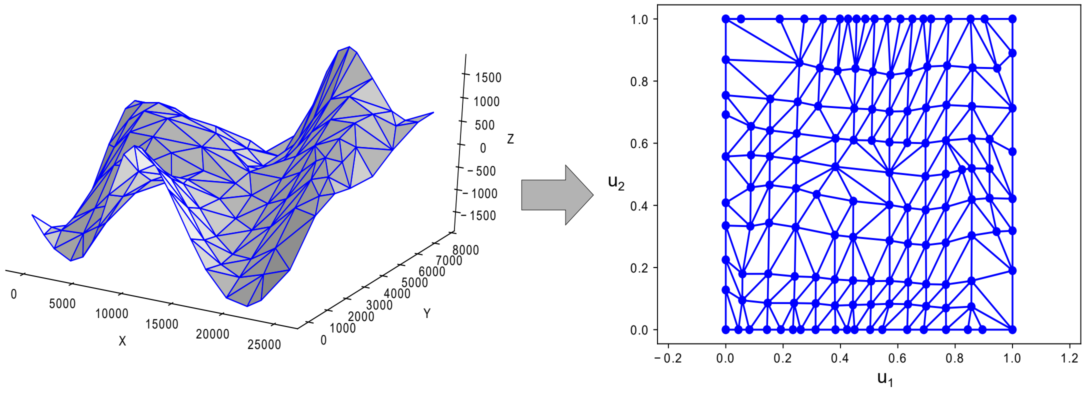


Figure 1: Mapping of the triangulated surface on the parametric domain.

This property allows for defining the B-Spline surface starting from the general definition of a NURBS surface of Eqs. (1)-(2). Indeed, for a B-Spline entity, the weights w_{i_1,i_2} take the same value. Considering the partition of unit property, for a B-Spline surface Eq. (2) simplifies to

$$R_{i_1,i_2}(u_1, u_2) = N_{i_1,p_1}(u_1) N_{i_2,p_2}(u_2). \tag{8}$$

For more details on NURBS surfaces the interested reader is addressed to [13].

2.2. The Shape Preserving Method

A mapping strategy is based on the assumption that, given two surfaces with a similar topology, a bijective mapping between them can always be defined [1]. Therefore, if the surface to be mapped is represented by means of a suitable triangulation, the mapping operation is named *parametrisation* and the result of this operation is the *parametrisation domain*. The parametrisation method considered in this study is the SPM [15]. In particular, the SPM is used to find the parameters (u_1, u_2), appearing in the definition of the NURBS surface of Eq. (1), and related to the TPs of the triangulated surface $S(x_i, y_i, z_i)$. The mapping operation generates an isomorphism between the triangulation P and the related graph G, as shown in Fig. 1. Consider the set of points (or nodes) belonging to the TS defined as $Q := \{\mathrm{OQ}_i^{\mathrm{T}} = (x_i, y_i, z_i)\,,\ 1 \leq i \leq N\}$, where N is the total number of TPs. The SPM can be applied if and only if the surface tessellation $S\,(G, Q)$ satisfies the following requirements:

- non degenerate triangular facets F, with vertices V and edges E must compose the connectivity graph $G = G\,(V, E, F)$;
- the surface should be open and of genus zero, i.e. without holes.

The procedure behind the SPM is divided in three macro-phases, see [15] for more details.

1. The TPs set, included in the STL file, is split into two different subsets: the set of *internal nodes*, i.e. $Q_{\mathrm{I}} := \{\mathrm{OQ}_1, \cdots, \mathrm{OQ}_n\}$, and the set of *boundary nodes*, i.e. $Q_{\mathrm{B}} := \{\mathrm{OQ}_{n+1}, \cdots, \mathrm{OQ}_N\}$, for some $n \in [1, N[$, the latter ordered in anticlockwise sequence.

2. The boundary nodes parametrisation is performed through the *chord length* method [13], in order to set the known terms of the linear convex combination system. In particular the boundary nodes are mapped on the boundary of a convex polygon $D \in [0,1] \,\mathrm{x}\, [0,1]$, i.e. a unit square, as follows:

$$\xi^B_{j+1} \coloneqq \xi^B_j + \frac{\|\mathrm{OQ}^B_{j+1} - \mathrm{OQ}^B_j\|}{L_{\mathrm{tot}}}, \quad j = 1, \dots, d, \tag{9}$$

where $L_{\mathrm{tot}} = \sum_{j=1}^{d} \|\mathrm{OQ}^B_{j+1} - \mathrm{OQ}^B_j\|$ and $\|\cdot\|$ is the Euclidean norm in the 3D space, while ξ^B_j indicates the value of the generic dimensionless parameter (i.e. either u_1 or u_2) located on the boundary of D. In Eq. (9), d represents the number of boundary nodes in Q_{B} located on the generic edge of the unit square D.

3. Then, the generic dimensionless parameter related to each internal node ξ^{I}_i is expressed as a linear convex combination of its N_i neighbours, i.e. a set of ξ vertices that are located in the vicinity of ξ^{I}_i.

$$\xi^I_i \coloneqq \sum_{j \in N_i} \lambda_{i,j} \xi_j, \quad i = 1, \dots, n, \text{ with} : \sum_{j \in N_i} \lambda_{i,j} = 1. \tag{10}$$

Since the dimensionless parameters related to the boundary nodes are known, those associated to the internal nodes can be obtained by solving the following linear system:

$$\begin{cases} \mathbf{\Lambda} \mathbf{u}^I_1 = \mathbf{b}_1, \\ \mathbf{\Lambda} \mathbf{u}^I_2 = \mathbf{b}_2, \end{cases} \tag{11}$$

with

$$\mathbf{b}_1 \coloneqq \{\lambda_{i,j} u^B_{1_j}\}, \tag{12}$$

$$\mathbf{b}_2 \coloneqq \{\lambda_{i,j} u^B_{2_j}\}, \tag{13}$$

where nodes identified by $j = n+1, \dots, N$, represent the boundary nodes contributions. The assessment of the weights matrix $\mathbf{\Lambda}$ is performed according to a two-steps strategy, which constitutes the kernel of the SPM.

(a) For each internal point $\mathrm{OQ}^{\mathrm{I}}_i$, the local (temporary) parametrisation of the one-ring neighbourhood (OQ_j) is computed, through a geodesic-based mapping, preserving (locally) the distance and the angles, i.e.

$$\|\mathbf{u}_j - \mathbf{u}_i\| = \|\mathrm{OQ}_j - \mathrm{OQ}^I_i\|, \ j = 1, \dots, N_i, \tag{14}$$

and for each triangle $[\mathbf{u}_k, \mathbf{u}_i, \mathbf{u}_j]$ in the neighbourhood of $\mathrm{OQ}^{\mathrm{I}}_i$

$$\mathrm{ang}(\mathbf{u}_k, \mathbf{u}_i, \mathbf{u}_j) = \rho \, \mathrm{ang}(\mathrm{OQ}_k, \mathrm{OQ}_i, \mathrm{OQ}_j), \text{ with} : \rho \coloneqq \frac{2\pi}{\theta_i}, \ \theta_i \coloneqq \sum_{k \in N_i} \mathrm{ang}(\mathrm{OQ}_k, \mathrm{OQ}_i, \mathrm{OQ}_j). \tag{15}$$

(b) The second step is to express $\mathbf{u}_i$ as a convex combination of the neighbouring mapped points $\{\mathbf{u}_j : j \in N_i\}$. Starting from the local temporary flat map, each internal edge $\overline{\mathbf{u}_i, \mathbf{u}_k}$ is prolonged to reach the intersection with the outer convex boundary. This operation allows identifying the triangle $(\mathbf{u}_k, \mathbf{u}_r, \mathbf{u}_{r+1})$, where the barycentric coordinates are evaluated as:

$$\mu_j^k = \frac{\text{area}(\mathbf{u}, \mathbf{u}_r, \mathbf{u}_{r+1})}{\text{area}(\mathbf{u}_k, \mathbf{u}_r, \mathbf{u}_{r+1})}, \qquad \mu_j^r = \frac{\text{area}(\mathbf{u}_k, \mathbf{u}, \mathbf{u}_{r+1})}{\text{area}(\mathbf{u}_k, \mathbf{u}_r, \mathbf{u}_{r+1})}, \qquad \mu_j^{r+1} = \frac{\text{area}(\mathbf{u}_k, \mathbf{u}_r, \mathbf{u})}{\text{area}(\mathbf{u}_k, \mathbf{u}_r, \mathbf{u}_{r+1})}. \quad (16)$$

Finally, the shape-preserving weights, appearing in matrix $\mathbf{\Lambda}$ of Eq. (17), are the average of the local barycentric coordinates μ_j^k.

$$\lambda_{ij} = \frac{1}{N_i} \sum_{k \in N_i} \mu_j^k. \quad (17)$$

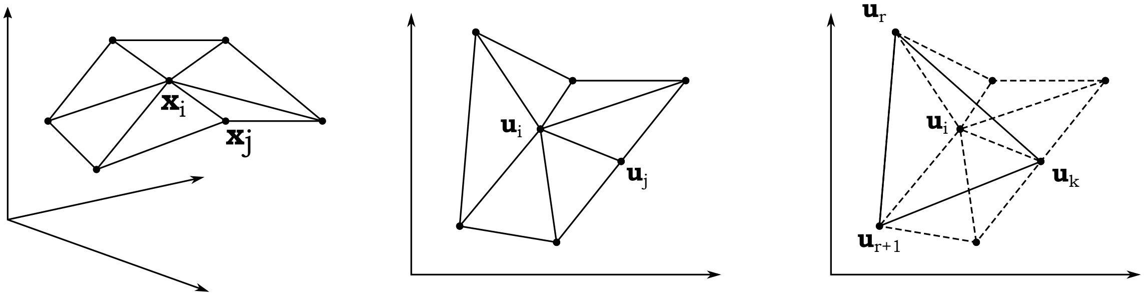


Figure 2: Calculation of the weights of the SPM.

3. Surface Fitting of Genus zero Open Surfaces

3.1. Design Variables

The goal of the surface fitting problem, presented in this study, is to find the optimum value of *all parameters* influencing the shape of a NURBS surface used to approximate the set of TPs. In particular, the approach discussed in this Section is very general and does not introduce neither simplifying hypotheses nor empirical rules to set the value of the parameters of the NURBS entity.

The variables affecting the shape of the NURBS surface are of different nature.

- *Integer variables*: the number of CPs along each parametric direction, i.e. $n_1 + 1,\ n_2 + 1$; the number of non-trivial knots along each parametric direction, i.e. $r_1 + 1,\ r_2 + 1$; the degrees of the blending functions $p_1,\ p_2$.
- *Continuous variables*: the non-decreasing sequence of knot vectors components $v_i^{(j)},\ i \in [p_j + 1, m_j - p_j - 1]$, the control points coordinates $\mathbf{X}_{i_1,i_2}$, the weights w_{i_1,i_2} and the set of the dimensionless parameters of the surface (u_{1_k}, u_{2_k}), for $k \in [0, N]$, corresponding to the set of TPs mapped on the NURBS surface.

In this work $p_1,\ p_2$ and $r_1,\ r_2$ are included in the vector of integer design variables $\boldsymbol{\zeta_1}$ as follows:

$$\boldsymbol{\zeta_1}^{\mathrm{T}} = \{p_1, p_2, r_1, r_2\}, \tag{18}$$

whilst, due to Eq. (6), n_j $(j = 1, 2)$ is calculated as:

$$n_j = r_j + p_j. \tag{19}$$

Consider, now, the set of continuous parameters. The dimensionless parameters of the surface are provided by the SPM, so they are no longer design variables. Moreover, the optimal CPs coordinates can be found through the analytical approach proposed in [20], by minimising the following cost function f

$$\min_{\mathbf{X}_{ij}} f\left(\mathbf{X}_{ij}\right), \text{ with } f\left(\mathbf{X}_{ij}\right) \coloneqq \sum_{\alpha=x,y,z} \sum_{k=1}^{N} \left[\alpha(u_{1k}, u_{2k}) - \alpha_k\right]^2 + \lambda J_\alpha, \tag{20}$$

where $\lambda \geq 0$ is a constant, whilst J_α is a smoothing term, i.e. the so-called TPSE [20], defined as

$$J_\alpha \coloneqq \int_0^1 \int_0^1 \left(\frac{\partial^2 \alpha}{\partial u_1^2}\right)^2 + 2\left(\frac{\partial^2 \alpha}{\partial u_1 \partial u_2}\right)^2 + \left(\frac{\partial^2 \alpha}{\partial u_2^2}\right)^2 \mathrm{d}u_1 \mathrm{d}u_2,\ \alpha = x, y, z. \tag{21}$$

By introducing the following linear index

$$\tau \coloneqq 1 + i_1 + i_2(n_1 + 1),\ \forall i_1 = 0, \cdots, n_1,\ i_2 = 0, \cdots, n_2\ \Rightarrow\ \tau = 1, \cdots, n_{\mathrm{CP}}, \tag{22}$$

the CPs coordinates can be grouped into the following vectors

$$\boldsymbol{\alpha}_{\mathrm{CP}}^{\mathrm{T}} \coloneqq \{\alpha_{0,0}, \cdots, \alpha_{n_1,n_2}\},\ \alpha = x, y, z. \tag{23}$$

Accordingly, the term J_α of Eq. (21) can be conveniently expressed in matrix form as:

$$J_\alpha = \boldsymbol{\alpha}_{\mathrm{CP}}^{\mathrm{T}} \mathbf{E} \boldsymbol{\alpha}_{\mathrm{CP}},\ \text{with } \mathbf{E} \coloneqq \mathbf{A} + 2\mathbf{B} + \mathbf{C}, \tag{24}$$

where matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C} \in \mathbb{R}^{n_{\mathrm{CP}} \times n_{\mathrm{CP}}}$ can be inferred from the corresponding 4D arrays, i.e.

$$\begin{aligned}
\mathrm{A}_{i,j,r,s} &\coloneqq \textstyle\int_0^1 \int_0^1 \dfrac{\partial^2 R_{i,j}(u_1, u_2)}{\partial u_1^2} \dfrac{\partial^2 R_{r,s}(u_1, u_2)}{\partial u_1^2} du_1 du_2,\\
\mathrm{B}_{i,j,r,s} &\coloneqq \textstyle\int_0^1 \int_0^1 \dfrac{\partial^2 R_{i,j}(u_1, u_2)}{\partial u_1 \partial u_2} \dfrac{\partial^2 R_{r,s}(u_1, u_2)}{\partial u_1 \partial u_2} du_1 du_2,\\
\mathrm{C}_{i,j,r,s} &\coloneqq \textstyle\int_0^1 \int_0^1 \dfrac{\partial^2 R_{i,j}(u_1, u_2)}{\partial u_2^2} \dfrac{\partial^2 R_{r,s}(u_1, u_2)}{\partial u_2^2} du_1 du_2.
\end{aligned} \tag{25}$$

