A metamodel based on non-uniform rational basis spline hyper-surfaces for optimisation of composite structures

Yohann Audoux, Marco Montemurro, Jérôme Pailhès

Version of Record: https://www.sciencedirect.com/science/article/pii/S0263822320309326
Manuscript_0555574b3967251c3a82fedf1d66a676

A Metamodel Based on Non-Uniform Rational Basis Spline Hyper-Surfaces for Optimisation of Composite Structures

Yohann Audoux[a], Marco Montemurro[a,*], Jérôme Pailhès[a]

[a] *Arts et Métiers Institute of Technology, Université de Bordeaux, CNRS, INRA, Bordeaux INP, HESAM Université, I2M UMR 5295, F-33405 Talence, France*

Abstract

This study presents a new strategy to generate a surrogate model used for design purposes. The metamodel is based on Non-Uniform Rational Basis Spline (NURBS) hyper-surfaces and is able to fit non-convex sets of target points (TPs). The proposed method aims at determining all the parameters involved in the definition of the NURBS hyper-surface, i.e. control points (CPs) coordinates, weights, degrees, CPs number and knot-vector components. To this purpose, the problem of finding a suitable metamodel is formulated as a constrained non-linear programming problem (CNLPP) wherein the above variables are optimised in order to fit a set of TPs. Nevertheless, when the number of CPs and the degrees of the basis functions are included among the design variables, the resulting problem is defined over a space having a variable dimension. This problem is solved by means of a special genetic algorithm able to determine simultaneously the optimum value of both the design space size (related to the integer variables of the NURBS hyper-surface) and the NURBS hyper-surface continuous parameters. The NURBS-based metamodel is then used to emulate the first buckling load of a composite stiffened panel and it is used in the framework of a meaningful design problem.

Keywords: Optimisation, Genetic Algorithms, NURBS hyper-surfaces, Surrogate models, Stiffened Panels, Composite structures

1. Introduction

Composite structures are massively used in many industrial fields because of their high stiffness-to-weight and strength-to-weight ratios that can lead to substantial gain of weight when compared to classical metallic alloys. However, heterogeneity and anisotropy of composite materials make the design of composite structures a hard task since the problem should be considered at different scales: (a) microscopic (i.e. that of the constitutive phases), (b) mesocopic (i.e. that of the constitutive lamina) and (c) macroscopic (i.e. that of the laminate).

Moreover, when dealing with the optimisation of composite structures, usually the physical responses involved into the problem formulation are the outputs of complex numerical models. Accordingly, the computational cost of theses responses can be high leading, thus, to a time consuming optimisation process. In this context, metamodelling strategies can be used to reduce the overall time needed to assess the physical responses of the structure. The metamodelling process consists of defining a suitable *surrogate* model requiring less

*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 (Marco Montemurro)

resources to be executed than the original model from which it is obtained. The meaning of resources depends on the problem at hand. As an example, image reduction aims to reduce the *number of data* needed to evaluate the metamodel [1], while metamodels used within an optimisation process aim to reduce the computatonal effort (i.e. the elapsed time) to evaluate the outputs for a given set of inputs [2, 3]. It is noteworthy that all current methods need a calibration phase that is excluded when evaluating the overall computational effort. For some complex real-world engineering problems, the calibration step can require a huge amount of time: in these circumstances the engineer should carefully evaluate the opportunity of formulating a metamodel. Consider a MIMO system characterised by N inputs and M outputs defined as:

$$\begin{array}{rccc} & \mathbb{R}^N & \rightarrow & \mathbb{R}^M \\ \mathbf{z}: & \mathbf{x} & \rightarrow & \mathbf{z}(\mathbf{x}), \end{array} \tag{1}$$

where $\mathbf{x} = \left(x^{(1)}, \ldots, x^{(N)}\right)$ is the vector collecting the N inputs of the system and $\mathbf{z}(\mathbf{x}) = \left(z^{(1)}(\mathbf{x}), \ldots, z^{(M)}(\mathbf{x})\right)$ is the transfer function of the MIMO system containing the M outputs. In some cases this function may be completely known but, for real-world engineering problems, this is not always true. From a mathematical viewpoint, the metamodelling process consists of determining a function $\hat{\mathbf{z}}(\mathbf{x})$ that needs less *resources* to be evaluated than $\mathbf{z}(\mathbf{x})$:

$$\begin{array}{rccc} & \mathbb{R}^N & \rightarrow & \mathbb{R}^M \\ \hat{\mathbf{z}}: & \mathbf{x} & \rightarrow & \hat{\mathbf{z}}(\mathbf{x}) = \mathbf{z}(\mathbf{x}) + \boldsymbol{\varepsilon}(\mathbf{x}), \end{array} \tag{2}$$

where $\hat{\mathbf{z}}(\mathbf{x})$ is the function approximating the real transfer function $\mathbf{z}(\mathbf{x})$ and $\boldsymbol{\varepsilon}(\mathbf{x})$ is the approximation error at point $\mathbf{x}$. The function $\boldsymbol{\varepsilon}(\mathbf{x})$ is a bounded function whose bounds are linked to the desired accuracy.

In the literature one can find a wide range of metamodelling techniques such as kriging [4], radial basis functions (RBFs) [5, 6], Artificial Neural Networks (ANNs) [7, 8], proper orthogonal decomposition (POD) [9, 10] and proper generalised decomposition (PGD) [1]. Some of these methods have been used within optimisation processes [11–15], but, as a general remark, setting the number and the value of the parameters tuning the behaviour of classical metamodelling techniques could be a quite difficult task, which often needs a trial-and-error approach.

The metamodelling strategy proposed in this study relies on the use of M-dimension (M-D) Non-Uniform Rational Basis Spline (NURBS) hyper-surfaces characterised by a N-D parametric space to fit a given set of data points, also called target points (TPs). Up to now, only few research works focus on the formulation/implementation of surrogate models based on the NURBS formalism [16–18]. NURBS curves and surfaces are standard geometrical entities widely used in Computer Aided Design (CAD) software. NURBS hyper-surfaces represent a generalisation of these entities. A NURBS hyper-surface is defined through the number of dimensions (related to the size of the problem at hand), the degree of blending-functions along each dimension, the overall number of control points (CPs), the coordinates of each CPs and the related weights, the knot vector components along each dimension. The large amount of parameters tuning the shape of a NURBS hyper-surface makes it a versatile tool for many mathematical and engineering applications, not only for formulating surrogate models [19–27]. However, this large amount of parameters also constitutes its main drawback: it is very hard to properly tune all these parameters without making some simplifying assumptions or preliminary choices as done in [16–18]. Up to now, hyper-surface fitting problems are solved by means of iterative pro-

cedures generalising those used in the surface fitting framework [28]. These procedures can be grouped into two categories. On the one hand, some procedures start from a minimal number of CPs which is iteratively increased until the algorithm reaches a given accuracy. On the other hand, some procedures make use of the opposed approach, i.e. they start from the maximum allowable number of CPs which are iteratively removed without degrading the desired accuracy.

The first attempt of using NURBS hyper-surfaces to formulate suitable metamodels goes back to the works of Turner [16, 18]. In particular, in [18] a NURBS-based surrogate model used in the framework of the characterisation of composite material properties is presented. In this background, Turner and Crawford developed an iterative procedure for the hyper-surface fitting problem focusing on the determination of a suitable number of CPs for the NURBS hyper-surface. However, the method proposed by Turner and Crawford is based upon an empirical rule to determine the position of new CPs that are added on the basis of the value of the cost function of the problem at hand (typically the maximum error of approximation). Indeed, their approach is characterised by some restrictions:

- the degrees of the NURBS blending functions are set *a priori*;
- the knot vectors components are uniformly distributed or calculated by means of simple empirical rules;
- the weights are evaluated through empirical formulae and only for those CPs whose local support contains TPs affected by a relative error greater than a given threshold.

Of course, these empirical rules (and the parameters tuning the behaviour of the related formulae) are strongly problem-dependent and the user must have a deep knowledge of the problem at hand and of the NURBS hyper-surfaces fundamentals, as well.

To go beyond the aforementioned restrictions, the metamodelling approach based on NURBS hyper-surfaces is here coupled with a special genetic algorithm [29–31] able of determining both the optimal number and the value optimal of the parameters affecting the NURBS hyper-surface shape, without introducing simplyfing hypotheses or empirical rules to set these quantities. In particular, when the number of CPs and the blending function degrees are included among the unknowns, the hyper-surface fitting problem can be formulated as a Constrained Non-Linear Programming Problem (CNLPP) defined over a space of variable dimension. Of course, when dealing with such a problem, a particular care must be put in the choice of the proper numerical tool to perform the solution search. To this purpose the ERASMUS code (EvolutionaRy Algorithm for optimiSation of ModUlar Systems) [29, 30], which is able to deal with problems characterised by a variable number of design variables, is used as optimisation tool to solve the hyper-surface fitting problem. The effectiveness of the proposed surrogate model based on NURBS hyper-surfaces is proved through a meaningful benchmark taken from the literature [32]: the least-weight design of a composite stiffened panel. In this context, the metamodel is used to emulate the first-buckling load of the stiffened panel as a function of the considered design variables.

The paper is organised as follows: the theoretical framework of NURBS hyper-surfaces is presented in Section 2. The mathematical formulation of the hyper-surface fitting problem is briefly presented in Section 3, while the hybrid optimisation tool used for solving the metamodel generation problem is presented in Section 4. The optimisation problem related to the stiffness panel is presented in Section 5. The result of the metamodelling generation process are discussed in Section 6, whilst Section 7 presents the results of the optimisation of the stiffened panel using the metamodel. Finally Section 8 ends the paper with some concluding remarks and prospects.

2. NURBS hyper-surfaces: theoretical framework

The fundamentals of NURBS entities are briefly provided in the most general case of NURBS hyper-surfaces. Curves and surfaces formulae, widely discussed in [28, 33–35], can be easily deduced from the following formulæ.

A NURBS hyper-surface is a polynomial-based function, defined over a *parametric space* (domain), taking values in the *NURBS space* (codomain). Therefore, if N is the dimension of the *parametric space* and M is the dimension of the *NURBS space*, a NURBS entity is defined as $\mathbf{H} : \mathbb{R}^N \to \mathbb{R}^M$. For example, one scalar parameter ($N = 1$) can describe both a plane curve ($M = 2$) and a 3D curve ($M = 3$). In the case of a surface, two scalar parameters are needed ($N = 2$) together with, of course, three physical coordinates ($M = 3$). The mathematical formula of a generic NURBS hyper-surface is

$$\mathbf{H}\left(u^{(1)}, ..., u^{(N)}\right) = \sum_{i_1=0}^{n_1} \cdots \sum_{i_N=0}^{n_N} R_{i_1,...,i_N}\left(u^{(1)}, ..., u^{(N)}\right) \mathbf{P}_{i_1,...,i_N}, \tag{3}$$

where $R_{i_1,...,i_N}\left(u^{(1)}, ..., u^{(N)}\right)$ are the piecewise rational basis functions, which are related to the standard Bernstein polynomials $N_{i_k,p_k}(u^{(k)})$, $k = 1, ..., N$ by means of the relationship

$$R_{i_1,...,i_N}\left(u^{(1)}, ..., u^{(N)}\right) = \frac{\omega_{i_1,...,i_N} \prod_{k=1}^{N} N_{i_k,p_k}\left(u^{(k)}\right)}{\sum_{j_1=0}^{n_1} \cdots \sum_{j_N=0}^{n_N} \left[\omega_{j_1,...,j_N} \prod_{k=1}^{N} N_{j_k,p_k}\left(u^{(k)}\right)\right]}. \tag{4}$$

In Eqs. (3) and (4), $\mathbf{H}\left(u^{(1)}, ..., u^{(N)}\right)$ is a M-dimension vector-valued rational function, $\left(u^{(1)}, ..., u^{(N)}\right)$ are scalar dimensionless parameters defined in the interval $[0, 1]$, whilst $\mathbf{P}_{i_1,...,i_N}$ are the so called *control points* (CPs). The j-th CP coordinate $\left(P^{(j)}_{i_1,...,i_N}\right)$ is stored in the array $\mathbf{P}^{(j)}$, whose dimension is $(n_1 + 1) \times \cdots \times (n_N + 1)$. The explicit expression of CPs coordinates in $\mathbb{R}^M$ is:

$$\begin{aligned} &\mathbf{P}_{i_1,...,i_N} = \left\{P^{(1)}_{i_1,...,i_N}, ..., P^{(M)}_{i_1,...,i_N}\right\}, \; P^{(j)}_{i_1,...,i_N} \in \mathbb{R}, \\ &i_k = 0, \ldots, n_k, \; k = 1, \ldots, N, \; j = 1, ..., M. \end{aligned} \tag{5}$$

The CPs layout is referred as *control polygon* for curves, *control net* for surfaces and *control hyper-net* for hyper-surfaces [28]. The generic CP does not actually belong to the NURBS entity but it affects its shape by means of its coordinates. A suitable scalar quantity $\omega_{i_1,...,i_N}$ (called weight) is related to the respective CP, i.e. $\mathbf{P}_{i_1,...,i_N}$. $N_{i_k,p_k}(u^{(k)})$ are the so-called *blending functions*. For each parametric direction $u^{(k)}$, $k = 1, ..., N$, a specific degree p_k is assigned. The recursive definition of the blending function $N_{i_k,p_k}(u^{(k)})$ is

$$\begin{aligned} N_{i_k,0}\left(u^{(k)}\right) &= \begin{cases} 1 \text{ if } U^{(k)}_{i_k} \leq u^{(k)} < U^{(k)}_{i_k+1}, \\ 0 \text{ otherwise,} \end{cases} \\ N_{i_k,q}\left(u^{(k)}\right) &= \frac{u^{(k)} - U^{(k)}_{i_k}}{U^{(k)}_{i_k+q} - U^{(k)}_{i_k}} N_{i_k,q-1}\left(u^{(k)}\right) + \frac{U^{(k)}_{i_k+q+1} - u^{(k)}}{U^{(k)}_{i_k+q+1} - U^{(k)}_{i_k+1}} N_{i_k+1,q-1}\left(u^{(k)}\right), \\ &q = 1, ..., p_k. \end{aligned} \tag{6}$$

where each blending function is defined on the knot vector

$$\mathbf{U}^{(k)} = \left\{ \underbrace{0, \cdots, 0}_{p_k+1}, U^{(k)}_{p_k+1}, \cdots, U^{(k)}_{m_k-p_k-1}, \underbrace{1, \cdots, 1}_{p_k+1} \right\}, \tag{7}$$

whose dimension is $m_k + 1$, with

$$m_k = n_k + p_k + 1. \tag{8}$$

Each knot vector $\mathbf{U}^{(k)}$ is a non-decreasing sequence of real numbers that can be interpreted as a discrete collection of values of the related dimensionless parameter $u^{(k)}$. The NURBS blending functions are characterised by several interesting properties: the interested reader is addressed to [28] for a deeper insight into the matter. Here, only the *local support property* is recalled because it is of paramount importance for the metamodelling strategy based on NURBS hyper-surfaces:

$$\begin{aligned} &R_{i_1,\ldots,i_N}\left(u^{(1)}, ..., u^{(N)}\right) \neq 0, \\ &\text{if } \left(u^{(1)}, ..., u^{(N)}\right) \in \left[U^{(1)}_{i_1}, U^{(1)}_{i_1+p_1+1}\right[\times \cdots \times \left[U^{(N)}_{i_N}, U^{(N)}_{i_N+p_N+1}\right[. \end{aligned} \tag{9}$$

Eq. (9) means that each CP (and the respective weight) affects only a precise zone of the *parametric space*, that is referred as *local support* or *influence zone*.

3. A metamodel based on NURBS hyper-surfaces

In this paper, an original approach is used to set up all the parameters tuning the shape of the NURBS hyper-surface representing the metamodel. This method aims at providing the function $\hat{\mathbf{z}}(\mathbf{x})$, approximating the real transfer function $\mathbf{z}(\mathbf{x})$ of a MIMO system, as follows:

$$\begin{array}{rccc} & E = \left[x^{(1)}_{\min}, x^{(1)}_{\max}\right] \times \ldots \times \left[x^{(N)}_{\min}, x^{(N)}_{\max}\right] & \rightarrow & \mathbb{R}^M \\ \hat{\mathbf{z}}: & \mathbf{x} & \rightarrow & \mathbf{H}\left(\mathbf{f}\left(\mathbf{x}\right)\right). \end{array} \tag{10}$$

where $\mathbf{x} = \left(x^{(1)}, \ldots, x^{(N)}\right)$ is the vector collecting the N inputs of the system, $\mathbf{H}\left(\mathbf{f}\left(\mathbf{x}\right)\right) = \left(X^{(1)}(\mathbf{x}), \ldots, X^{(M)}(\mathbf{x})\right)$ is the vector containing the M approximated outputs and $\mathbf{f}\left(\mathbf{x}\right)$ is a bijective function realising the mapping of the space E into the parametric domain $[0,1]^N$ of the NURBS hyper-surface, i.e.

$$\begin{array}{rccc} & E & \rightarrow & [0,1]^N \\ \mathbf{f}: & \mathbf{x} & \rightarrow & \mathbf{f}\left(\mathbf{X}\right) = \left(f^{(1)}\left(X^{(1)}\right), \ldots, f^{(N)}\left(X^{(N)}\right)\right) = \mathbf{u}, \end{array} \tag{11}$$

where $\mathbf{u} = \left(u^{(1)}, \ldots, u^{(N)}\right)$ are the parametric coordinates of the metamodel inputs $\mathbf{x}$ and $f^{(k)}\left(x^{(k)}\right)$ is a bijective function. Generally, the function $f^{(k)}\left(x^{(k)}\right)$ can be defined as,

$$\begin{array}{rccc} & \left[x^{(k)}_{\min}, x^{(k)}_{\max}\right] & \rightarrow & [0,1] \\ f^{(k)}: & x^{(k)} & \rightarrow & f^{(k)}\left(x^{(k)}\right) = \dfrac{x^{(k)} - x^{(k)}_{\min}}{x^{(k)}_{\max} - x^{(k)}_{\min}} = u^{(k)}, \quad k = 1, ..., N. \end{array} \tag{12}$$

The surrogate model is formulated without introducing neither simplifying hypotheses

nor empirical rules to set *all the parameters* affecting the shape of the NURBS hyper-surface. The resulting metamodelling strategy is thus problem-independent and constitutes the main advantage of the proposed approach.