The relationship between the 4D arrays of the above equation and the corresponding matrices can be immediately obtained by considering the linear index of Eq. (22):

$$\begin{gathered}
\mathrm{M}_{\gamma\tau} = \mathrm{M}_{i,j,r,s},\ \mathrm{M} = \mathrm{A}, \mathrm{B}, \mathrm{C},\\
\gamma, \tau = 1, \cdots, n_{\mathrm{CP}};\ i, r = 0, \cdots, n_1;\ j, s = 0, \cdots, n_2.
\end{gathered} \tag{26}$$

Of course, the local minimum of function f of Eq. (20) occurs when its gradient with respect to CPs coordinates $\boldsymbol{\alpha}_{\text{CP}}$ is null, i.e. $\dfrac{\partial f}{\partial \alpha_\tau} = 0,\ \forall \tau = 1, \cdots, n_{\text{CP}},\ \alpha = x, y, z$. By imposing this condition, one obtains the following linear system:

$$(\mathbf{G} + \lambda \mathbf{E})\, \boldsymbol{\alpha}_{\text{CP}} = \mathbf{D}^T \mathbf{b}_\alpha, \quad \alpha = x, y, z. \tag{27}$$

In Eq. (27), $\mathbf{b}_\alpha \in \mathbb{R}^N$, $\mathbf{D} \in \mathbb{R}^{N \times n_{\text{CP}}}$ and $\mathbf{G}, \mathbf{E} \in \mathbb{R}^{n_{\text{CP}} \times n_{\text{CP}}}$. In particular, the α coordinate of the TPs are collected in vector $\mathbf{b}_\alpha$:

$$\mathbf{b}_\alpha^{\text{T}} \coloneqq \{\alpha_1, \cdots, \alpha_N\}, \tag{28}$$

while matrices $\mathbf{D}$ and $\mathbf{G}$ collect the piecewise rational blending functions evaluated at the parametric coordinates (u_{1k}, u_{2k}), $k = 1, \cdots, N$ related to each TP (where u_{jk} are obtained through the SPM), i.e.

$$\mathbf{D} \coloneqq \begin{bmatrix} R_{0,0}(u_{11}, u_{21}) & \dots & R_{n_1,n_2}(u_{11}, u_{21}) \\ R_{0,0}(u_{12}, u_{22}) & \dots & R_{n_1,n_2}(u_{12}, u_{22}) \\ \vdots & \vdots & \vdots \\ R_{0,0}(u_{1N}, u_{2N}) & \dots & R_{n_1,n_2}(u_{1N}, u_{2N}) \end{bmatrix}, \ \mathbf{G} \coloneqq \mathbf{D}^{\text{T}}\mathbf{D}. \tag{29}$$

It is noteworthy that matrices $\mathbf{G}$ and $\mathbf{E}$ are symmetric and positive semidefinite, see [20] for more details. According to the guidelines provided in [20], the coefficient λ of Eqs. (20) and (27) has been set as follows:

$$\lambda = \frac{\|\mathbf{G}\|}{\|\mathbf{E}\|}, \tag{30}$$

where $\| \cdot \|$ is the l_2 norm, i.e. $\|\mathbf{M}\| = \left(\sum_{ij} M_{ij}^2\right)^{\frac{1}{2}}$.

Inasmuch as CPs coordinates are determined through Eq. (27), they can be excluded from the set of continuous design variables. Therefore, the rest of continuous design variables can be grouped in two different vectors:

- $\boldsymbol{\zeta}_2$ is related to the non-decreasing values of non-trivial KVs components $v_i^{(j)},\ i \in [p_j + 1, m_j - p_j - 1],\ j = 1, 2$;
- $\boldsymbol{\zeta}_3$ collects the values of weights w_{i_1,i_2}.

Nevertheless, since KVs are made of a non-decreasing sequence of real numbers, instead of directly using the non-trivial knots values $v_i^{(j)}$ as design variables, a more efficient choice consists in employing the following formula

$$v_i^{(j)} \coloneqq \beta_i^{(j)} v_{i-1}^{(j)} + \left(1 - \beta_i^{(j)}\right) v_{i+1}^{(j)},\ i \in [p_j + 1, m_j - p_j - 1],\ j = 1, 2, \tag{31}$$

where $\beta_i^{(j)}$ are the design variables that vary in the interval $]0,1[$. It must be noticed that Eq. (31) allows for avoiding the introduction of further constraint equations, during optimisation, to check if each KV is made of a non-decreasing sequence of real numbers. Therefore, the expression of vectors $\boldsymbol{\zeta}_2$ and $\boldsymbol{\zeta}_3$ reads:

$$\begin{aligned} \boldsymbol{\zeta}_2^{\mathrm{T}} &= \{\beta_{p_1+1}^{(1)}, \cdots, \beta_{m_1-p_1-1}^{(1)}, \beta_{p_2+1}^{(2)}, \cdots, \beta_{m_2-p_2-1}^{(2)}\} \Rightarrow \boldsymbol{\zeta}_2 \in \mathbb{R}^{r_1+r_2}, \\ \boldsymbol{\zeta}_3^{\mathrm{T}} &= \{w_{0,0}, \cdots, w_{n_1,n_2}\} \Rightarrow \boldsymbol{\zeta}_3 \in \mathbb{R}^{n_{\mathrm{CP}}}. \end{aligned} \tag{32}$$

According to Eqs. (18) and (32), the overall number of independent design variables (both integer and continuous) is:

$$N_{\mathrm{DV}} = 4 + r_1 + r_2 + (r_1 + p_1 + 1)(r_2 + p_2 + 1) \tag{33}$$

3.2. Problem Formulation and Numerical Strategy

As stated beforehand, in this work the surface fitting problem is formulated in the most general case, by integrating all independent design variables (both integer and continuous quantities) into the problem formulation. However, a quick glance to Eqs. (18), (32) and (33) suffices to understand that, regardless of the adopted formulation for the objective function and the optimisation constraints, the resulting problem is defined over a domain of variable dimension. In particular, the number of design variables N_{DV} (and, thus, the problem dimension) depends upon the optimal value of the components of vector $\boldsymbol{\zeta}_1$.

As discussed in [21, 31, 32, 34–37], this unconventional problem belongs to the class of optimisation problems dealing with *modular systems* belonging to different families. Roughly speaking, for a given family (or class) of modules, each module is characterised by the same design variables, which can take different values in the most general case of non-identical modules. When the goal is the simultaneous optimisation of the number of modules and of the design variables characterising each module, the resulting problem is defined over a domain of changing dimension, thus requiring a special formulation and a dedicated resolution strategy [21].

Following the approach proposed in [21, 34], in this study, a two-step optimisation process has been implemented to deal with the surface fitting problem. In this background, the general surface fitting problem is split into two sub-problems that are solved subsequently: each step of the process is characterised, hence, by a suitable problem formulation and the related optimisation algorithm. The first optimisation step consists of the *meta-heuristic exploration phase* (MEP) and aims at finding a suitable pseudo-optimal solution. During this step only integer parameters and KV components are considered as design variables and the exploration of the domain of changing dimension is carried out by means of the ERASMUS algorithm [21]. Then, the best solution of the MEP is used as initial guess for the subsequent *deterministic optimisation phase* (DOP) whose goal is to refine the pseudo-optimal solution resulting from the MEP. During the DOP, the integer variables are kept constants, while both KVs components and weights are included among the design variables. The DOP is articulated in two steps. Firstly, solely the KVs components are optimised, while the weights are kept constant. Secondly the KVs are kept equal to the values provided by the first step of the DOP and weights are optimised. For both steps the solution search is performed by means of the *active-set* algorithm of the MATLAB *fmincon* family, available in the MATLAB *optimization toolbox* [38]. The flow-chart of the optimisation process is illustrated in Fig. 3.

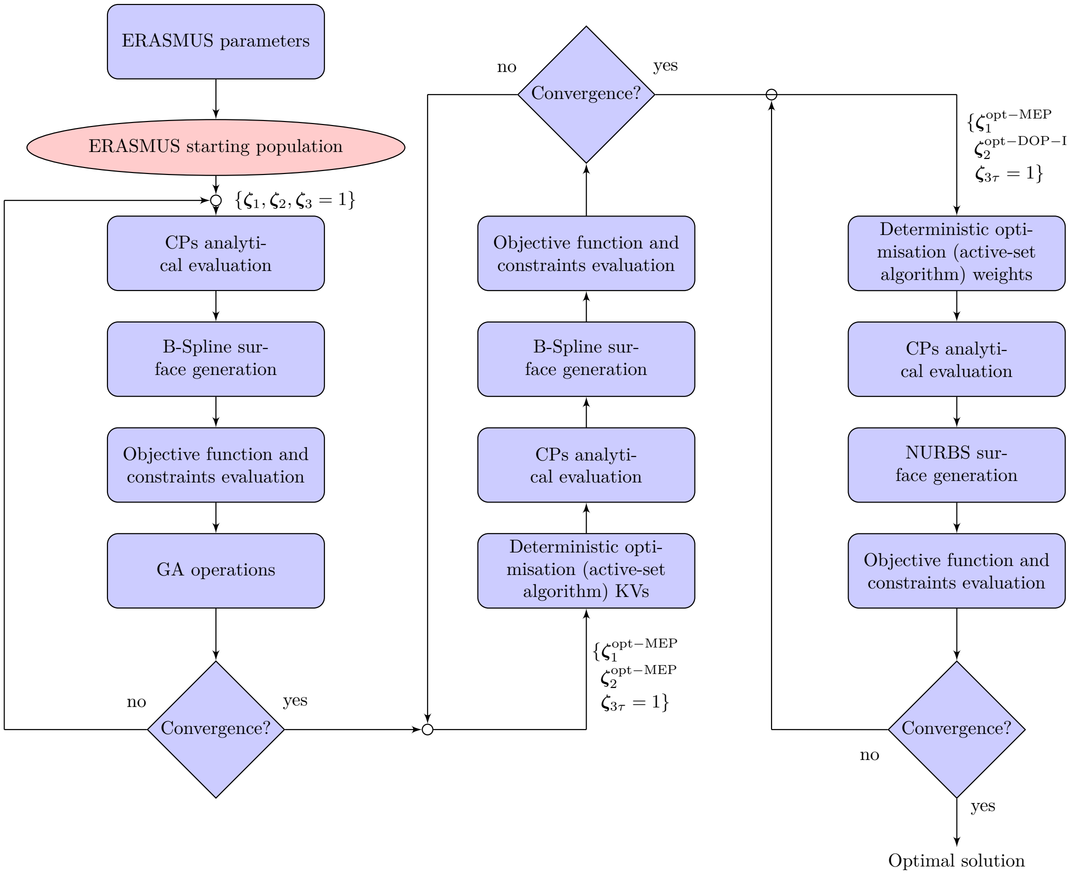

ERASMUS parameters
ERASMUS starting population
$\{\zeta_1, \zeta_2, \zeta_3 = 1\}$
CPs analytical evaluation
B-Spline surface generation
Objective function and constraints evaluation
GA operations
no
yes
Convergence?
$\{\zeta_1^{\mathrm{opt-MEP}}$
$\zeta_2^{\mathrm{opt-MEP}}$
$\zeta_{3\tau} = 1\}$
Deterministic optimisation (active-set algorithm) KVs
CPs analytical evaluation
B-Spline surface generation
Objective function and constraints evaluation
no
Convergence?
yes
$\{\zeta_1^{\mathrm{opt-MEP}}$
$\zeta_2^{\mathrm{opt-DOP-I}}$
$\zeta_{3\tau} = 1\}$
Deterministic optimisation (active-set algorithm) weights
CPs analytical evaluation
NURBS surface generation
Objective function and constraints evaluation
Convergence?
no
yes
Optimal solution

Figure 3: Flow chart of the the hybrid surface fitting strategy.

3.2.1. The Meta-Heuristic Exploration Phase

During the MEP, only $\boldsymbol{\zeta}_1$ and $\boldsymbol{\zeta}_2$ are considered as design variables, whilst $\boldsymbol{\zeta}_3$ is set equal to a unit vector (i.e. $\zeta_{3\tau} = 1, \forall \tau = 1, \cdots, n_{\mathrm{CP}}$) . According to Eqs. (7)-(8), this choice corresponds to consider a B-Spline surface instead of a NURBS one. To take into account for the variable number of KV components, the objective function has been modified with respect to Eq. (20), i.e.

$$\Phi(\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \boldsymbol{\zeta}_3) \coloneqq \tilde{f}^{\frac{1}{r_1+r_2}}, \text{ with :}$$
$$\tilde{f}(\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \boldsymbol{\zeta}_3) \coloneqq \sum_{\alpha=x,y,z} \sum_{k=1}^{N} \left[\frac{\alpha(u_{1k}, u_{2k}) - \alpha_k}{L_{\max}} \right]^2 + \lambda J_\alpha K_{\max}. \tag{34}$$

In Eq. (34), the quantities $L_{\max}$ and $K_{\max}$ are used to get a dimensionless objective function. In particular, $L_{\max}$ is the maximum distance between TPs, whilst $K_{\max}$ is the maximum Gaussian curvature evaluated on the tessellation. The quantity $\frac{1}{r_1+r_2}$ appears as a power of the function $\tilde{f}$ in order to find a good balance between KVs size and accuracy of the surface approximation. It is noteworthy that the role of the term J_α ($\alpha = x, y, z$) is twofold. On the one hand, it allows avoiding over-fitting by controlling the smoothness of the fitting surface. On the other hand, it allows defining a well-posed mathematical problem, because it limits the growth of the degrees p_j ($j = 1, 2$) of the blending functions during optimisation [20, 34].