Of course, this ambitious goal can be achieved through a pertinent formulation of the metamodel generation problem. Such a problem is formulated as a constrained hyper-surface fitting problem and the surrogate model based on NURBS hyper-surfaces is smartly coupled to the ERASMUS algorithm [29, 30] able of determining both the number and the value of the parameters involved in the definition of the NURBS hyper-surface.

3.1. Design variables

Eqs. (3) and (4) show that the parameters involved in the definition of a NURBS hyper-surface are of different nature:

- *integer variables*, like the number of both knot vector components and CPs, ($m_k + 1, n_k + 1$, respectively) as well as the degrees of the blending functions p_k along each dimension;
- *continuous variables*, like the internal knot vector values $U^{(k)}_{p_k+1}, ..., U^{(k)}_{m_k-p_k-1}$, CPs coordinates $\mathbf{P}_{i_1,...,i_N}$, weights $\omega_{i_1,...,i_N}$ and the dimensionless parameters $u^{(k)}_s$ at which the NURBS hyper-surface is evaluated.

Some of these parameters are interdependent, whereas other can be smartly chosen. In particular, as far as the dimensionless parameters are concerned, i.e. $u^{(k)}_s$, they are obtained according to the mapping $\mathbf{f}$ of Eq. (12) applied to the inputs of the MIMO for each TP, i.e.

$$u^{(k)}_s = \frac{x^{(k)}_s - x^{(k)}_{\min}}{x^{(k)}_{\max} - x^{(k)}_{\min}}, \quad k = 1, \dots, N, \quad s = 1 \dots, n_{\text{TP}}, \tag{13}$$

where n_{TP} is the overall number of TPs. Therefore, the NURBS dimensionless parameters do not belong to the set of design variables. Moreover, the number of CPs along each parametric direction can be determined once the size of the knot vector and the degree of the blending functions along the same direction are known, according to Eq. (8). Accordingly, the number of CPs is excluded from the design variables vector.

The determination of the optimum value of CPs coordinates can be carried out by a dedicated algorithm. In particular, when the size of the knot vector and its internal components, along each direction, are known, the degree of the blending functions (along each direction) is given and the values of $u^{(k)}_s$ have been computed by means of Eq. (13), finding the optimum value of the CPs coordinates is a quite trivial task. Indeed, the NURBS hyper-surface fitting problem is convex in terms of CPs coordinates. The interested reader can refer to [28, 30, 31] for more information on the iterative algorithms to obtain the CPs coordinates.

Finally, the independent parameters tuning the NURBS hyper-surface shape are:

- the N degrees p_k;
- the N knot lengths $m_k + 1$;
- the $m_k - 2p_k - 1$ internal components of the knot vector $\mathbf{U}^{(k)}$, $k = 1, ..., N$;
- the $n_{\text{CP}} = \prod_{k=1}^{N} (n_k + 1)$ weights $\omega_{i_1,...,i_N}$.

These parameters are collected in the vector of design variables $\boldsymbol{\xi}$ as

$$\begin{aligned}\boldsymbol{\xi}^{\mathrm{T}} &= \Big(p_1,\ldots,p_N,m_1,\ldots,m_N,U^{(1)}_{p_1+1},\ldots,U^{(1)}_{m_1-p_1-1},\ldots,\\ &\quad U^{(N)}_{p_N+1},\ldots,U^{(N)}_{m_N-p_N-1},\omega_{0,\ldots,0},\ldots,\omega_{n_1,\ldots,n_N}\Big)\,.\end{aligned} \tag{14}$$

It is noteworthy that the number of independent parameters defining the NURBS hyper-surface is given by the following equation

$$n_{\mathrm{var}} = 2N + \sum_{k=1}^{N}\left(m_k - 2p_k - 1\right) + \prod_{k=1}^{N}\left(n_k+1\right), \tag{15}$$

which depends upon the integer variables of the hyper-surface.

3.2. Objective function and optimisation constraints

The problem of generating a suitable metamodel can be formulated as a constrained hyper-surface fitting problem. The goal is to minimise the Euclidean distance (in M dimensions) between the NURBS hyper-surface and the set of TPs. Nevertheless, the main idea is to search for the best value of the parameters tuning the shape of the NURBS hyper-surface minimising the overall number of CPs and blending functions degrees by keeping a sufficient accuracy (according to the problem requirements). Of course, the number of CPs is related to the number of data needed to evaluate the outputs of the metamodel, whereas the blending function degrees are related to the processing time of the metamodel (recall that the NURBS blending functions are recursively evaluated according to Eq. (6)). Accordingly, three objective functions Φ_i have been compared in this study, i.e.

$$\Phi_1(\boldsymbol{\xi}) = \left(\sum_{s=1}^{n_{\mathrm{TP}}}\left(\mathbf{z}\left(\mathbf{x}_s\right)-\mathbf{H}\left(\mathbf{u}_s\right)\right)^2\right)^{\left(\sum_{k=0}^{N}\frac{n_k}{n_{k_{\max}}}+\sum_{k=0}^{N}\frac{p_k}{p_{k_{\max}}}\right)}, \tag{16}$$

$$\Phi_2(\boldsymbol{\xi}) = a\frac{\sum_{s=1}^{n_{\mathrm{TP}}}\left(\mathbf{z}\left(\mathbf{x}_s\right)-\mathbf{H}\left(\mathbf{u}_s\right)\right)^2}{\varepsilon_0} + b\sum_{k=0}^{N}\frac{n_k}{n_{k_{\max}}} + (1-a-b)\sum_{k=0}^{N}\frac{p_k}{p_{k_{\max}}}, \tag{17}$$

$$\Phi_3(\boldsymbol{\xi}) = a\sum_{k=0}^{N}\frac{n_k}{n_{k_{\max}}} + (1-a)\sum_{k=0}^{N}\frac{p_k}{p_{k_{\max}}}, \tag{18}$$

where $\mathbf{z}\left(\mathbf{x}_s\right)$ is the vector collecting the outputs of the s-th TP for the input vector $\mathbf{x}_s$ while $\mathbf{H}\left(\mathbf{u}_s\right)$ is the corresponding point belonging to the NURBS hyper-surface, evaluated at the parametric coordinates $\mathbf{u}_s$. In Eqs. (17) and (18) a and b are suitable weighting coefficients balancing the requirements on the euclidean distance between the hyper-surface and the set of TPs, on the number of CPs and on the NURBS hyper-surface processing time (via the degrees). In Eq. (17), the parameter ε_0 is a reference value of the Euclidean distance between the set of TPs and the NURBS hyper-surface in order to get a dimensionless term.

It is noteworthy that when the objective function takes the form of Eq. (18) an additional requirement on the model accuracy must be added in terms of a constraint on the

maximum relative error on the j-th output, i.e. $\varepsilon_{\max}^{(j)}$, as:

$$\varepsilon_{\max}^{(j)}\left(\boldsymbol{\xi}\right) \leq \varepsilon_{\text{th}}^{(j)},\ j=1,\ldots,M, \tag{19}$$

where $\varepsilon_{\text{th}}^{(j)}$ is a threshold related to the required accuracy for the j-th output of the surrogate model, whilst $\varepsilon_{\max}^{(j)}\left(\boldsymbol{\xi}\right)$ is the maximum relative error defined as

$$\varepsilon_{\max}^{(j)}\left(\boldsymbol{\xi}\right) = \max_{\mathbf{u}_s}\left(\varepsilon^{(j)}\left(\mathbf{u}_s\right)\right), \tag{20}$$

where $\varepsilon^{(j)}\left(\mathbf{u}_s\right)$ is the relative error at parametric coordinates $\mathbf{u}_s = \left(u_s^{(1)},...,u_s^{(N)}\right)$ given by

$$\varepsilon^{(j)}\left(\mathbf{u}_s\right) = \frac{\left|H^{(j)}\left(\mathbf{u}_s\right) - z^{(j)}\left(\mathbf{x}_s\right)\right|}{z_{\max}^{(j)} - z_{\min}^{(j)}}, \quad j=1,\ldots,M, \quad s=1,...,n_{\text{TP}}. \tag{21}$$

In Eq. (21), $z^{(j)}\left(\mathbf{x}_s\right)$ is the j-th output evaluated as the s-th TP at inputs $\mathbf{x}_s = \left(x_s^{(1)},\ldots,x_s^{(N)}\right)$, while $H^{(j)}\left(\mathbf{u}_s\right)$ is the counterpart evaluated on the NURBS hyper-surface at the parametric coordinates $\mathbf{u}_s = \left(u_s^{(1)},\ldots,u_s^{(N)}\right)$. The scalar quantities $z_{\max}^{(j)}$ and $z_{\min}^{(j)}$ are the maximum and minimum values of the j-th output over the set of TPs, respectively.

The hyper-surface fitting problem can be stated in the form of a CNLPP as follows

$$\min_{\boldsymbol{\xi}} \Phi_\tau(\boldsymbol{\xi}), \quad \tau = 1,\ 2 \text{ or } 3$$

$$\text{subject to :}$$

$$\begin{cases} \varepsilon_{\max}^{(j)}\left(\boldsymbol{\xi}\right) \leq \varepsilon_{\text{th}}^{(j)},\ j=1,\ldots,M, \quad \text{(to be considered only if } \tau=3), \\ 0 \leq U_{l_k}^{(k)} \leq 1,\ l_k = p_k+1,\ldots,m_k-p_k+1,\ k=1,\ldots,N, \\ U_{l_k}^{(k)} \leq U_{l_k+1}^{(k)},\ l_k = p_k+1,\ldots,m_k-p_k+1,\ k=1,\ldots,N, \\ n_{\text{CP}} \leq n_{\text{TP}}, \\ n_k - p_k \geq 0,\ k=1,\ldots,N, \\ \omega_{i_1,\ldots,i_N} \geq 0,\ i_k = 0,\ldots,n_k. \end{cases} \tag{22}$$

Problem (22) is a non-standard CNLPP for different reasons. Firstly, unlike the methods available in the literature [16, 28], the proposed strategy aims at providing all the design variables $\boldsymbol{\xi}$ without introducing neither simplifying hypotheses nor empirical rules to set the parameters involved in the NURBS hyper-surface definition. Secondly, the number of design variables is integrated into the design variables vector $\boldsymbol{\xi}$ and depends upon the integer parameters of the NURBS hyper-surface. As explained in [29], problem (22) can be smartly stated as an optimisation problem of modular systems belonging to different families. Generally speaking, a *modular system* is composed by *elementary units*, i.e. the *modules*. Each module is characterised by the same vector of unknowns, i.e. the design variables of the module, which can take, a priori, different values for every module (in the most general case of different modules). For problem (22), two different classes of modules can be identified: the knot vector components and the weights. CNLPPs dealing with modular systems are unconventional because they are defined over a domain of variable dimension, which depends upon a linear combination of the integer variables characterising

the modular system. In particular, for problem (22) the problem dimension is given by Eq. (15).

4. Numerical strategy for metamodel generation

Considering the peculiar nature of problem (22), a hybrid optimisation tool developed at I2M laboratory in Bordeaux and called HERO (Hybrid EvolutionaRy Optimisation) has been used. It is composed of the genetic algorithm ERASMUS (EvolutionaRy Algorithm for optimiSation of ModUlar Systems) [29], interfaced with the *active-set* algorithm of MATLAB *fmincon* family, available in the MATLAB *optimization toolbox* [36].

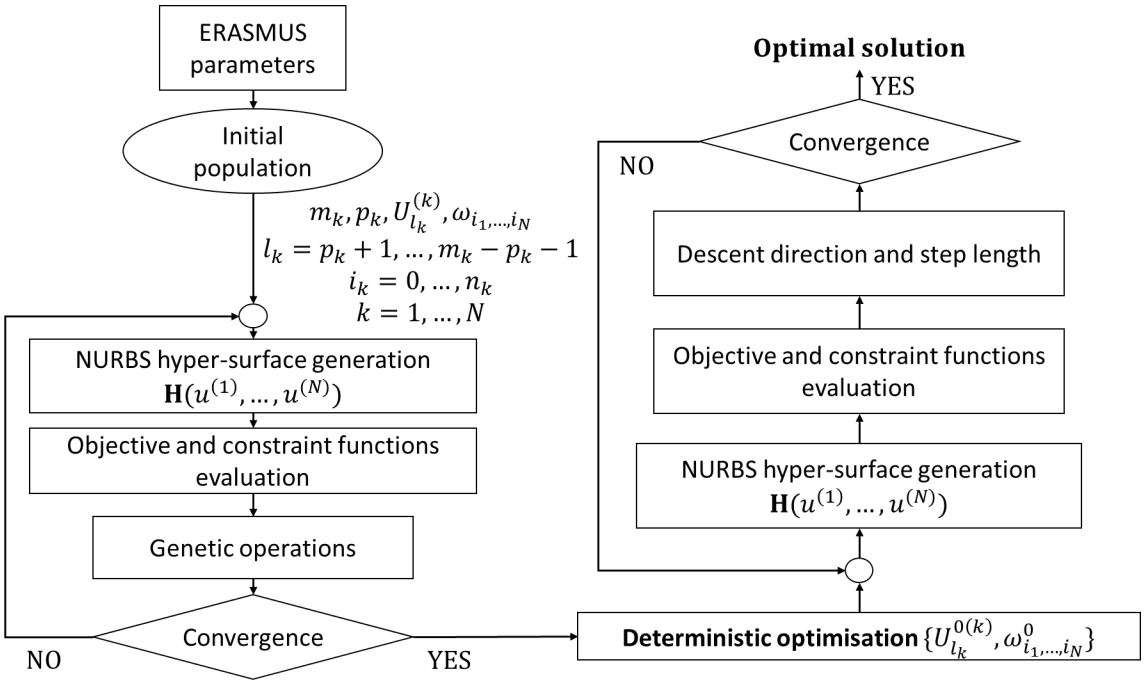

Figure 1: Hybrid EvolutionaRy Optimisation (HERO) algorithm.

As shown in Fig. 1, the optimisation procedure for problem (22) is split in two phases. During the first step, solely the GA ERASMUS is used to perform the solution search and the full set of design variables is taken into account. ERASMUS is a special GA able to deal with optimisation problems characterised by a variable number of design variables, and more specifically, optimisation problems of *modular systems*. This goal can be achieved thanks to the original representation of information in ERASMUS. In particular, the individual's *genotype* is organised in *modular parts*, each one composed of *chromosomes* which are in turn made of *genes* (each gene codes a specific design variable).

In agreement with the paradigm of natural sciences, individuals characterised by a different number of chromosomes (i.e. modular structures composed of a different number of modules) belong to different *species*. ERASMUS has been conceived for crossing also different species, thus making possible (and without distinction) the *simultaneous optimisation of species and individuals*. This task can be attained thanks to dedicated genetic operators, which have been implemented to perform the reproduction phase between individuals belonging to different species: the general architecture of ERASMUS is illustrated in Fig. 2. Therefore, ERASMUS is able to simultaneously optimise both the number of modules (for each class of modules) and the values of the design variables characteris-

ing each module. The effectiveness of ERASMUS has been proven on a large number of real-world engineering problems [20–22, 32, 37–41].