As stated above, the optimal value of the CPs coordinates is the solution of the linear system of Eq. (27). However, for some particular combinations of degrees, number and values of KVs components (i.e. p_j, r_j and $v_i^{(j)}$, repsectively) the matrix $\mathbf{G} + \lambda\mathbf{E}$ in Eq. (27) could posses almost null eigenvalues, so its inversion could be ill-conditioned. In order to overcome this issue, a check on the possible singularity of this matrix has to be performed to ensure the presence of dimensionless parameters u_{jk} ($j = 1, 2,\ k = 1, \cdots, N$) in each knot span [13, 20]. Accordingly, a constraint on the rank of matrix $\mathbf{G} + \lambda\mathbf{E}$ must be introduced, i.e.

$$g(\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \boldsymbol{\zeta}_3) \coloneqq n_{\mathrm{CP}} - \mathrm{rank}\left(\mathbf{G} + \lambda\mathbf{E}\right) = 0. \tag{35}$$

Finally, the optimisation problem for the MEP can be stated in the form of an unconventional CNLPP as

$$\begin{aligned}
&\min_{\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2} \quad \Phi(\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \boldsymbol{\zeta}_3),\\
&\text{subject to:}\\
&\begin{cases}
g(\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \boldsymbol{\zeta}_3) = 0,\\
\boldsymbol{\zeta}_{1-\mathrm{lb}} \leq \boldsymbol{\zeta}_1 \leq \boldsymbol{\zeta}_{1-\mathrm{ub}},\\
0 \leq \zeta_{2k} \leq 1, \quad k = 1, \cdots, r_1 + r_2,\\
\zeta_{3\tau} = 1, \quad \forall\tau = 1. \cdots, n_{\mathrm{CP}},
\end{cases}
\end{aligned} \tag{36}$$

where $\boldsymbol{\zeta}_{1-\mathrm{lb}}$ and $\boldsymbol{\zeta}_{1-\mathrm{ub}}$ are suitable lower and upper bounds on the integer design variables. It is noteworthy that problem (36) is defined over a domain of changing dimension whose size (which corresponds to the number of design variables) is $N_{\mathrm{DV-MEP}} = 4 + r_1 + r_2$.

The solution search for problem (36) is carried out by means of the ERASMUS algorithm [21]. In this context, a B-Spline surface can be considered as a modular system where r_j and p_j are the design variables of the non-modular part of the system, whilst each KV $\beta^{(j)}$ represents the generic module whose variables are $v_i^{(j)}$, $i = p_j + 1, ..., m_j - p_j - 1$, $j = 1, 2$. A B-Spline surface is, thus, composed of two modules corresponding to the KVs.

In ERASMUS, an individual represents a candidate solution for the problem at hand. The individual's genotype of ERASMUS for problem (36) is illustrated in Fig. 4. As it can be inferred from this figure, the genotype is made of three sections. The first one is the *standard section*, which is made of two chromosomes constituted of two genes coding the integer design variables of the non-modular part of the B-Spline surface, i.e. p_j, r_j, $j = 1, 2$. Second and third sections are *modular sections*. The number of chromosomes of each modular section is equal to the number of non-trivial knots r_j (coded within the standard section) and each chromosome is made of a single gene coding the variable $\beta_i^{(j)}$ related to the knot $v_i^{(j)}$ according to Eq. (31). Inasmuch as the value of r_j can be different for each individual, the length of the modular sections (i.e. the number of chromosomes) is not necessarily the same among the individuals belonging to the same populations. In ERASMUS the length of the modular section is related to the concept of *species*: individuals with a different number of chromosomes belong to different species. As a consequence, when the surface fitting problem is formulated in the most general case, the ERASMUS algorithm represent a very efficient tool to perform the solution search because it allows for the *simultaneous evolution* of both species and individuals. In other words, the evolution of the population is obtained by simulating the reproduction phase (through dedicated genetic operators) among individuals of the same species and among individuals belonging to different species. For a deeper insight in the matter, the interested reader is addressed to [21].

3.2.2. The Deterministic Optimisation Phase

As stated above, the DOP is split in two step. During the first step only KVs components are considered as design variables and the pseudo-optimal solution found at the end of the MEP, i.e. $\boldsymbol{\zeta}_1^{\mathrm{opt-MEP}}$, $\boldsymbol{\zeta}_2^{\mathrm{opt-MEP}}$, is used as initial guess. In particular integer variables are kept constant and equal to $\boldsymbol{\zeta}_1^{\mathrm{opt-MEP}}$, weights take unit value as in the case of MEP, whilst only vector $\boldsymbol{\zeta}_2$ represents the design variables vector of the first step of the DOP.

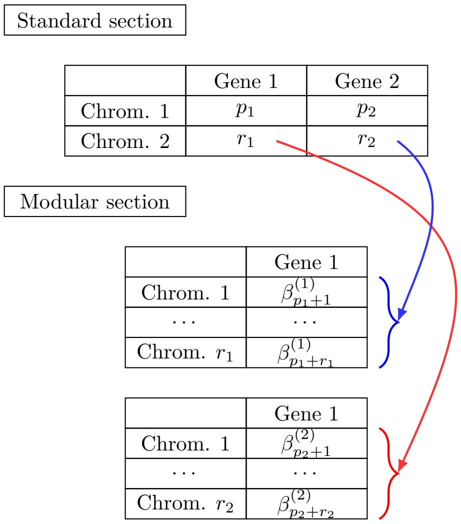
The surface fitting problem is stated as a conventional CNLPP as follows:

$$\min_{\boldsymbol{\zeta}_2} \quad \tilde{f}(\boldsymbol{\zeta}_1,\boldsymbol{\zeta}_2,\boldsymbol{\zeta}_3),$$

subject to:

$$\begin{cases} g(\boldsymbol{\zeta}_1,\boldsymbol{\zeta}_2,\boldsymbol{\zeta}_3)=0, \\ \boldsymbol{\zeta}_1=\boldsymbol{\zeta}_1^{\mathrm{opt-MEP}}, \\ 0 \leq \zeta_{2k} \leq 1, \quad k=1,\cdots,r_1+r_2, \\ \zeta_{3\tau}=1,\ \tau=1,\cdots,n_{\mathrm{CP}}, \end{cases} \tag{37}$$

where $\tilde{f}$ is defined in Eq. (34). In this case, the number of design variables does not change during the iterations and is equal to $N_{\mathrm{DV-DOP-I}}=r_1+r_2$.
The optimised solution, i.e. $\boldsymbol{\zeta}_2^{\mathrm{opt-DOP-I}}$, together with the values of the $\boldsymbol{\zeta}_1^{\mathrm{opt-MEP}}$ and $\zeta_{3\tau}=1,\ \forall\ \tau$ is used as initial guess for the second step of the DOP. In this case only NURBS weights, collected



Standard section

	Gene 1	Gene 2
Chrom. 1	p_1	p_2
Chrom. 2	r_1	r_2

Modular section

	Gene 1
Chrom. 1	$\beta^{(1)}_{p_1+1}$
⋯	⋯
Chrom. r_1	$\beta^{(1)}_{p_1+r_1}$

	Gene 1
Chrom. 1	$\beta^{(2)}_{p_2+1}$
⋯	⋯
Chrom. r_2	$\beta^{(2)}_{p_2+r_2}$

Figure 4: Individual's genotype.

in the vector $\boldsymbol{\zeta}_3$, are considered as design variables, the remeaning quantities being equal to the optimal values provided by the previous optimisation calculations. For the second step of the DOP the CNLPP reads:

$$\begin{aligned}
&\min_{\boldsymbol{\zeta}_3} \quad \tilde{f}(\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \boldsymbol{\zeta}_3),\\
&\text{subject to:}\\
&\begin{cases}
g(\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \boldsymbol{\zeta}_3) = 0,\\
\boldsymbol{\zeta}_1 = \boldsymbol{\zeta}_1^{\text{opt}-\text{MEP}},\\
\boldsymbol{\zeta}_2 = \boldsymbol{\zeta}_2^{\text{opt}-\text{DOP}-\text{I}},\\
\boldsymbol{\zeta}_{3-\text{lb}} \leq \boldsymbol{\zeta}_3 \leq \boldsymbol{\zeta}_{3-\text{ub}},
\end{cases}
\end{aligned} \tag{38}$$

where $\boldsymbol{\zeta}_{3-\text{lb}}$ and $\boldsymbol{\zeta}_{3-\text{ub}}$ are lower and upper bounds on the weights. As stated above, the goal of the DOP is to reach the nearest feasible minimiser starting from the pseudo-optimal solution found at the end of the MEP. To this end, the active-set algorithm available in the MATLAB *fmincon* function [38] has been used to perform the solution search. In order to solve problems (37) and (38) by means of the active-set algorithm, the derivatives of the objective function $\tilde{f}$ with respect to KVs components and CPs weights must be computed. The analytical expression of these derivatives is provided in Appendix A.

Remark 3.1. *Since the optimisation constraint $g(\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \boldsymbol{\zeta}_3)$ of Eq. (35) is a discontinuous function related to the rank of matrix $\mathbf{G} + \lambda\mathbf{E}$, its gradient is not evaluated during the DOP. In particular a preliminary check is done before evaluating the objective function and its gradient: if constraint $g(\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \boldsymbol{\zeta}_3)$ is not met, the objective function is penalised to a high value.*

4. General Strategy for Genus G Surfaces (Open and Closed)

The approach presented in Sections 2 and 3 can be applied to open surfaces of genus $g = 0$. Nevertheless, by introducing some modifications it can be extended also to the most general case of open and closed surfaces of genus $g = G > 0$, i.e. open and closed surfaces characterised by handles and/or holes. To this purpose, a semi-automatic multi-step procedure for surface reconstruction of complex surfaces of genus G, greater than zero, is proposed in this Section.

The whole procedure is articulated in three steps. The first one is the manual segmentation (to be performed by the user) of the complex discrete surface available in the form of a tesselation. The goal is to split the tessellation in suitable patches meeting the requirements foreseen by SPM (see Section 2.2) in order to perform the mapping of each patch. The segmentation, as shown in Fig. 5, consists of partitioning the tessellation in open patches of genus $g = 0$. Furthermore, the user has to properly define the sorting of patches and, for each patch, the a set of four corners, which represent the extremes of the unit square wherein each patch is mapped through the SPM. Thanks to this requirement, after carrying out the mapping phase by means of the SPM, consecutive patches will have consistent parametric (dimensionless) coordinates at adjacent (coincident) edges.

The second step is the mapping phase, which is carried out for each patch according to the SPM discussed in Section 2.2.

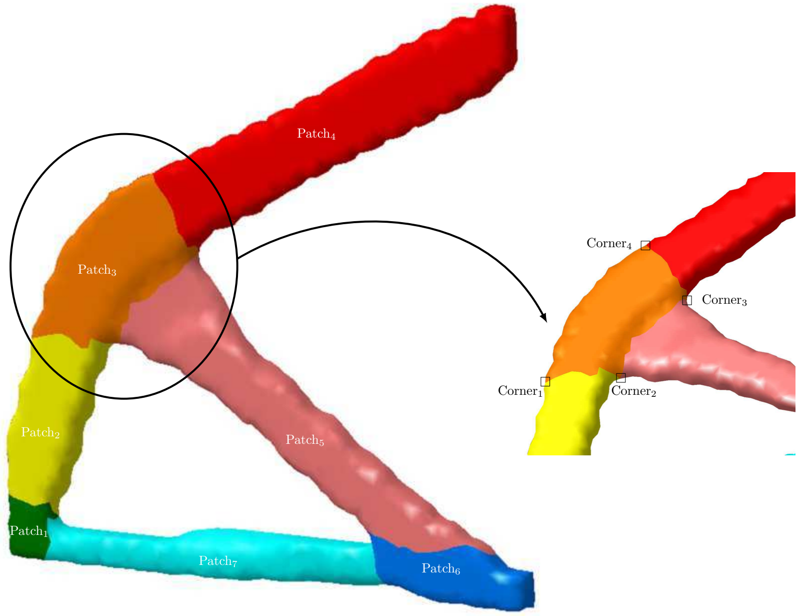


Figure 5: Manual segmentation.

Then, the information obtained from these steps are exploited in the fitting phase (the third and last step of the procedure), which is performed on a set of opportunely connected NURBS surfaces (denoted as poly-NURBS entity in the following). The fitting phase is articulated in two steps. Firstly, the connectivity map between the NURBS surfaces composing the poly-NURBS entity is defined according to the user-defined patch sorting introduced in the segmentation phase. The connectivity map matrix $\mathbf{M}_{\mathrm{conn}} \in \mathbb{R}^{n_p \times n_p}$ (n_p being the number of patches defined during the segmentation step), establishing the relationship between the NURBS surfaces fitting adjacent patches, is built as follows: the element ij is equal to the ID of the edge shared between patches i and j if they are adjacent, otherwise it is zero.

Secondly, C0 and C1 conditions are imposed between adjacent NURBS surfaces by following the order defined by the connectivity matrix $\mathbf{M}_{\mathrm{conn}}$ through a master-slave approach , as shown in Fig. 6. This operation is articulated in the following two steps.

1. C0 continuity between adjacent NURBS surfaces is ensured via the equivalence of the CPs coordinates for those CPs located on the boundary between the master patch and the surrounding slave patches.