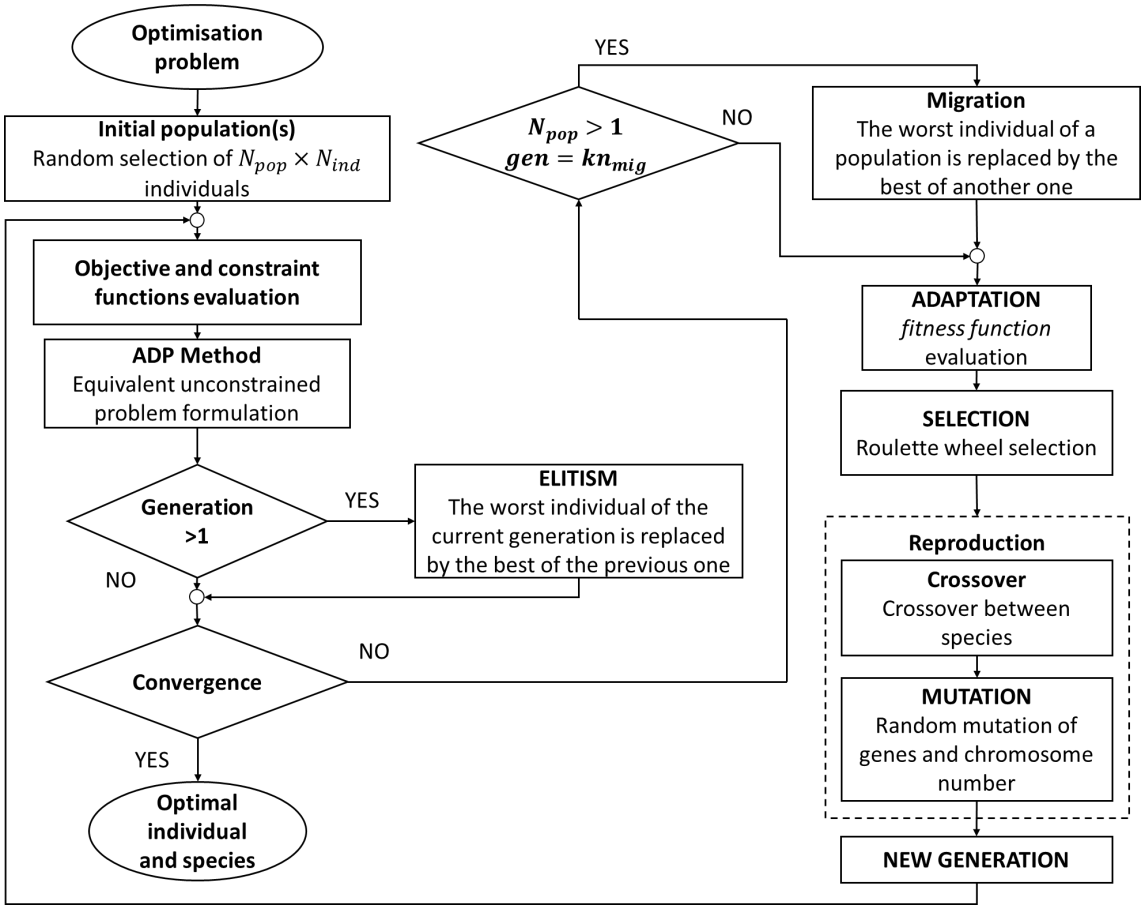

Figure 2: ERASMUS algorithm [29].

Due to the strong non-linearity of problem (22), the aim of the genetic calculation is to provide a potential sub-optimal point in the design space, which constitutes the initial guess for the subsequent phase, i.e. the local optimisation performed via the *active-set* algorithm of *fmincon* MATLAB function. During this second phase, only the components of the knot vector along each dimension and the weights are considered as design variables, see Fig. 1. The second phase of HERO allows finding a local minimum starting from the pseudo-optimal solution resulting from the first GA exploration of the design space.

4.1. Meta-heuristic exploration: first step

When the number of internal knots and the degree along each parametric direction are included among the design variables, problem (22) is defined over a space of variable dimension. Moreover, this CNLPP is characterised by a large number of design variables. In particular, when all the weights are included into the design variables vector, the computational cost could become prohibitive. To this purpose, a dedicated strategy, able to determine whether weights should be integrated or not in the optimisation process, has been developed. This task is achieved by splitting the exploration of the design space into two steps. In the first step, all weights are set to one, i.e. only B-Spline hyper-surfaces are used to fit the set of TPs. Then, if the error threshold is no satisfied, the *local support* properties of NURBS hyper-surfaces is used to assess which weights must be integrated

as design variables for the second step of the exploration. If the approximation error is satisfied after the first step, all weights are kept equal to one and the B-Spline hyper-surface obtained by ERASMUS is used as a starting point for the subsequent deterministic optimisation.

As a result, the CNLPP is stated as:

$$\min_{\boldsymbol{\xi}_\mathrm{I}} \Phi_\tau(\boldsymbol{\xi}_\mathrm{I}), \quad \tau = 1,\ 2 \text{ or } 3,$$

$$\text{subject to :}$$

$$\begin{cases} \varepsilon_{\max}^{(j)}(\boldsymbol{\xi}_\mathrm{I}) \leq \varepsilon_{\mathrm{th}}^{(j)},\ j = 1, \ldots, M, \quad \text{(to be considered only if } \tau = 3), \\ 0 < U_{l_k}^{(k)} < 1,\ l_k = p_k + 1, ..., m_k - p_k + 1,\ k = 1, ..., N, \\ U_{l_k}^{(k)} \leq U_{l_k+1}^{(k)},\ l_k = p_k + 1, ..., m_k - p_k,\ k = 1, ..., N, \\ n_{\mathrm{CP}} \leq n_{\mathrm{TP}}, \\ n_k - p_k \geq 0,\ k = 1, ..., N, \\ \omega_{i_1,...,i_N} = 1,\ i_k = 0, ..., n_k,\ k = 1, ..., N. \end{cases} \tag{23}$$

The vector $\boldsymbol{\xi}_\mathrm{I}$ collects the optimisation variables as follows,

$$\boldsymbol{\xi}_\mathrm{I}^\mathrm{T} = \left(p_1, \ldots, p_N, m_1, \ldots, m_N, U_{p_1+1}^{(1)}, \ldots, U_{m_1-p_1-1}^{(1)}, \ldots, U_{p_N+1}^{(N)}, \ldots, U_{m_N-p_N-1}^{(N)}\right). \tag{24}$$

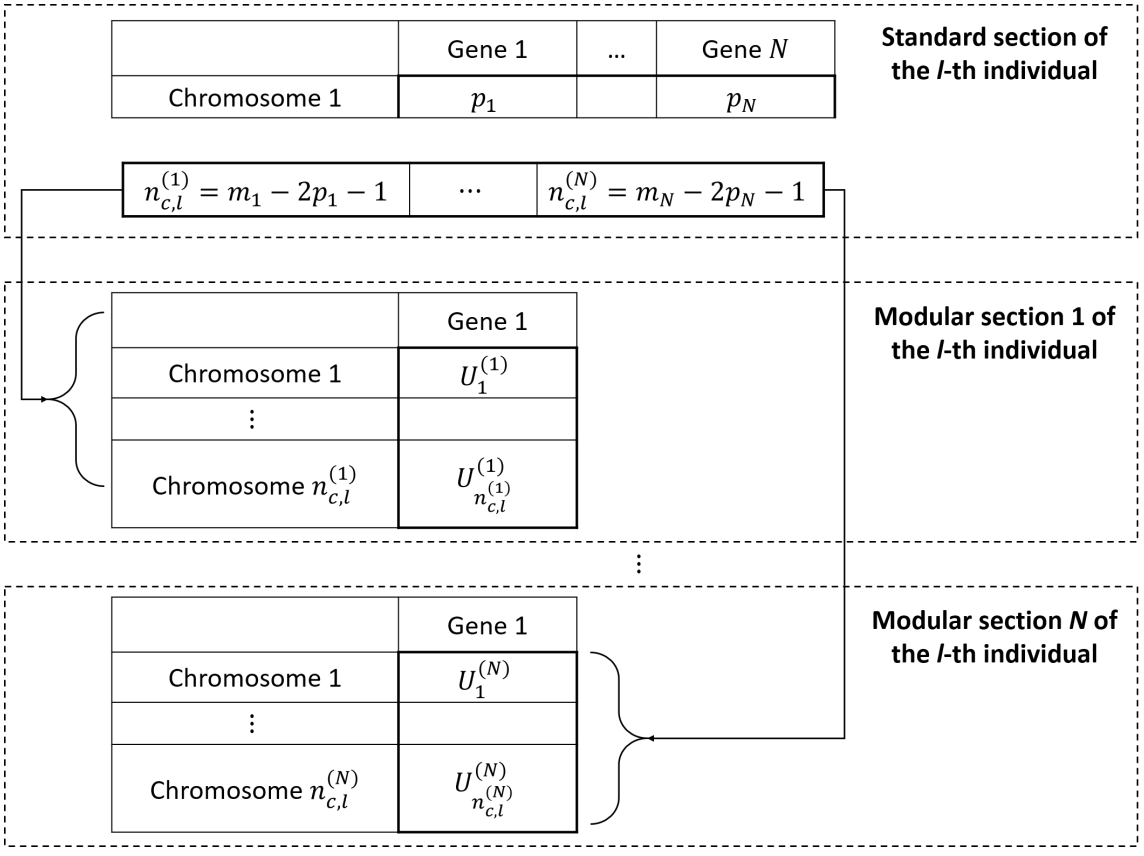

Figure 3: Individual's genotype of ERASMUS for the NURBS hyper-surface fitting problem.