2. C1 continuity is imposed by forcing the collinearity of the rows of CPs located on the edges shared between consecutive patches by means of the following formula

$$\begin{cases} \alpha^{\mathrm{s}}_{i+1,k_1} = 2\alpha^{\mathrm{m}}_{i+1,k_1} - \alpha^{\mathrm{m}}_{n_1,k_1}, \text{ with } k_1 = 0,\ldots,n_2, \\ \alpha^{\mathrm{s}}_{k_2,j+1} = 2\alpha^{\mathrm{m}}_{k_2,j+1} - \alpha^{\mathrm{m}}_{k_2,n_2}, \text{ with } k_2 = 0,\ldots,n_1. \end{cases} \tag{39}$$

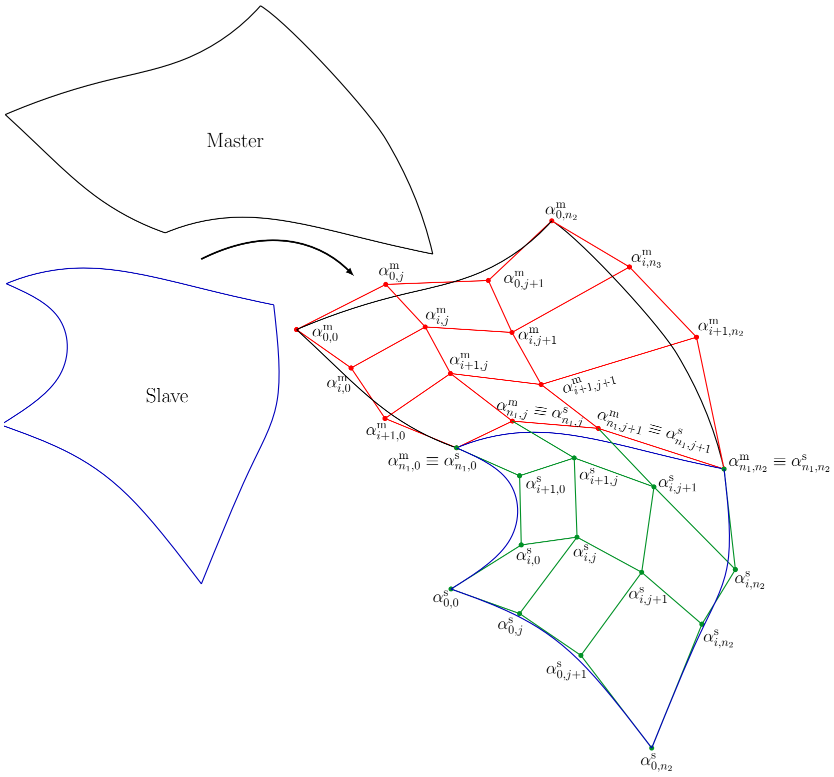

Master
Slave
$\alpha^{\mathrm{m}}_{0,n_2}$
$\alpha^{\mathrm{m}}_{i,n_3}$
$\alpha^{\mathrm{m}}_{0,j}$
$\alpha^{\mathrm{m}}_{0,j+1}$
$\alpha^{\mathrm{m}}_{i,j}$
$\alpha^{\mathrm{m}}_{0,0}$
$\alpha^{\mathrm{m}}_{i,j+1}$
$\alpha^{\mathrm{m}}_{i+1,n_2}$
$\alpha^{\mathrm{m}}_{i+1,j}$
$\alpha^{\mathrm{m}}_{i,0}$
$\alpha^{\mathrm{m}}_{i+1,j+1}$
$\alpha^{\mathrm{m}}_{n_1,j} \equiv \alpha^{\mathrm{s}}_{n_1,j}$
$\alpha^{\mathrm{m}}_{n_1,j+1} \equiv \alpha^{\mathrm{s}}_{n_1,j+1}$
$\alpha^{\mathrm{m}}_{i+1,0}$
$\alpha^{\mathrm{m}}_{n_1,0} \equiv \alpha^{\mathrm{s}}_{n_1,0}$
$\alpha^{\mathrm{m}}_{n_1,n_2} \equiv \alpha^{\mathrm{s}}_{n_1,n_2}$
$\alpha^{\mathrm{s}}_{i+1,0}$
$\alpha^{\mathrm{s}}_{i+1,j}$
$\alpha^{\mathrm{s}}_{i,j+1}$
$\alpha^{\mathrm{s}}_{i,0}$
$\alpha^{\mathrm{s}}_{i,j}$
$\alpha^{\mathrm{s}}_{i,n_2}$
$\alpha^{\mathrm{s}}_{0,0}$
$\alpha^{\mathrm{s}}_{i,j+1}$
$\alpha^{\mathrm{s}}_{0,j}$
$\alpha^{\mathrm{s}}_{i,n_2}$
$\alpha^{\mathrm{s}}_{0,j+1}$
$\alpha^{\mathrm{s}}_{0,n_2}$

Figure 6: C0 and C1 connectivity conditions.

Once the NURBS surfaces composing the poly-NURBS entity are properly connected, problems (36) and (37) - (38) are solved in cascade. The algorithm of the semi-automatic multi-step procedure is presented here below.

It is noteworthy that, during the MEP, the connectivity between adjacent patches requires also a correct definition of the independent design variables (both integer and continuous) between patches, as shown in Fig. 7. In the case of the poly-NURBS entity, the overall number of independent design variables can be obtained as:

$$N_{\mathrm{DV-MEP}} = 4n_p - 2\sum_{k=1}^{2n_p}\sum_{l=1}^{2n_p} Z_{kl} + \sum_{k=1}^{2n_p} r_k - a_k, \tag{40}$$

where $\mathbf{r}, \mathbf{a} \in \mathbb{R}^{2n_p}$ and $\mathbf{Z} \in \mathbb{R}^{2n_p \times 2n_p}$ are particular arrays defined as

$$\mathbf{r}^{\mathrm{T}} \coloneqq \{r_1^{(1)}, \cdots, r_1^{(n_p)}, r_2^{(1)}, \cdots, r_2^{(n_p)}\}, \tag{41}$$

$$\mathbf{Z} \coloneqq \frac{1}{2}\left(\mathbf{M} + \overline{\mathbf{M}}\right), \ \mathbf{a} \coloneqq \mathbf{Z}\mathbf{r}, \tag{42}$$

Algorithm 1 Genus G surface reconstruction strategy

```
1. Manual segmentation of the tessellation: creation of patches of genus 0; corners between adjacent patches must coincide.
2. ∀ patch:
    (a) Performing the mapping phase according to the SPM (Section 2.2).
    (b) Ensuring the coincidence of parameters at each edge shared between adjacent patches.
3. Build the connectivity matrix $\mathbf{M}_{\text{conn}}$ for the poly-NURBS entity to be used during the fitting phase.
4. Patches roto-translation, to align the local reference systems with the global one, to preserve the data coherence in the fitting phase.
5. Imposition of C0 and C1 continuity between adjacent NURBS surfaces.
6. Solve problems (36) and (37) - (38) to define NURBS parameters. During the optimisation CPs coordinates are updated according to Eq. (27).
```

where matrices $\mathbf{M}$ and $\overline{\mathbf{M}}$ are defined as

$$
\begin{aligned}
&M_{kl} \coloneqq \begin{cases} 1, \text{ if } \mathbf{v}_k^{(1)} = \mathbf{v}_l^{(1)}, \ \forall\ k, l = 1, \cdots, n_p, \\ 1, \text{ if } \mathbf{v}_k^{(1)} = \mathbf{v}_l^{(2)}, \ \forall\ k = 1, \cdots, n_p,\ l = n_p + 1, \cdots, 2n_p, \\ 1, \text{ if } \mathbf{v}_k^{(2)} = \mathbf{v}_l^{(2)}, \ \forall\ k, l = n_p + 1, \cdots, 2n_p, \\ 0, \ \text{otherwise}, \end{cases} \\
&M_{lk} = M_{kl},
\end{aligned}
\tag{43}
$$

$$
\begin{aligned}
&\overline{M}_{kl} \coloneqq M_{kl},\ k = 1, \cdots, 2n_p,\ l = k, \cdots, 2n_p, \\
&\overline{M}_{lk} = -\overline{M}_{kl}.
\end{aligned}
\tag{44}
$$

An example of the design variables inheritance scheme among patches is illustrated in Fig. 7.

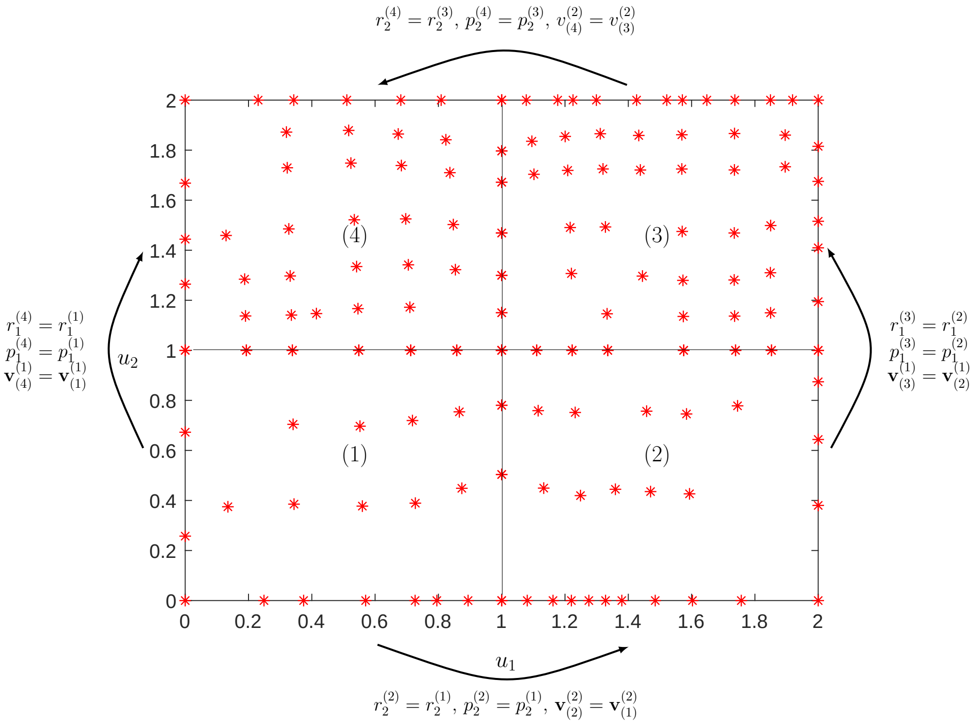


Figure 7: Design variables inheritance among patches (red stars denote the parameters u_1 and u_2 resulting from the SPM for each patch).

5. Studied Cases and Results

The effectiveness of the proposed strategy is tested on some meaningful benchmarks taken from the literature. These study cases focus on the surface reconstruction problem of both genus zero and genus $G > 0$ surfaces. In particular, five benchmark problems are illustrated and solved in this section: (BK1) the carpet-like surface, (BK2) the ear surface [39], (BK3) the face surface [40], (BK4) the thigh-bone surface [41], (BK5) a genus $g = 1$ surface, representing a region of the boundary of an optimised topology resulting from a 3D topology optimisation problem taken from [22]. The number of TPs, for the different test cases, is provided in Table 1.

	BK1	BK2	BK3	BK4				BK5			
patch n.	-	-	-	1	2	3	4	1	2	3	4
N	142	5630	15276	1998	1449	4350	2169	1741	1243	1187	1254

Table 1: Number of TPs for each benchmark.

The design variables and their respective bounds are listed in Table 2, for each benchmark considered in this study.

Benchmark	$[p_{1-\mathrm{lb}}, p_{1-\mathrm{ub}}]$	$[p_{2-\mathrm{lb}}, p_{2-\mathrm{ub}}]$	$[r_{1-\mathrm{lb}}, r_{1-\mathrm{ub}}]$	$[r_{2-\mathrm{lb}}, r_{2-\mathrm{ub}}]$	$[\mathbf{v}^{(1)}_{\mathrm{lb}}, \mathbf{v}^{(1)}_{\mathrm{ub}}]$	$[\mathbf{v}^{(2)}_{\mathrm{lb}}, \mathbf{v}^{(2)}_{\mathrm{ub}}]$	$[w_{ij-\mathrm{lb}}, w_{ij-\mathrm{ub}}]$
BK1	$[1, 6]$	$[1, 6]$	$[1, 17]$	$[1, 17]$	$[0.001, 0.999]$	$[0.001, 0.999]$	$[1.0, 10.0]$
BK2			$[16, 35]$	$[16, 35]$			
BK3			$[16, 35]$	$[16, 35]$			
BK4			$[5, 20]$	$[5, 20]$			
BK5			$[5, 20]$	$[5, 20]$			

Table 2: Design variables bounds.

The parameters tuning the behaviour of the ERASMUS algorithm, used during the MEP to solve problem (36), are listed in Table 3. Moreover, the handling of optimisation constraints is carried out through the automatic dynamic penalization (ADP) technique, see [42]. The genotype of the individual, representing a candidate solution for problem (36), is illustrated in Fig. 4. Lower and upper bounds on the number of chromosomes of the modular sections appearing in Table 3 are taken equal to lower and upper bounds of integer variables r_i $(i = 1, 2)$ reported in Table 2.
The parameters of the active-set algorithm, used during the DOP to solve problems (37) and (38), are given in Table 4. It is noteworthy that the quantity N_{DV} represents the number of design variables, which is not the same between first and second steps of the DOP.

Genetic parameters	Value
Number of populations (N_{pop})	1
Number of individuals (N_{ind})	200
Number of generations (N_{gen})	150
Crossover probability (p_{cross})	0.85
Gene mutation probability (p_{mut})	$1/N_{\mathrm{ind}}$
Chromosomes shift probability (p_{shift})	0.5
Chromosomes number mutation probability ($p_{\mathrm{mut-ch}}$)	$\dfrac{n_{\mathrm{ch-ub}} - n_{\mathrm{ch-lb}}}{N_{\mathrm{ind}}}$
Selection	Roulette-wheel
Elitism	Active

Table 3: Genetic parameters of the ERASMUS algorithm

Parameter	Value
Solver	*active-set*
Maximum number of objective function evaluations	$100 \times N_{\mathrm{DV}}$
Maximum number of iterations	1000
Tolerance on objective function	1×10^{-6}
Tolerance on constraints	1×10^{-6}
Tolerance on input variables change	10^{-6}
Tolerance on gradient norm of the Lagrange's function	10^{-6}

Table 4: *Active-set* algorithm parameters

The numerical results, for each case, are collected in Tables 5 and 6, for MEP and DOP, respectively. For the sake of clarity, the results of the DOP reported in Table 6 are expressed in terms of normalised objective function. Regarding the first step of the DOP, the optimised solution provided by the MEP is used to normalise the merit function of Eq. (37). Analogously, for the second step of the DOP, i.e. Eq. (38), the optimal solution of the first step of the DOP is used for normalisation purposes. Therefore, the normalised merit functions (for both steps) read:

$$\tilde{f}_{\mathrm{DOP-I}} := \frac{\tilde{f}}{\tilde{f}_{\mathrm{opt-MEP}}},\ \tilde{f}_{\mathrm{DOP-II}} := \frac{\tilde{f}}{\tilde{f}_{\mathrm{opt-DOP-I}}}. \tag{45}$$

The term $\tilde{f}_{\mathrm{av}}$ reported in Table 6 represents the average value of the objective function of Eq. (34), which can be related to the average (dimensionless) distance of the surface from the TPs. This term is defined as:

$$\tilde{f}_{\mathrm{av}} := \frac{\sqrt{\tilde{f}}}{N}. \tag{46}$$

Benchmark	p_1	p_2	r_1	r_2	n_1	n_2	Φ
BK1	5	5	1	1	6	6	0.0944
BK2	2	2	19	19	21	21	0.9635
BK3	2	2	18	18	20	20	1.1500
BK4 - patch 1	2	2	5	5	7	7	4.3974
BK4 - patch 2	2	5	5	6	7	8	
BK4 - patch 3	2	5	5	12	7	17	
BK4 - patch 4	2	5	5	6	7	11	
BK5 - patch 1	3	2	5	6	8	8	3.0333
BK5 - patch 2	3	3	5	5	8	8	
BK5 - patch 3	3	5	5	5	8	10	
BK5 - patch 4	3	3	5	5	8	8	

Table 5: MEP: numerical results.