It is noteworthy that a B-Spline hyper-surface can be considered as a modular system where m_k and p_k are constant design variables, whilst each knot vector $\mathbf{U}^{(k)}$ represents the generic module whose variables are $U_{l_k}^{(k)}$, $l_k = p_k + 1, ..., m_k - p_k - 1$, $k = 1, ..., N$. This system is, thus, composed of N modules corresponding to the N knot vectors of the B-Spline hyper-surface. The individual's genotype of ERASMUS for problem (23) is given in Fig. 3, where $n_{c,l}^{(k)}$ is the number of chromosomes of the k-th modular part (related to the knot vector $\mathbf{U}^{(k)}$) of the l-th individual. This quantity corresponds to the number of internal components of the knot vector $\mathbf{U}^{(k)}$ and is related to m_k and p_k by the following relation:

$$n_{c,l}^{(k)} = m_k - 2p_k - 1. \tag{25}$$

This step aims at determining the number of optimisation variables (i.e. the number of parameters tuning the shape of the NURBS hyper-surface). Thus, the degrees and the knot vector lengths given by ERASMUS remain constant for the rest of the optimisation process. Moreover, by means of the user-defined error threshold $\varepsilon_{\text{th}}^{(j)}$ and the *local support* property of NURBS hyper-surfaces, the following set $\mathbf{\Omega}$ can be defined

$$\mathbf{\Omega} = \left\{ \omega_{i_1,...,i_N},\ i_k = 0, ..., n_k \quad | \quad \exists \mathbf{u}_s \in \mathbb{U} \cap \mathbb{S}^{(i_1,...,i_N)} \right\}, \tag{26}$$

where $\mathbb{S}^{(i_1,...,i_N)}$ is the local support of control point $\mathbf{P}_{i_1,...,i_N}$, i.e.

$$\mathbb{S}^{(i_1,...,i_N)} = \left[U_{i_1}^{(1)}, U_{i_1+p_1+1}^{(1)} \right[\times ... \times \left[U_{i_N}^{(N)}, U_{i_N+p_N+1}^{(N)} \right[, \tag{27}$$

while $\mathbb{U}$ is the set of TPs at which the maximum relative error does not meet the user-defined threshold

$$\mathbb{U} = \left\{ \mathbf{u}_s,\ s = 1, ..., n_{\text{TP}} \quad | \quad \exists j \in [1, M] : \varepsilon^{(j)}(\mathbf{u}_s) > \varepsilon_{\text{th}}^{(j)} \right\}. \tag{28}$$

If $\mathbf{\Omega}$ is not empty, the weights belonging to $\mathbf{\Omega}$ are introduced in the optimisation process for the second step of the exploration. Moreover, if $\mathbf{\Omega}$ is empty, the second step of the meta-heuristic exploration does not occur since no optimisation variables are added to the optimisation process.

4.2. Meta-heuristic exploration: second step

During the second step of the meta-heuristic exploration, the number of optimisation variables is set because the number of knot vector components and the degree along each direction result from the previous step. Therefore, only continuous design variables are considered at this stage, namely the weights belonging to the set $\mathbf{\Omega}$ as well as the knot vectors components. They are grouped as follows:

$$\boldsymbol{\xi}_{\Pi}^{\text{T}} = \left(U_{p_1+1}^{(1)}, \ldots, U_{m_1-p_1-1}^{(1)}, \ldots, U_{p_N+1}^{(N)}, \ldots, U_{m_N-p_N-1}^{(N)}, \mathbf{\Omega} \right). \tag{29}$$

This step takes place only if the set $\mathbf{\Omega}$ is not empty and aims at providing a potential sub-optimal point constituting the initial guess for the subsequent phase, i.e. the local optimisation performed via the deterministic algorithm. When $\mathbf{\Omega}$ is empty, the solution provided by ERASMUS at the end of the first step is used as a starting point for the

deterministic algorithm. For this second step, the CNLPP is formulated as follows:

$$\min_{\boldsymbol{\xi}_{\mathrm{II}}} \psi(\boldsymbol{\xi}_{\mathrm{II}}) = \frac{1}{M}\sum_{j=1}^{M} \frac{\left[\sum_{s=1}^{n_{\mathrm{TP}}} \left(H^{(j)}_{\boldsymbol{\xi}_{\mathrm{II}}}(\mathbf{u}_s) - Q^{(j)}_s\right)^{\chi}\right]^{\left(\frac{1}{\chi}\right)}}{\left[\sum_{s=1}^{n_{\mathrm{TP}}} \left(H^{(j)}_{\mathrm{I}}(\mathbf{u}_s) - Q^{(j)}_s\right)^{\chi}\right]^{\left(\frac{1}{\chi}\right)}},$$

$$\text{subject to :}$$

$$\begin{cases} 0 < U^{(k)}_{l_k} < 1,\ l_k = p_k + 1, ..., m_k - p_k + 1,\ k = 1, ..., N, \\ U^{(k)}_{l_k} \leq U^{(k)}_{l_k+1},\ l_k = p_k + 1, ..., m_k - p_k,\ k = 1, ..., N, \\ \omega_{i_1,...,i_N} \geq 0,\ i_k = 0, ..., n_k,\ k = 1, ..., N. \end{cases} \tag{30}$$

In Eq. (30), ψ is the χ-norm function, which is used to approximate the maximum relative error of approximation and gives good results for $\chi \geq 20$, while $\mathbf{H}_{\mathrm{I}}$ is the hyper-surface resulting from the first step of the procedure. Obviously, the hyper-surface $\mathbf{H}_{\mathrm{I}}$, corresponding to the optimisation variables $\boldsymbol{\xi}^{(\mathrm{opt})}_{\mathrm{I}}$ found at the end of the first step, is introduced in the initial population of this second step to boost convergence.

4.3. Deterministic optimisation: third step

If the set $\boldsymbol{\Omega}$ is not empty, the pseudo-optimal solution found at the end of the second step $\boldsymbol{\xi}^{\mathrm{opt}}_{\mathrm{II}}$ is used as initial guess for the deterministic optimisation phase. On the contrary, if $\boldsymbol{\Omega}$ is empty, the pseudo-optimal solution used as initial guess for the deterministic optimisation phase is the one provided by the first step, $\boldsymbol{\xi}^{\mathrm{opt}}_{\mathrm{I}}$. The CNLPP formulation is the same as that provided in Eq. (30). In this case the design variables vector is indicated as $\boldsymbol{\xi}_{\mathrm{III}}$, while the quantity $\mathbf{H}_{\mathrm{II}}$ is used to get a dimensionless objective function. $\mathbf{H}_{\mathrm{II}}$ is the NURBS hyper-surface associated to the vector of optimisation variables $\boldsymbol{\xi}^{(\mathrm{opt})}_{\mathrm{II}}$ given by ERASMUS at the end of the second step. Note that the vector $\boldsymbol{\xi}_{\mathrm{III}}$ collects design variables as $\boldsymbol{\xi}_{\mathrm{II}}$ does according to Eq. (29). Of course, this step only aims at reaching the nearest local optimum from the pseudo-optimal solution provided by the previous step (i.e. first step if $\boldsymbol{\Omega}$ is empty and second step otherwise).

5. The application: least-weight design of a composite stiffened panel

The optimisation problem considered in this work has been taken from [32] and focuses on the repetitive unit (RU) of a composite stiffened panel typical of aircraft wings. The RU is composed of a skin and a stringer with an Omega-shaped cross-section, as illustrated in Fig. 4. The skin and the stringer are both made of orthotropic unidirectional carbon/epoxy laminæ, whose properties are listed in Table 1 (the values have been taken from [32]).

The main hypotheses at the basis of the macroscopic mechanical response of the RU focus essentially on the laminate behaviour and geometry (for both skin and stringer): (a) each laminate is made of identical plies (i.e. same thickness t_{ply} and material); (b) the material of the constitutive layer has a linear elastic isotropic behaviour; (c) each laminate is quasi-homogeneous and fully orthotropic [32, 42]; (d) at the macroscopic scale the elastic response of each laminate is described in the theoretical framework of the First-order Shear Deformation Theory (FSDT) and its stiffness matrices are expressed in terms of polar parameters (PPs); (e) no delamination occurs at the plies interface for both skin and stringer.

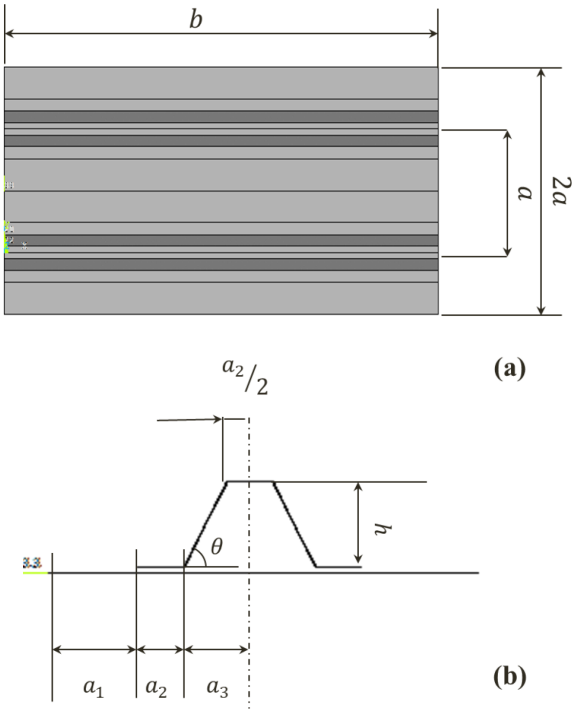

Figure 4: (a) Geometry and dimensions of the stiffened panel (only two RUs) and (b) geometrical parameters of the RU (taken from [32])

It is noteworthy that only the first-level problem (FLP) of the multi-scale two-level (MS2L) optimisation strategy, presented in [32], is considered in this study.