The Carpet-like surface (BK1)

The first benchmark consists of a genus zero open surface. The STL file of BK1 has been generated in CATIA® environment. The amount of TPs composing the STL file is provided in Table 1. Firstly, the mapping of the TPs cloud has been obtained via the SPM. Secondly the surface fitting phase is performed by considering a single NURBS entity and by solving, in cascade, problems (36)-(38). The related optimal solutions are reported in Tables 5 and 6, while the optimal NURBS surface obtained at the end of the process, together with the related TPs cloud and the mapping resulting from the SPM, is illustrated in Fig. 8.
As it can been inferred from these results, during the MEP, the $N = 142$ TPs are fitted by a unique NURBS surface having $p_1 = p_2 = 5$ and only $r_1 = r_2 = 1$ non-trivial KVs components.

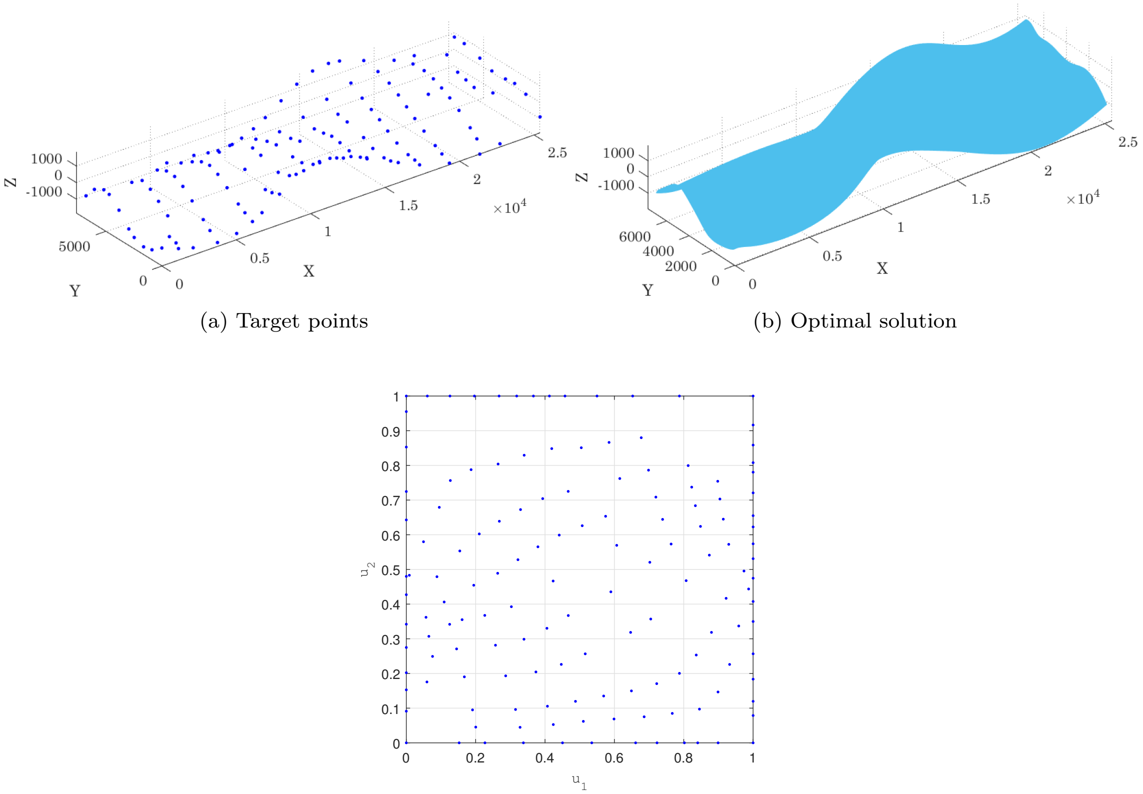
Problem	$\tilde{f}_{\mathrm{DOP-I}}$	$\tilde{f}_{\mathrm{DOP-II}}$	$\tilde{f}_{\mathrm{av}}$
BK1	0.0860	0.0844	$5.66e^{-5}$
BK2	0.9546	0.8923	$8.10e^{-5}$
BK3	0.0882	0.9596	$1.02e-^{4}$
BK4 - patch 1	0.3124	0.7488	$2.11e^{-4}$
BK4 - patch 2			
BK4 - patch 3			
BK4 - patch 4			
BK5 - patch 1	0.0975	0.0696	$8.77e^{-6}$
BK5 - patch 2			
BK5 - patch 3			
BK5 - patch 4			

Table 6: DOP: numerical results.

This correspond to an overall number of CPs equal to $(n_1 + 1)(n_2 + 1) = 49$. This result is due to the formulation of the objective function Φ of Eq. (34), where the main purpose of the power $\frac{1}{r_1+r_2}$ is to limit the KVs sizes. This result is very interesting, essentially for two reasons. Firstly, the ratio of the TPs number to the CPs number is about 2.9, which implies a significant reduction in the information needed to describe such a surface with a good level of accuracy. Secondly, due to the low values of variables r_i, few design variables are involved in the subsequent DOP, which means reduced computational costs. Moreover, from the results reported in Table 6, one can infer that the optimisation steps constituting the DOP allow obtaining a strong reduction of the pseudo-optimal solution provided by the MEP. In particular, at the end of the first step of the DOP the improvement is of about 91.4%, whilst at the end of the second step is of about 91.6%. This means that KVs components and weights plays a key-role in the quality of the optimised NURBS surface.

The Ear Surface (BK2)

The second benchmark is a genus zero open surface of complex shape, whose projection over a plane has a non-unique solution for some points. The STL file, taken from [39], has been elaborated in CATIA® environment and is composed of a cloud of $N = 5630$ TPs. As in the case of BK1, also for this example a unique NURBS surface is employed for the fitting phase.
The TPs cloud, the surface parametrisation resulting from the application of the SPM and the optimal NURBS surface resulting from the optimisation process are illustrated in Fig. 9. As it can been inferred from the results provided in Tables 5 and 6, a NURBS surface with a control net made of 484 CPs is sufficient to fit the set of TPs with a good level of accuracy. Indeed, at the end of the MEP, the pseudo-optimal solution is characterised by a very good value of the merit function: the improvement due to the first step of the DOP is of about 4.5%, while that of the second step (over the first one) is equal to 11% (the effect of weights on the smoothness of the surface remains important). Moreover, the ratio of the TPs number to the CPs number is about 11.63, which implies a strong reduction in the information needed to describe such a surface without degrading too much the accuracy. From the analysis of the results listed in Table 5-6 and from a visual inspection of Fig. 9, one can infer that the smoothing term of Eq. (21) fulfils its main



(a) Target points

(b) Optimal solution

(c) Parametrisation

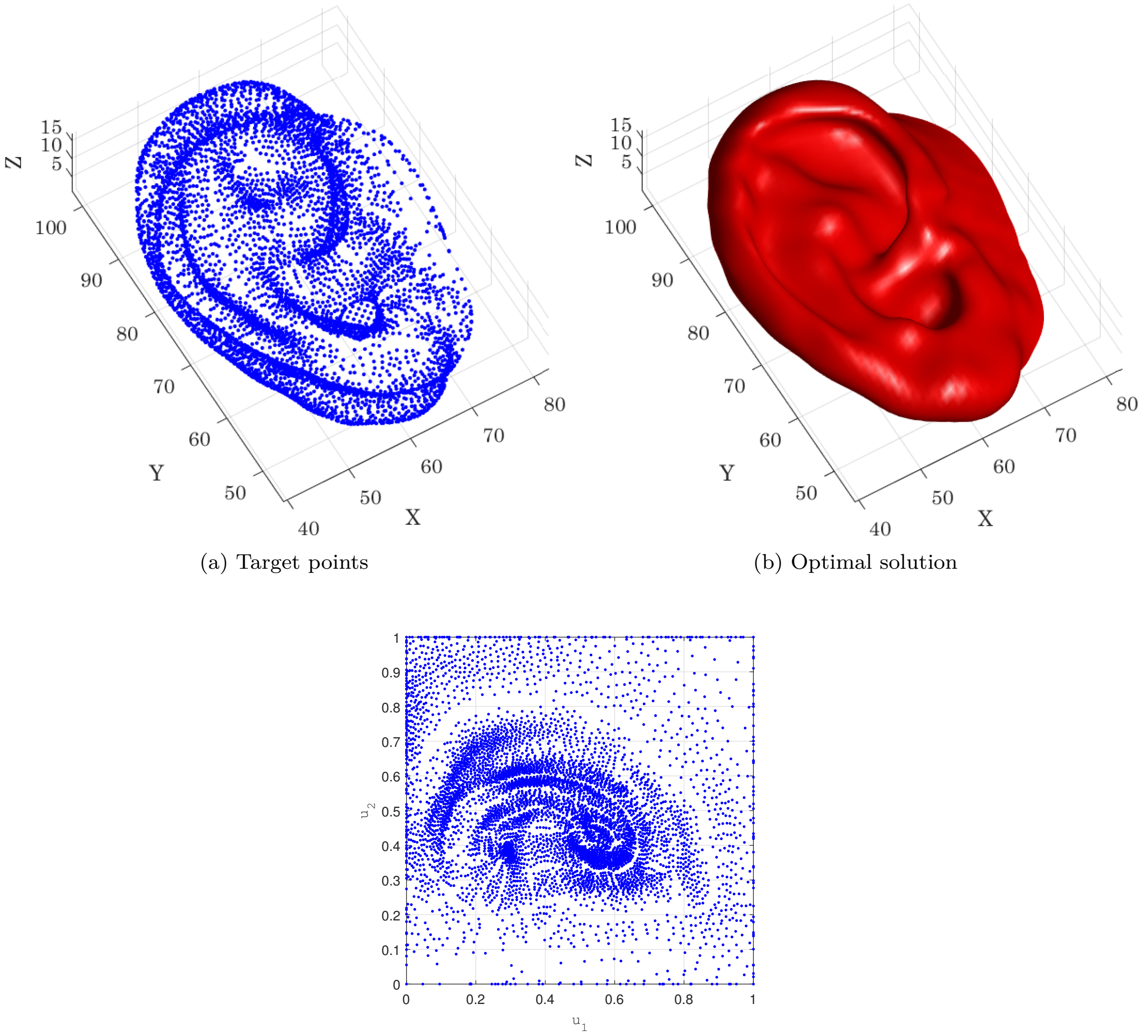
Figure 8: The Carpet-like Surface (BK1)

purpose by controlling the value of the degrees of the final fitting surface. Of course, in this case, the size of the KVs is bigger than that of the KVs of BK1, because of the complex shape of the TPs cloud.

The Face Surface (BK3)

This benchmark deals with the surface reconstruction of a genus zero open surface having the same complexity as that of BK2. The STL file, taken from [40], has been elaborated in CATIA® environment and is composed of a cloud of $N = 15276$ TPs. Also in this case only one NURBS surface is employed for the fitting phase.

The TPs cloud, the surface parametrisation resulting from the application of the SPM and the optimal NURBS surface are illustrated in Fig. 10. As it can been inferred from the results provided in Tables 5 and 6, a NURBS surface with 441 CPs is sufficient to fit the set of TPs with a good level of accuracy. However, at the end of the MEP, the pseudo-optimal solution is still located far away from the local minimizer: the improvement due to the first step of the DOP is of about 91%, while that of the second step (over the first one) is of about 4%. Moreover, the ratio of the TPs number to the CPs number is about 34.64, which implies a strong reduction in the information needed to



(a) Target points

(b) Optimal solution

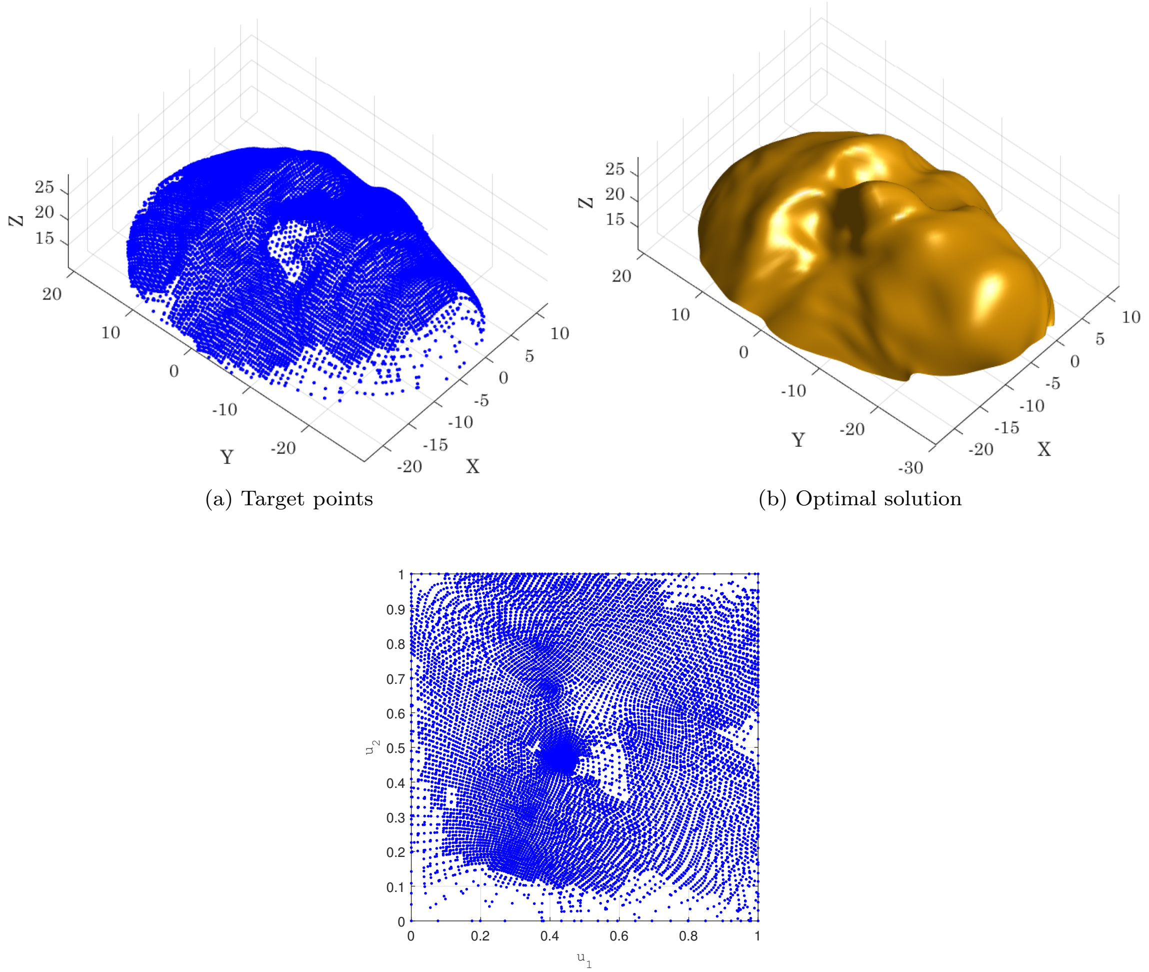
(c) Parametrisation

Figure 9: The Ear Surface (BK2)

describe such a surface without degrading too much the accuracy.
Regarding, the effect of the smoothing term of Eq. (21), the same remarks as those of benchmark BK2 can be repeated here.

The Thigh-bone Surface (BK4)

This benchmark consists of a genus zero open surface with a complex topology characterised by sub-domains having inhomogeneous shapes with protrusions and strong curvatures gradients.
To deal with the surface reconstruction of BK4, the general strategy for genus G surfaces, discussed



(a) Target points

(b) Optimal solution

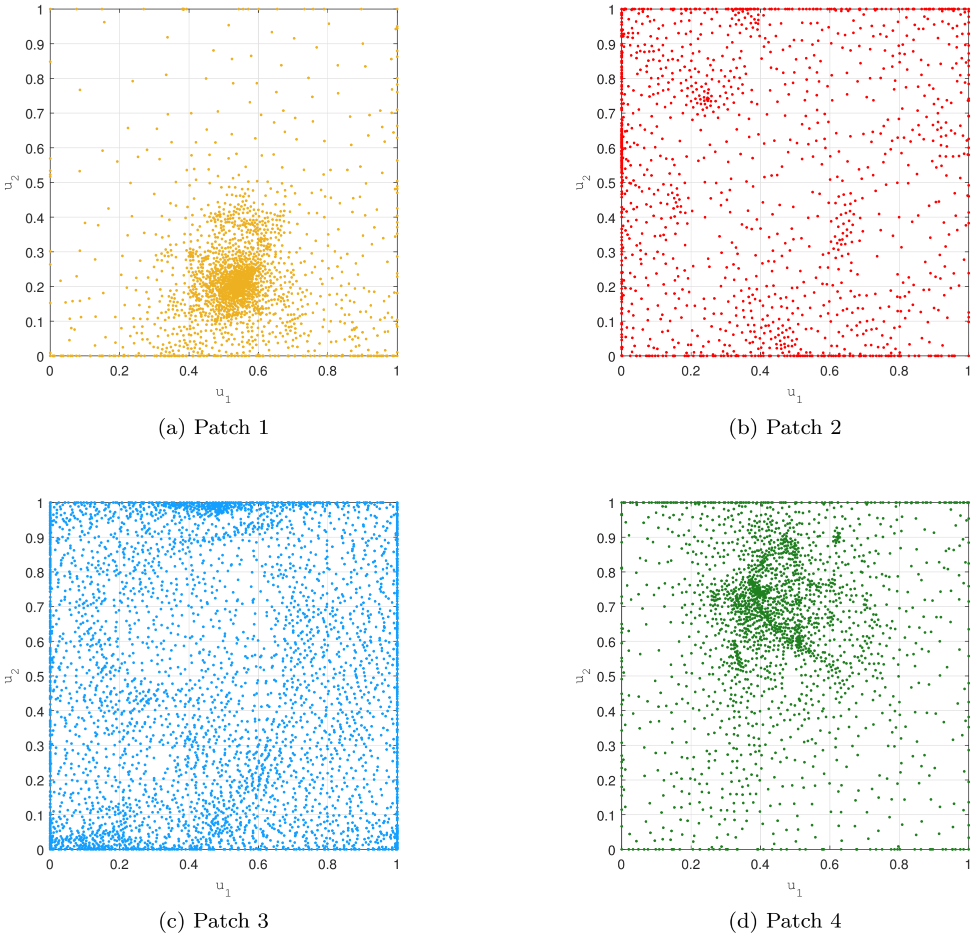
(c) Parametrisation

Figure 10: The Face Surface (BK3)

in Section 4, has been employed. To this purpose, the STL file, taken from [41], has been elaborated in CATIA® environment and it has been manually split in four patches for an overall number of $N = 9966$ TPs, see Table 1. Of course, the segmentation of the tessellation is a manual step whose results depends upon the user's experience: in this case the patches have been defined in order to isolate the thigh-bone protrusion and epicondyles, in order to have sufficient information to correctly carry out the mapping phase.

The results of the SPM to get the surface parametrisation (for each patch) are illustrated in Fig. 11, while the TPs cloud and the optimal NURBS surfaces at the end of the MEP and of the DOP are shown in Fig. 12. As it can been inferred from the results provided in Tables 5 and 6, the pseudo-optimal solution found at the end of the MEP is located far away from the local minimizer:

the improvement due to the first step of the DOP is of about 69%, while that of the second step (over the first one) is of about 25% (this result confirms the importance of the weights in influencing the shape of the NURBS surface for each patch). Moreover, the ratio of the TPs number to the CPs number is 18.72, 15.95, 30.21 and 22.59 for patches 1, 2, 3 and 4, respectively. As for the other benchmarks, this result implies a strong reduction in the information needed to describe such a surface by keeping a sufficient level of accuracy.
Regarding, the effect of the smoothing term of Eq. (21), the same remarks as those of benchmarks BK2 and BK3 can be repeated here, mainly for the patches describing protrusions and epicondyles.



(a) Patch 1

(b) Patch 2

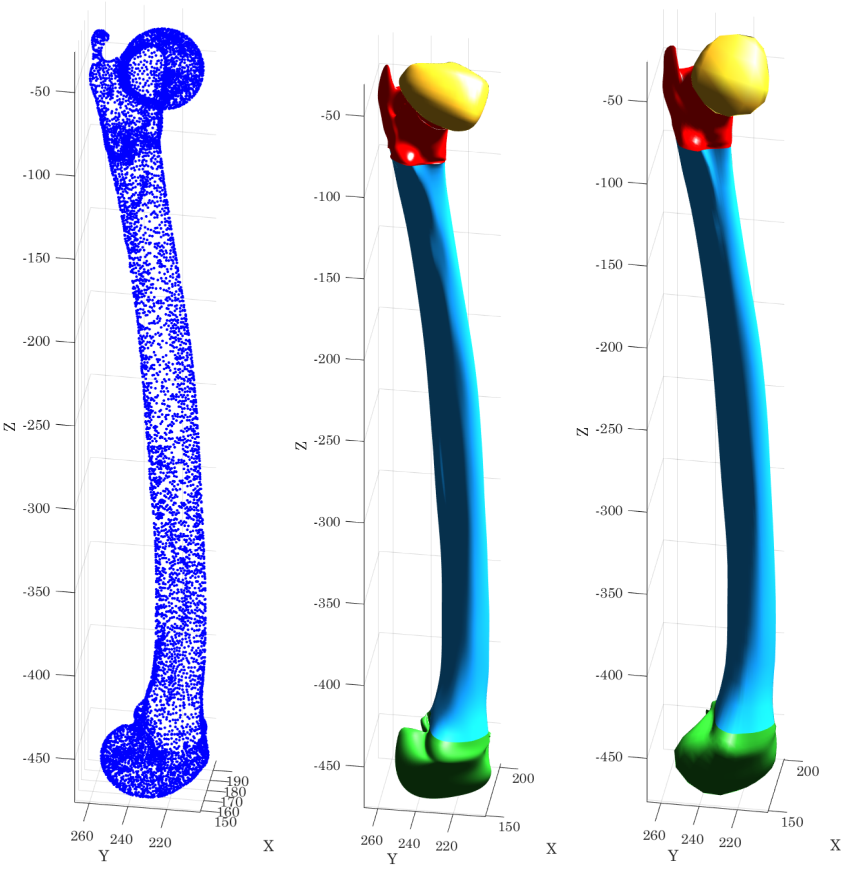
(c) Patch 3

(d) Patch 4

Figure 11: The thigh-bone surface (BK4) - parametrisation through the SPM

A Genus One Open Surface (BK5)

The fifth benchmark is a genus $g = 1$ open surface representing a subdomain of the boundary of a 3D optimised topology (minmising the compliance of a 3D domain subject to given boundary



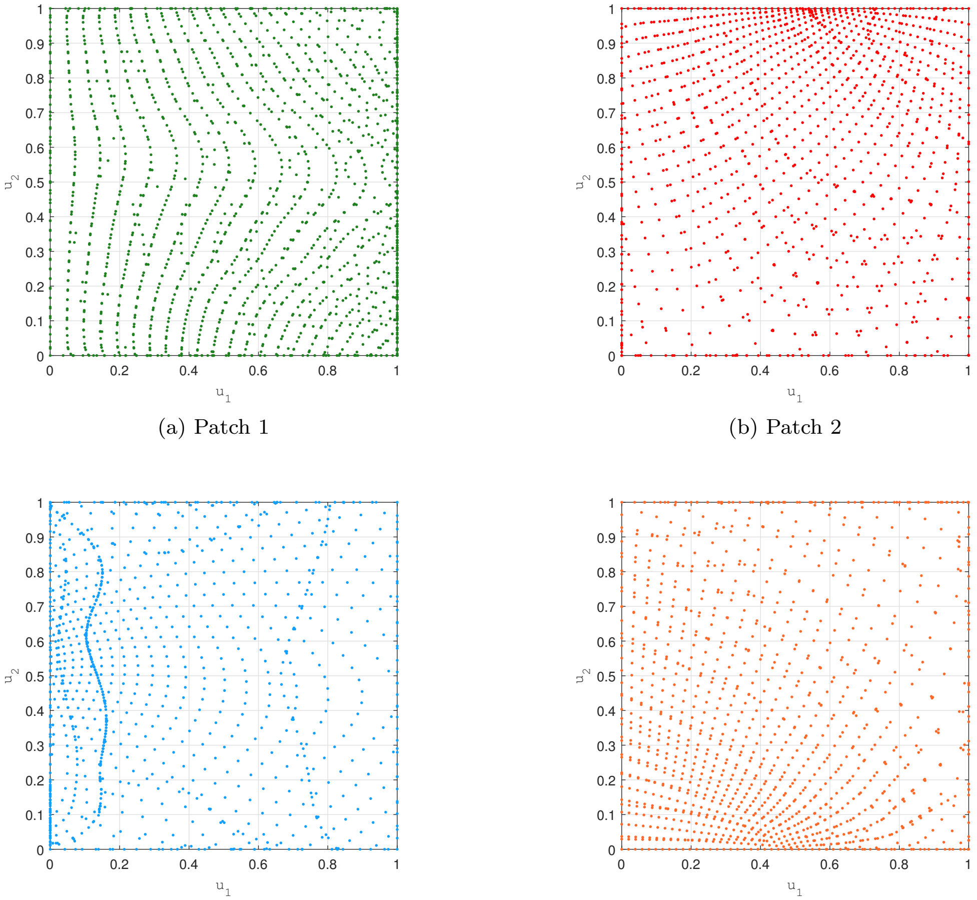
(a) Target points (b) Optimal solution - MEP (c) Optimal solution - DOP

Figure 12: The thigh-bone surface (BK4)

conditions and to a requirement on the overall volume) taken from [22].

To deal with the surface reconstruction of BK5, the general strategy for genus G surfaces is considered. To this purpose, the STL file has been elaborated in CATIA® environment and it has been manually split in four patches for an overall number of $N = 5425$ TPs, as indicated in Table 1. The results of the SPM to get the surface parametrisation (for each patch) are illustrated in Fig. 13, while the TPs cloud and the optimal NURBS surfaces at the end of the MEP and of the DOP are shown in Fig. 14. As it can been inferred from the results provided in Tables 5 and 6, the pseudo-optimal solution found at the end of the MEP is located far away from the local minimizer: the improvement due to the first step of the DOP is of about 90%, while that of the second step (over the first one) is of about 93% (this result confirms the importance of the weights in influencing the shape of the NURBS surface for each patch). Moreover, the ratio of the TPs number to the CPs number is 21.49, 15.35, 11.99 and 15.48 for patches 1, 2, 3 and 4, respectively. As for the other benchmarks, this result implies a strong reduction in the information needed to describe such a

surface by keeping a sufficient level of accuracy.
Regarding, the effect of the smoothing term of Eq. (21), the same remarks as those of the other benchmarks can be repeated here.



(a) Patch 1

(b) Patch 2

(c) Patch 3

(d) Patch 4

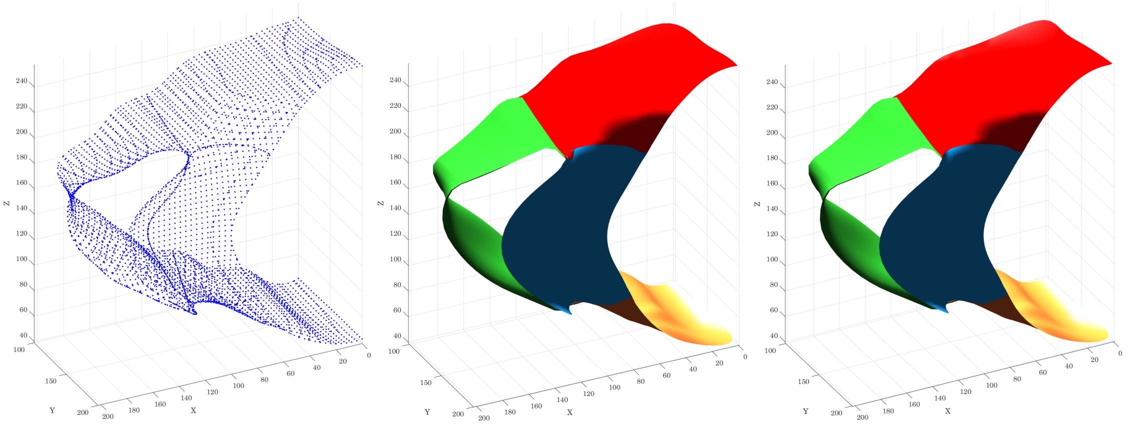
Figure 13: A genus $g = 1$ open surface (BK5) - parametrisation through the SPM

Discussion on the Design Variables Bounds

The choice of proper bounds of design variables has a strong impact on the result of the surface fitting problem. Therefore, some remarks inherent to the definition of these bounds, reported in Table 2, are provided here below. In particular, lower and upper bounds have been established according to the following considerations.

Continuous design variables bounds are simple to set.

- The knot vector components are defined between 0 and 1.