In particular, the FLP aims at determining the optimal value of both mechanical and geometric parameters of the laminates composing the RU of the panel. At this stage, each laminate composing the RU is modelled as an equivalent homogeneous anisotropic plate [43, 44], whose behaviour is described in terms of PPs. The goal of the FLP is the minimisation of the RU mass subject to requirements on the laminate feasibility and on the first buckling load of the RU (more details are available in [32]). The design variables of the FLP can be of geometric or mechanical nature.

In this study, only the skin and the stringer overall thickness are taken as geometric design variables, whilst the other geometric parameters are set equal to those of the reference solution listed in Table 2. Nevertheless, the overall thickness of each laminate composing the RU is a multiple of the thickness of the elementary layer, t_{ply} (see Table 1), i.e.

$$t_\alpha = n_\alpha t_{\text{ply}}, \quad \alpha = S, B, \tag{31}$$

where n_S and n_B are the number of layers of skin and stiffener, respectively. For optimisation purposes, it is very useful to consider dimensionless design variables. Accordingly, variables n_S and n_B are grouped in the vector of dimensionless geometric design variables $\boldsymbol{\eta}_g$ as follows:

$$\boldsymbol{\eta}_g^{\mathrm{T}} = (n_S, n_B)\,. \tag{32}$$

As discussed in [32], in the framework of FSDT [45] the constitutive law of the laminate

Technical constants		Polar parameters of $\mathbf{Q}$ [a]		Polar parameters of $\hat{\mathbf{Q}}$ [b]	
E_1 [MPa]	161000.0	T_0 [MPa]	23793.3868	T [MPa]	5095.4545
E_2 [MPa]	9000.0	T_1 [MPa]	21917.8249	R [MPa]	1004.5454
G_{12} [MPa]	6100.0	R_0 [MPa]	17693.3868	Φ [deg]	90.0
ν_{12} [MPa]	0.26	R_1 [MPa]	19072.0711		
ν_{23} [MPa]	0.10	Φ_0 [deg]	0.0		
		Φ_1 [deg]	0.0		
Density and thickness					
ρ [kg/mm^3]	1.58×10^{-6}				
t_{pli} [mm]	0.125				

[a] In-plane reduced stiffness matrix of the ply.
[b] Out-of-plane reduced shear stiffness matrix of the ply.

Table 1: Material properties of the carbon/epoxy ply taken from [32]

a [mm]	150.00	
b [mm]	600.00	
a_2 [mm]	15.00	
a_3 [mm]	21.50	
h [mm]	30.00	
M_{ref} [kg]	0.92	
λ_{ref} [N]	466 830	
Stacking sequence	Part	Number of plies
$[(45/-45/90_2)_2\,/\,(45/-45)_3]_s$	skin(S)	28
$[45_2/0_2/-45_2/90_4/-45_2/0_2/45_2]_s$	stringer (B)	32

Table 2: Reference solution for the stiffened panel design problem

(expressed within its global frame $R = \{O; x, y, z\}$) can be stated as:

$$\begin{Bmatrix} \mathbf{N} \\ \mathbf{M} \\ \mathbf{F} \end{Bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{O} \\ \mathbf{B} & \mathbf{D} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \widehat{\mathbf{A}} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\varepsilon}_0 \\ \boldsymbol{\chi}_0 \\ \boldsymbol{\gamma}_0 \end{Bmatrix} , \tag{33}$$

where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{D}$ are the membrane, membrane/bending coupling and bending stiffness matrices of the laminate, while $\widehat{\mathbf{A}}$ is the out-of-plane shear stiffness matrix. $\mathbf{N}$, $\mathbf{M}$ and $\mathbf{F}$ are the vectors of membrane forces, bending moments and shear forces per unit length, respectively, whilst $\boldsymbol{\varepsilon}_0$, $\boldsymbol{\chi}_0$ and $\boldsymbol{\gamma}_0$ are the vectors of in-plane strains, curvatures and out-of-plane shear strains of the laminate middle plane, respectively (Voigt's notation has been employed [45]).

In order to analyse the elastic response of the multilayer structure, it is very useful to introduce the laminate normalised stiffness matrices [32]:

$$\mathbf{A}^* = \frac{1}{t}\mathbf{A}, \quad \mathbf{B}^* = \frac{2}{t^2}\mathbf{B}, \quad \mathbf{D}^* = \frac{12}{t^3}\mathbf{D}, \quad \widehat{\mathbf{A}}^* = \frac{1}{t}\widehat{\mathbf{A}}, \tag{34}$$

where t is the total thickness of the laminate. In this study a fully orthotropic, *quasi-homogeneous* laminate is considered. As discussed in [42–44], quasi-homogeneity property

can be obtained by imposing the following conditions:

$$\mathbf{B}^* = \mathbf{O}, \quad \mathbf{C}^* = \mathbf{A}^* - \mathbf{D}^* = \mathbf{O}. \tag{35}$$

In Eq. (35), $\mathbf{C}^*$ is the so-called homogeneity matrix [43, 44]. According to the above formula, a quasi-homogeneous laminate is an uncoupled laminate showing the same behaviour in terms of membrane and bending normalised stiffness matrices. Moreover the fully orthotropic behaviour (both in terms of membrane and bending stiffness matrices) can be obtained by imposing the following condition:

$$\Phi_0^{A^*} - \Phi_1^{A^*} = K^{A^*} \frac{\pi}{4}, \text{ with } K^{A^*} = 0, 1, \tag{36}$$

where $\Phi_0^{A^*}$ and $\Phi_1^{A^*}$ are the polar angles of the membrane stiffness matrix, whilst K^{A^*} is a parameter tuning the orthotropy shape [42–44]. Inasmuch as the laminate is quasi-homogeneous, Eq. (36) holds for the bending stiffness matrix too. For more details on all the possible laminate elastic symmetries, the reader is addressed to [42–44].
It can be proven that, in the FSDT framework, for a fully orthotropic, quasi-homogeneous laminate the overall number of independent mechanical design variables describing its mechanical response reduces to only three, i.e. the anisotropic polar parameters $R_{0K}^{A^*}$ and $R_1^{A^*}$ and the polar angle $\Phi_1^{A^*}$ (this last represents the orientation of the main orthotropy axis) of matrix $\mathbf{A}^*$. In this study the polar angle of matrix $\mathbf{A}^*$ is set equal to $\Phi_1^{A^*} = 0$, which means that the main orthotropy axis is aligned to the RU axis. More details can be found in [43, 44, 46].

As for the geometric variables, it is useful to introduce dimensionless PPs, i.e.

$$\rho_0 = \frac{R_{0K}^{A^*}}{R_0}, \quad \rho_1 = \frac{R_1^{A^*}}{R_1}. \tag{37}$$

As discussed in [32], further constraints, defining the laminate feasibility domain, have to be considered on the laminate PPs, which arise from the combination of the layers orientations and positions within the stack. These constraints read:

$$\begin{cases} -1 \leq \rho_0 \leq 1, \\ 0 \leq \rho_1 \leq 1, \\ 2\left(\rho_1\right)^2 - 1 - \rho_0 \leq 0. \end{cases} \tag{38}$$

Of course, the dimensionless PPs as well as the related feasibility constraints have to be considered for each laminate constituting the panel RU (i.e. skin and stringer). Therefore, the vector $\boldsymbol{\eta}_m$ collects the dimensionless PPs for skin and stiffener as follows:

$$\boldsymbol{\eta}_m^{\mathrm{T}} = \left(\rho_{0S}, \rho_{1S}, \rho_{0B}, \rho_{1B}\right). \tag{39}$$

The design variables (both geometrical and mechanical) are collected in the following vector:

$$\boldsymbol{\eta}^{\mathrm{T}} = \left(\boldsymbol{\eta}_g^{\mathrm{T}}, \boldsymbol{\eta}_m^{\mathrm{T}}\right). \tag{40}$$

Finally, the optimisation problem can be formulated as a classical CNLPP:

$$\begin{cases} \min_{\boldsymbol{\eta}} \dfrac{M(\boldsymbol{\eta})}{M_{\text{ref}}}, \\ \text{subject to:} \\ 1.05 - \dfrac{\lambda(\boldsymbol{\eta})}{\lambda_{\text{ref}}} \leq 0, \\ 2(\rho_{1S})^2 - 1 - \rho_{0S} \leq 0, \\ 2(\rho_{1B})^2 - 1 - \rho_{0B} \leq 0, \end{cases} \tag{41}$$

In Eq. (41) M is the overall mass of the RU, λ is the first buckling load of the stiffened panel, whilst M_{ref} and λ_{ref} are the counterparts for the reference solution. The design space of the first-level problem, together with the type of each design variable, is detailed in Table 3. This problem has been solved in [32] thanks to the GA ERASMUS [29] coupled with a finite element (FE) model of the RU implemented in the commercial software ANSYS [47]. In this study, the mechanical response given by the commercial code (i.e. the first buckling load of the RU computed by means of an eigenvalue buckling analysis) has been replaced by the metamodel based on NURBS hyper-surfaces generated through the metamodelling strategy described above.