(a) Target point

(b) Optimal solution - MEP

(c) Optimal solution - DOP

Figure 14: A genus $g = 1$ open surface (BK5)

- The weights of the NURBS curve can get, *a priori*, any real value in the range $]0, \infty[$. After a preliminary check on the first three benchmarks, it was observed that the surface shape is affected by the ratio $w_{ij-\text{ub}}/w_{ij-\text{lb}}$ rather than by the value of the weight related to each CP. Moreover, the influence of weights become significant only in presence of singularities or strong curvatures gradients. Taking into account these considerations, it has been set $w_{ij-\text{lb}} = 1$ and $w_{ij-\text{ub}} = 10$.

Unlike continuous design variables, the *discrete design variables* have a major influence on the shape of the NURBS surface and their bounds must be carefully set.

- The minimum degree is, of course, $p_{j-\text{lb}} = 1$, $j = 1, 2$. The maximum degree has been fixed in order to avoid the introduction of noise that can become important when the upper bound is not properly set. Accordingly, the maximum degree has been set to $p_{j-\text{ub}} = 6$.

- In order to establish lower and upper bounds for the number of the non-trivial KVs components r_j, $j = 1, 2$ the user should think about an ideal number of CPs tuning the shape of the approximating NURBS surface. Indeed, this problem applies also in case of standard curve/surface fitting methods (which are not capable of automatically optimise discrete parameters), where the user does not dispose of any criterion to choose a suitable number of CPs. In the framework of the proposed method, the ERASMUS algorithm is able to automatically determine the optimum number of both KVs components and degrees of the basis functions and, thus, the related optimal number of CPs $n_{\text{CP}} = (p_1 + r_1 + 1)(p_2 + r_2 + 1)$. Of course, the bounds on variables r_j can be inferred according to empirical rules (taken from practice), utilised to define a criterion for setting the minimum and maximum number of control points. In particular, the bounds on n_j can be set according to the following rules:

  1. usually, the number of TPs should be, at least, three times the overall number of CPs;
  2. a suitable interval can be defined around this average value; in particular, the maximum number of CPs along each parametric direction must be lower than the number of TPs, whilst the minimum one should be always greater than or equal to 2.

Since the proposed hybrid surface reconstruction strategy (and the related optimisation algorithms) is very efficient, it can be asserted that it is not important to choose the “right" narrow interval. When the shape of the surface is particularly complex and does not let the user guess the size of the interval, a wider range can be set, being the GA ERASMUS able to determine automatically the optimum value of the discrete parameters. Indeed, thanks to the special features of the ERASMUS algorithm [21] it can be stated that the user has a lower impact in the context of the proposed surface reconstruction approach when compared to classical ones.

6. Conclusions

In this paper, an efficient and general surface reconstruction strategy for open and closed surfaces of genus greater than or equal to zero has been presented. This strategy is, indeed, able of fitting convex and non-convex sets of TPs and is articulated into two phases: the *mapping phase* and the *fitting phase*. The proposed approach relies on the following features.

- The mapping phase makes use of the SPM method to retrieve a proper parametrisation of each sub-domain composing the TPs cloud.
- The fitting phase is formulated in the most general sense, i.e. without introducing simplifying hypotheses or rules on the set of parameters affecting the shape of the NURBS surface, which is used as a parametric entity to fit the set of TPs. To this purpose, a new expression of the objective function (taking into account for both integer and continuous variables tuning the shape of the NURBS entity), together with a suitable constraint on the non-singularity of the blending functions matrix, has been introduced. Moreover, the problem is stated as a special CNLPP wherein the number of unknowns is included among the design variables. In fact, when integer parameters (i.e. degrees and number of knots along each parametric direction) are included together with the continuous ones (i.e. the knots values and the weights) among the design variables, the resulting CNLPP is defined over a space of changing dimension, thus requiring a special optimisation algorithm to find a feasible solution.
- The non-convexity of the problem and the fact that the design space has a changing dimension are the two main reasons at the basis of the use of advanced numerical strategies to solve the related CNLPP. To this purpose, the solution search for the surface fitting problem is performed by means of a hybrid optimisation tool composed by the union of the GA ERASMUS coupled to a deterministic algorithm. The optimisation strategy is split in two step. The first one is the MEP, where the ERASMUS algorithm is used to determine, simultaneously, the best value of both integer and continuous design variables. Thanks to a two-level Darwinian strategy (allowing for the simultaneous evolution of species and individuals) this algorithm is able to find a solution for a CNLPP defined over a space of variable dimension (i.e. with a variable number of design variables). The aim of the MEP is to provide the starting guess for the subsequent DOP where only continuous variable are optimised, while integer one are kept constant.
- The fitting phase (and the related CNLPP formulation) has also been generalised to the case of open and closed surfaces of genus greater than zero where an assembly of NURBS surfaces (referred as poly-NURBS entities), properly connected, is used to fit the non-convex set of TPs.

The effectiveness of the proposed approach is proven through some meaningful benchmarks taken from the literature.

As far as prospects of this work are concerned, two challenges still need to be faced. The first one is the development of a completely automatic segmentation strategy to define the patches in which the TPs cloud must be split. The aim of this strategy is to reduce the user's arbitrary decisions that could affect the shape of the patches (and, thus, the result of the mapping phase and the quality of the fitting surface). The second challenge consists of a general formulation of the segmentation problem as a constrained optimisation problem. In particular, what is the best strategy to split a given tessellation? What is the optimal number of patches ensuring a correct parametrisation of the whole tessellation? What is the optimal segmentation strategy minimising the distortion and the error of the final fitting surface? Research is ongoing on the above aspects.

Acknowledgements

This work benefited from the support of the project COFFA ANR-17-CE10-0008 of the French National Research Agency (ANR).

Appendix A. Analytical expression of the gradient of the objective function

The derivation of the analytical expression of the objective function gradient of Eq. (34) with respect to the continuous design variables, i.e. KVs components ($\boldsymbol{\zeta}_2$) and NURBS surface weights ($\boldsymbol{\zeta}_3$) is here presented. The gradient of $\tilde{f}$ reads:.

$$\frac{\partial \tilde{f}(\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \boldsymbol{\zeta}_3)}{\partial \boldsymbol{\zeta}_i} = \frac{1}{L_{\max}} \frac{\partial}{\partial \zeta_i} \left(\sum_{\alpha=x,y,z} \sum_{k=1}^{N} \left[\alpha(u_{1k}, u_{2k}) - \alpha_k\right]^2 \right) + \lambda K_{\max} \frac{\partial J_\alpha}{\partial \boldsymbol{\zeta}_i}, \qquad i = 2, 3. \tag{A.1}$$

The derivative of the first term of the right-hand side of Eq. (A.1) with respect to parameters $\beta_i^{(j)}$, which are related to the non-trivial KVs components according to Eq. (31), reads:

$$\begin{aligned}
&\frac{\partial}{\partial \beta_i^{(j)}} \left(\sum_{\alpha=x,y,z} \sum_{k=1}^{N} \left[\alpha(u_{1k}, u_{2k}) - \alpha_k\right]^2 \right) = \\
&\quad = \sum_{\alpha=x,y,z} \sum_{k=1}^{N} 2 \left[\alpha(u_{1k}, u_{2k}) - \alpha_k\right] \frac{\partial \alpha}{\partial \beta_i^{(j)}} = \\
&\quad = \sum_{\alpha=x,y,z} \sum_{k=1}^{N} 2 \left[\alpha(u_{1k}, u_{2k}) - \alpha_k\right] \frac{\partial \alpha}{\partial v_i^{(j)}} \frac{\partial v_i^{(j)}}{\partial \beta_i^{(j)}} = \\
&\quad = \sum_{\alpha=x,y,z} \sum_{k=1}^{N} 2 \left[\alpha(u_{1k}, u_{2k}) - \alpha_k\right] \frac{\partial \alpha}{\partial v_i^{(j)}} \left(v_{i-1}^{(j)} - v_{i+1}^{(j)} \right), \qquad i = p_j + 1, \dots, p_j + r_j, \quad j = 1, 2,
\end{aligned} \tag{A.2}$$

where $\dfrac{\partial \alpha}{\partial v_i^{(j)}}$ is the expression of the B-Spline surface derivatives with respect to the non-trivial KVs components, available in [43].

The analytical expression of the derivative with respect to $\beta_i^{(j)}$ of the second term of the right-hand side of Eq. (A.1) reads:

$$
\begin{aligned}
\frac{\partial J_\alpha}{\partial \beta_i^{(j)}} &= \int_0^1 \int_0^1 2\frac{\partial^2 \alpha}{\partial u_1^2}\frac{\partial}{\partial \beta_i^{(j)}}\left(\frac{\partial^2 \alpha}{\partial u_1^2}\right) + 4\frac{\partial^2 \alpha}{\partial u_1 \partial u_2}\frac{\partial}{\partial \beta_i^{(j)}}\left(\frac{\partial^2 \alpha}{\partial u_1 \partial u_2}\right) + 2\frac{\partial^2 \alpha}{\partial u_2^2}\frac{\partial}{\partial \beta_i^{(j)}}\left(\frac{\partial^2 \alpha}{\partial u_2^2}\right) \mathrm{d}u_1 \mathrm{d}u_2 = \\
&= \int_0^1 \int_0^1 2\frac{\partial^2 \alpha}{\partial u_1^2}\frac{\partial}{\partial v_i^{(j)}}\left(\frac{\partial^2 \alpha}{\partial u_1^2}\right)\frac{\partial v_i^{(j)}}{\partial \beta_i^{(j)}} + 4\frac{\partial^2 \alpha}{\partial u_1 \partial u_2}\frac{\partial}{\partial v_i^{(j)}}\left(\frac{\partial^2 \alpha}{\partial u_1 \partial u_2}\right)\frac{\partial v_i^{(j)}}{\partial \beta_i^{(j)}} + \\
&+ 2\frac{\partial^2 \alpha}{\partial u_2^2}\frac{\partial}{\partial v_i^{(j)}}\left(\frac{\partial^2 \alpha}{\partial u_2^2}\right)\frac{\partial v_i^{(j)}}{\partial \beta_i^{(j)}} \mathrm{d}u_1 \mathrm{d}u_2 = \\
&= \int_0^1 \int_0^1 2\frac{\partial^2 \alpha}{\partial u_1^2}\frac{\partial}{\partial v_i^{(j)}}\left(\frac{\partial^2 \alpha}{\partial u_1^2}\right)\left(v_{i-1}^{(j)} - v_{i+1}^{(j)}\right) + 4\frac{\partial^2 \alpha}{\partial u_1 \partial u_2}\frac{\partial}{\partial v_i^{(j)}}\left(\frac{\partial^2 \alpha}{\partial u_1 \partial u_2}\right)\left(v_{i-1}^{(j)} - v_{i+1}^{(j)}\right) + \\
&+ 2\frac{\partial^2 \alpha}{\partial u_2^2}\frac{\partial}{\partial v_i^{(j)}}\left(\frac{\partial^2 \alpha}{\partial u_2^2}\right)\left(v_{i-1}^{(j)} - v_{i+1}^{(j)}\right) \mathrm{d}u_1 \mathrm{d}u_2,
\end{aligned}
\tag{A.3}
$$

where $\frac{\partial}{\partial v_i^{(j)}}\left(\frac{\partial^2 \alpha}{\partial u_1^2}\right)$, $\frac{\partial}{\partial v_i^{(j)}}\left(\frac{\partial^2 \alpha}{\partial u_1 \partial u_2}\right)$ and $\frac{\partial}{\partial v_i^{(j)}}\left(\frac{\partial^2 \alpha}{\partial u_2^2}\right)$ are the derivatives of the second-order partial derivatives of the B-Spline surface with respect to the non-trivial KVs components. For example, in the case of the KV components along the first parametric coordinate, i.e. $v_k^{(1)}$ these derivatives read:

$$
\frac{\partial}{\partial v_k^{(1)}}\left(\frac{\partial^2 \alpha}{\partial u_1^2}\right) = \sum_{i_1=0}^{n_1+1}\sum_{i_2=0}^{n_2} \frac{\partial^2 \bar{N}_{i_1,p_1}(u_1)}{\partial u_1^2} N_{i_2,p_2}(u_2)\bar{\alpha}_{i_1,i_2}, \tag{A.4}
$$

$$
\frac{\partial}{\partial v_k^{(1)}}\left(\frac{\partial^2 \alpha}{\partial u_2^2}\right) = \sum_{i_1=0}^{n_1+1}\sum_{i_2=0}^{n_2} \bar{N}_{i_1,p_1}(u_1)\frac{\partial^2 N_{i_2,p_2}(u_2)}{\partial u_2^2}\bar{\alpha}_{i_1,i_2}. \tag{A.5}
$$

$$
\frac{\partial}{\partial v_k^{(1)}}\left(\frac{\partial^2 \alpha}{\partial u_1 u_2}\right) = \sum_{i_1=0}^{n_1+1}\sum_{i_2=0}^{n_2} \frac{\partial \bar{N}_{i_1,p_1}(u_1)}{\partial u_1}\frac{\partial N_{i_2,p_2}(u_2)}{\partial u_2}\bar{\alpha}_{i_1,i_2}, \tag{A.6}
$$

In the above equations the term $\bar{N}_{i_1,p_1}(u_1)$ represents the i_1-th basis function defined over the modified KV $\bar{\mathbf{v}}^{(1)}$, which is obtained by increasing the multiplicity of its non-trivial component $v_k^{(1)}$. Furthermore, $\bar{\alpha}_{i_1,i_2}$ is the generic component of the CPs matrix $\bar{\alpha}_{\mathrm{CP}}$, modified to take into account for the new knot added to $\bar{\mathbf{v}}^{(1)}$ (see [43] for more details):

$$
\bar{\alpha}_{i_1,i_2} = \begin{cases} 0, & i_1 = 0,\ldots,k-p_1-1,\ i_2 = 0,\cdots,n_2, \\ \dfrac{\alpha_{i_1-1,i_2} - \alpha_{i_1,i_2}}{v_{i_1+p_1}^{(1)} - v_{i_1}^{(1)}}, & i_1 = k-p_1,\ldots,k,\ i_2 = 0,\cdots,n_2, \\ 0, & i_1 = k+1,\ldots,n_1+1\ i_2 = 0,\cdots,n_2. \end{cases} \tag{A.7}
$$

The derivatives with respect to the components of the KV $\mathbf{v}^{(2)}$ can be obtained by following the same passages and are not reported here for the sake of brevity.

Consider, now the gradient of the first term of the right-hand side of Eq. (A.1), which represents the distance between the NURBS surface and the TPs, with respect to the weights. Its analytical expression can be easily derived as follows:

$$\begin{aligned}
&\frac{\partial}{\partial w_{i_1,i_2}}\left(\sum_{\alpha=x,y,z}\sum_{k=1}^{N}\left[\alpha(u_{1k},u_{2k})-\alpha_k\right]^2\right)= \\
&\quad=\sum_{\alpha=x,y,z}\sum_{k=1}^{N}2\left[\alpha(u_{1k},u_{2k})-\alpha_k\right]\frac{\partial\alpha}{\partial w_{i_1,i_2}}= \\
&\quad=\sum_{\alpha=x,y,z}\sum_{k=1}^{N}2\left[\alpha(u_{1k},u_{2k})-\alpha_k\right]\frac{N_{i_1,p_1}(u_{1k})N_{i_2,p_2}(u_{2k})}{W(u_{1k},u_{2k})}\left[\alpha_{i_1,i_2}-\alpha(u_{1k},u_{2k})\right],
\end{aligned} \tag{A.8}$$

where the function $W\left(u_{1k},u_{2k}\right)$ is defined as

$$W(u_{1k},u_{2k}) \coloneqq \sum_{i_1=0}^{n_1}\sum_{i_2=0}^{n_2}N_{i_1,p_1}(u_{1k})N_{i_2,p_2}(u_{2k})w_{i_1,i_2}, \tag{A.9}$$

The analytical expression of the gradient function of the term J_α with respect to NURBS weight is more complicated. The first passage leads to:

$$\frac{\partial J_\alpha}{\partial w_{i,j}}=\int_0^1\int_0^1 2\frac{\partial^2\alpha}{\partial u_1^2}\frac{\partial}{\partial w_{i,j}}\left(\frac{\partial^2\alpha}{\partial u_1^2}\right)+4\frac{\partial^2\alpha}{\partial u_1\partial u_2}\frac{\partial}{\partial w_{i_1,i_2}}\left(\frac{\partial^2\alpha}{\partial u_1\partial u_2}\right)+2\frac{\partial^2\alpha}{\partial u_2^2}\frac{\partial}{\partial w_{i,j}}\left(\frac{\partial^2\alpha}{\partial u_2^2}\right)\mathrm{d}u_1\mathrm{d}u_2, \tag{A.10}$$

where the derivatives of the surface second-order partial derivatives with respect to the weights can be expressed as:

$$\begin{aligned}
\frac{\partial^2}{\partial u_1^2}\left(\frac{\partial\alpha}{\partial w_{i,j}}\right)=&\frac{\partial^2N_{i,p_1}}{\partial u_1^2}N_{j,p_2}\frac{(\alpha_{i,j}-\alpha)}{W}+ \\
&-\frac{2}{W}\left[\frac{\partial\alpha}{\partial u_1}+\frac{\partial W}{\partial u_1}\frac{(\alpha_{i,j}-\alpha)}{W}\right]\left(\frac{\partial N_{i,p_1}}{\partial u_1}N_{j,p_2}-\frac{N_{i,p_1}N_{j,p_2}}{W}\frac{\partial W}{\partial u_1}\right)+ \\
&-\frac{N_{i,p_1}N_{j,p_2}}{W}\left[\frac{\partial^2\alpha}{\partial u_1^2}+\frac{\partial^2W}{\partial u_1^2}\frac{(\alpha_{i,j}-\alpha)}{W}\right],
\end{aligned} \tag{A.11}$$

$$\begin{aligned}
\frac{\partial^2}{\partial u_2^2}\left(\frac{\partial\alpha}{\partial w_{i,j}}\right)=&\frac{\partial^2N_{j,p_2}}{\partial u_2^2}N_{i,p_1}\frac{(\alpha_{i,j}-\alpha)}{W}+ \\
&-\frac{2}{W}\left[\frac{\partial\alpha}{\partial u_2}+\frac{\partial W}{\partial u_2}\frac{(\alpha_{i,j}-\alpha)}{W}\right]\left(\frac{\partial N_{j,p_2}}{\partial u_2}N_{i,p_1}-\frac{N_{i,p_1}N_{j,p_2}}{W}\frac{\partial W}{\partial u_2}\right)+ \\
&-\frac{N_{j,p_2}N_{i,p_1}}{W}\left[\frac{\partial^2\alpha}{\partial u_2^2}+\frac{\partial^2W}{\partial u_2^2}\frac{(\alpha_{i,j}-\alpha)}{W}\right],
\end{aligned} \tag{A.12}$$

$$
\begin{aligned}
\frac{\partial}{\partial u_1 u_2}\left(\frac{\partial^2 \alpha}{\partial w_{i,j}}\right) = {} & \frac{\partial N_{i,p_1}}{\partial u_1}\frac{\partial N_{j,p_2}}{\partial u_2}\frac{(\alpha_{i,i_2}-\alpha)}{W} - \frac{N_{i,p_1}N_{j,p_2}}{W}\left[\frac{\partial^2 \alpha}{\partial u_1 u_2} + \frac{\partial^2 W}{\partial u_1 u_2}\frac{(\alpha_{i,j}-\alpha)}{W}\right] + \\
& - \frac{\partial N_{i,p_1}}{\partial u_1}\frac{N_{j,p_2}}{W}\left[\frac{\partial \alpha}{\partial u_2} + \frac{\partial W}{\partial u_2}\frac{(\alpha_{i,j}-\alpha)}{W}\right] - \frac{\partial N_{j,p_2}}{\partial u_2}\frac{N_{i,p_1}}{W}\left[\frac{\partial \alpha}{\partial u_1} + \frac{\partial W}{\partial u_1}\frac{(\alpha_{i,j}-\alpha)}{W}\right] + \\
& + \frac{N_{i,p_1}N_{j,p_2}}{W^2}\left[\frac{\partial W}{\partial u_2}\frac{\partial \alpha}{\partial u_1} + \frac{\partial W}{\partial u_1}\frac{\partial \alpha}{\partial u_2} + \frac{2}{W}\frac{\partial W}{\partial u_1}\frac{\partial W}{\partial u_2}(\alpha_{i,j}-\alpha)\right].
\end{aligned}
\tag{A.13}
$$

In the above expressions, the partial derivatives of terms $W(u_1, u_2)$ and $N_{i_j,p_j}(u_j)$, with respect to the parametric coordinates u_j, $j = 1, 2$ are computed via the Algorithms A 3.8 - A 4.4 available in [13].

References

[1] A. S. K. Hormann, B. Lévy, Mesh Parameterization: Theory and Practice., INRIA (Aug. 2007).

[2] M. S. Floater, Mean value coordinates, Computer Aided Geometric Design 20 (1) (2003) 19–27. URL http://www.sciencedirect.com/science/article/pii/S0167839603000025

[3] X. Chen, A. Pang, Y. Zhu, Y. Li, X. Luo, G. Zhang, P. Wang, Y. Zhang, S. Li, J. Yu, Towards 3d human shape recovery under clothing, CoRR abs/1904.02601 (2019). arXiv:1904.02601. URL http://arxiv.org/abs/1904.02601

[4] E. F. Hefetz, E. Chien, O. Weber, A subspace method for fast locally injective harmonic mapping, Computer Graphics Forum 38 (2) (2019) 105–119.

[5] X. Gu, S.-T. Yau, Global Conformal Surface Parameterization, in: L. Kobbelt, P. Schroeder, H. Hoppe (Eds.), Eurographics Symposium on Geometry Processing, The Eurographics Association, 2003.
URL http://dx.doi.org/10.2312/SGP/SGP03/127-137

[6] S. D. Porumbescu, B. Budge, L. Feng, K. I. Joy, Shell maps, ACM Transactions on Graphics 24 (3) (2005) 626–633. doi:10.1145/1073204.1073239.

[7] B. Allen, B. Curless, Z. Popović, The space of human body shapes, in: ACM SIGGRAPH 2003 Papers on - SIGGRAPH '03, ACM Press, 2003. doi:10.1145/1201775.882311.

[8] H. Biermann, I. Martin, F. Bernardini, D. Zorin, Cut-and-paste editing of multiresolution surfaces, in: Proceedings of the 29th annual conference on Computer graphics and interactive techniques - SIGGRAPH '02, ACM Press, 2002. doi:10.1145/566570.566583.

[9] V. Kraevoy, A. Sheffer, Cross-parameterization and compatible remeshing of 3d models, in: ACM SIGGRAPH 2004 Papers on - SIGGRAPH '04, ACM Press, 2004. doi:10.1145/1186562.1015811.

[10] X. Gu, S.-T. Yau, Computing conformal structures of surfaces, Communications in Information and Systems 2 (2) (2002) 121–146. doi:10.4310/cis.2002.v2.n2.a2.

[11] W.-C. Li, N. Ray, B. Lévy, Automatic and interactive mesh to t-spline conversion, in: Proceedings of the Fourth Eurographics Symposium on Geometry Processing, SGP '06, Eurographics Association, Goslar, DEU, 2006, p. 191–200.

[12] D. Julius, V. Kraevoy, A. Sheffer, D-charts: Quasi-developable mesh segmentation, Computer Graphics Forum 24 (3) (2005) 581–590. doi:10.1111/j.1467-8659.2005.00883.x.

[13] L. Piegl, W. Tiller, The NURBS book, Springer, 1997.

[14] S. J. Rahi, K. Sharp, Mapping complicated surfaces onto a sphere, International Journal of Computational Geometry & Applications 17 (04) (2007) 305–329.

[15] M. S. Floater, Parametrization and smooth approximation of surface triangulations, Computer Aided Geometric Design 14 (3) (1997) 231–250.

[16] J. Tao, B. Deng, J. Zhang, A fast numerical solver for local barycentric coordinates, Computer Aided Geometric Design 70 (2019) 46–58.

[17] H. Lee, Y. Tong, M. Desbrun, Geodesics-based one-to-one parameterization of 3d triangle meshes, IEEE Multimedia 12 (1) (2005) 27–33.

[18] OpenCascade, Surfaces from Scattered Points User's Guide, version 7.3.0 Edition.

[19] Q. Mao, S. Liu, S. Wang, X. Ma, Surface fitting for quasi scattered data from coordinate measuring systems, Sensors 18 (2) (2018) 214.

[20] M. S. Floater, Meshless Parameterization and B-Spline Surface Approximation, in: The Mathematics of Surfaces IX, Springer London, 2000, pp. 1–18.

[21] M. Montemurro, A contribution to the development of design strategies for the optimisation of lightweight structures, Habilitation à diriger des recherches, Université de Bordeaux, France (12 2018).
URL http://hdl.handle.net/10985/15155

[22] G. Costa, M. Montemurro, J. Pailhès, NURBS hyper-surfaces for 3D topology optimization problems, Mechanics of Advanced Materials and Structures (may 2019).
URL https://doi.org/10.1080/15376494.2019.1582826

[23] G. Costa, M. Montemurro, J. Pailhès, A 2D topology optimisation algorithm in NURBS framework with geometric constraints, International Journal of Mechanics and Materials in Design 14 (4) (2018) 669–696.

[24] G. Costa, M. Montemurro, J. Pailhès, N. Perry, Maximum length scale requirement in a topology optimisation method based on NURBS hyper-surfaces, CIRP Annals 68 (2019) 153–156.

[25] G. Costa, M. Montemurro, J. Pailhès, Minimum length scale control in a NURBS-based SIMP method, Computer Methods in Applied Mechanics and Engineering 354 (2019) 963–989. doi:10.1016/j.cma.2019.05.026.

[26] T. Rodriguez, M. Montemurro, P. L. Texier, J. Pailhès, Structural Displacement Requirement in a Topology Optimization Algorithm Based on Isogeometric Entities, Journal of Optimization Theory and Applications 148 (2020) 259–276.

[27] G. Bertolino, M. Montemurro, G. D. Pasquale, Multi-scale shape optimisation of lattice structures: an evolutionary-based approach, International Journal on Interactive Design and Manufacturing 13 (4) (2019) 1565–1578.

[28] M. Montemurro, A. Catapano, D. Doroszewski, A multi-scale approach for the simultaneous shape and material optimisation of sandwich panels with cellular core, Composites Part B : Engineering 91 (2016) 458–472.

[29] M. Montemurro, A. Catapano, On the effective integration of manufacturability constraints within the multi-scale methodology for designing variable angle-tow laminates, Composite Structures 161 (2017) 145–159.

[30] M. Montemurro, A. Catapano, A general B-Spline surfaces theoretical framework for optimisation of variable angle-tow laminates, Composite Structures 209 (2019) 561–578.

[31] Y. Audoux, M. Montemurro, J. Pailhès, A surrogate model based on Non-Uniform Rational B-Splines hypersurfaces, CIRP Procedia 70 (2018) 463–468.

[32] Y. Audoux, M. Montemurro, J. Pailhès, Non-Uniform Rational Basis Spline hyper-surfaces for metamodelling. Computer Methods in Applied Mechanics and Engineering, Computer Methods in Applied Mechanics and Engineering (2020).
URL https://doi.org/10.1016/j.cma.2020.112918

[33] Y. Audoux, M. Montemurro, J. Pailhès, A metamodel based on non-uniform rational basis spline hyper-surfaces for optimisation of composite structures, Composite Structures 247 (2020). doi:10.1016/j.compstruct.2020.112439.

[34] G. Costa, M. Montemurro, J. Pailhès, A General Hybrid Optimization Strategy for Curve Fitting in the Non-uniform Rational Basis Spline Framework, Journal of Optimization Theory and Applications 176 (1) (2018) 225–251.

[35] M. Montemurro, A. Vincenti, P. Vannucci, Design of elastic properties of laminates with minimum number of plies, Mechanics of Composite Materials 48 (2012) 369–390.

[36] M. Montemurro, A. Vincenti, P. Vannucci, A two-level procedure for the global optimum design of composite modular structures - Application to the design of an aircraft wing. Part 1: theoretical formulation, Journal of Optimization Theory and Applications 155 (1) (2012) 1–23.

[37] M. Montemurro, A. Vincenti, P. Vannucci, A two-level procedure for the global optimum design of composite modular structures - Application to the design of an aircraft wing. Part 2: numerical aspects and examples, Journal of Optimization Theory and Applications 155 (1) (2012) 24–53.

[38] The Mathworks Inc., 3 Apple Hill Drive, Natick, Optimization Toolbox User's Guide (2018).

[39] STL Ear, https://www.cgstudio.com/.

[40] STL Face, https://grabcad.com/.

[41] STL Bone, https://3dprint.nih.gov.

[42] M. Montemurro, A. Vincenti, P. Vannucci, The automatic dynamic penalisation method (adp) for handling constraints with genetic algorithms, Computer Methods in Applied Mechanics and Engineering 256 (2013) 70 – 87.

[43] L. A. Piegl, W. Tiller, Computing the derivative of NURBS with respect to a knot, Computer Aided Geometric Design 15 (9) (1998) 925–934.