Design variable	Type	Lower bound	Upper bound	Step
ρ_{0S}	continuous	-1.0	1.0	-
ρ_{1S}	continuous	0.0	1.0	-
ρ_{0B}	continuous	-1.0	1.0	-
ρ_{1B}	continuous	0.0	1.0	-
n_S	integer	20	32	1
n_B	integer	20	32	1

Table 3: Design space for problem (41).

6. Numerical results: metamodel generation

6.1. The finite element model

The design problem of Eq. (41) have been solved by Montemurro *et al.* in [32] by coupling the GA ERASMUS with the commercial FE code ANSYS. However, the computational cost of the optimisation process to find a solution for the FLP is quite high. The main idea is to replace the FE model by a metamodel based on NURBS hyper-surfaces, generated by means of the HERO strategy. The constrained hyper-surface fitting problems of Eqs. (16)-(18) have been solved by generating a database of TPs obtained as a result of the eigenvalue buckling analysis.

In particular, the metamodel aims at providing an approximation $\hat{\lambda}$ of the first buckling load of the panel RU λ as a function of the design variables of the FLP:

$$\hat{\lambda} \approx \lambda(\boldsymbol{\xi}). \tag{42}$$

The FE model of the panel RU is implemented in the FE commercial code and is shown in Fig. 5. The FE model performs an eigenvalue analysis and is made of SHELL281 elements

(8-nodes element) and *multipoint constraint elements* MPC184, both having 6 degrees of freedom (DOFs) per node. The behaviour of SHELL281 elements is set by directly defining the normalised stiffness matrices $\mathbf{A}^*$, $\mathbf{B}^*$, $\mathbf{D}^*$ and $\mathbf{H}^*$, while MPC184 elements are used to ensure the compatibility of generalised displacements between skin and stringer. In addition, MPC184 elements have been used to apply boundary conditions (BCs) on two *pilot nodes*, i.e. $A = \{0, 0, z_g\}$ and $B = \{b, 0, z_g\}$, located at the ends of the RU. More details on the FE model can be found in [32].

The BCs for nodes A and B are:

$$\begin{aligned} &\text{node } A\text{: } u_i = 0, \quad \beta_i = 0, \quad i = x, y, z, \\ &\text{node } B\text{: } F_x = -1\text{N}, \quad u_y = u_z = 0, \quad \beta_i = 0, \quad i = x, y, z, \end{aligned} \tag{43}$$

where u_i and β_i are nodal displacements and rotations, respectively, while F_x is the x component of the nodal force. It is noteworthy that the first-buckling load of the stiffened panel is calculated by considering periodic boundary conditions (PBCs) on its RU. This fact implicitly implies the hypothesis of a panel having an "infinite" length along y-axis, according to the frame illustrated in Fig. (5). PBCs read:

$$\begin{aligned} &u_i\left(x, -\frac{a}{2}, 0\right) - u_i\left(x, \frac{a}{2}, 0\right) = 0, \quad \forall x \in]0, b[\,, \quad i = x, y, z, \\ &\beta_i\left(x, -\frac{a}{2}, 0\right) - \beta_i\left(x, \frac{a}{2}, 0\right) = 0, \quad \forall x \in]0, b[\,, \quad i = x, y, z. \end{aligned} \tag{44}$$

PBCs of Eq. (44) are defined for each couple of nodes belonging to the skin lateral edges (i.e nodes located at $y = \pm\frac{a}{2}$) execpt those placed on the lines at $x = 0$ and $x = b$, these last being respectively linked to pilot nodes A and B. PBCs are defined via ANSYS constraint equations (CEs) [47] between homologous nodes of the skin lateral edges. A sensitivity analysis (not reported here for sake of brevity) on the proposed FE model with respect to mesh size has been conducted [32] showing that a mesh having 56959 DOFs is sufficient to properly evaluate the first buckling load of the stiffened panel.

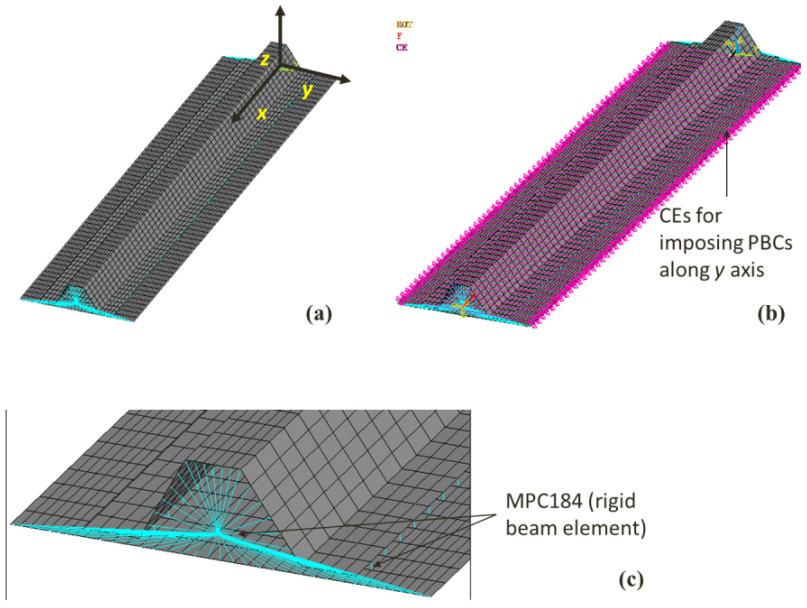

Figure 5: (a) FE model of the panel RU, (b) constraint equations for the PBC imposed on y axis and (c) the detail of the MPC184 elements

The TPs database is generated by assessing the first buckling load λ for different values of the design variables n_S, n_B, ρ_{0S}, ρ_{1S}, ρ_{0B} and ρ_{1B}, as listed in Table 4. The database is stored in the form of a 6-D array $\mathbf{Q}$:

$$Q_{s_1,s_2,s_3,s_4,s_5,s_6} = \lambda(n_S^{s_1}, n_B^{s_2}, \rho_{0S}^{s_3}, \rho_{1S}^{s_4}, \rho_{0B}^{s_5}, \rho_{1B}^{s_6}), \quad s_k = 0, ..., r_k, \quad k = 1, \cdots, 6. \tag{45}$$

TPs database	n_S	n_B	ρ_{0S}	ρ_{1S}	ρ_{0B}	ρ_{1B}	Overall n. of TPs
N. of points $(r_k + 1)$	7	7	7	7	7	7	117 649

Table 4: TPs database used to generate and validate the metamodel.

It is noteworthy that the metamodel based on NURBS hyper-surfaces can provide a value of the first buckling load λ also for non-integer values of the design variables n_S and n_B. However, since continuous values of these variables are meaningless only integer values of those parameters are given as inputs to the metamodel. In addition, inasmuch as feasibility constraints of Eq. (38) on the laminate dimensionless PPs are available in a closed form, only feasible combinations of variables $\rho_{0\alpha}$ and $\rho_{1\alpha}$ ($\alpha = S, B$) have been considered to generate the TPs database.

6.2. Generation of the metamodel: results

The metamodel based on NURBS hyper-surfaces has been generated as a results of CNLPPs of Eqs. (23)-(30). The parameters tuning the behaviour of the ERASMUS code and of the active-set algorithm are provided in Tables 5 and 6, respectively.

Genetic parameters		
	Metamodel generation (Problem (23))	Stiffened panel optimisation (Problem (41))
N. of populations	3	3
Population size	100	200
N. of generations	100	400
Crossover probability	0.85	0.85
Mutation probability	0.005	0.005
Shift probability	0.5	0.5
Isolation time	5	5
Selection	roulette-wheel	roulette-wheel
Elitism	active	active

Table 5: Genetic parameters of the GA ERASMUS for metamodelling problems (23) and (41).

It is noteworthy that, for each considered objective function formulation, at the end of the first optimisation step the set $\mathbf{\Omega}$ is empty. Therefore, the second step of the optimisation process does not take place and a B-Spline hyper-surface is sufficient to describe the trend of the first buckling load in terms of the considered design variables.

The results of the HERO strategy for the three objective function formulations of Eqs. (16), (17) and (18), are presented in Table 7. Firstly, it can be seen that the degrees are relatively low regardless of the objective function form. Indeed, the greater the degree along the generic direction, the lower the number of design variables because the number

fmincon parameters	
Algorithm	Active Set
Maximum number of objective function evaluations	$100 \times n_{var}$
Maximum number of iterations	400
Minimum objective function improvement	10^{-6}
Minimum input variables change	10^{-6}
Minimum gradient norm of the Lagrange's function	10^{-6}
Constraint violation threshold	10^{-6}

Table 6: Parameters of the *fmincon* function [36] for problem (30).

NURBS hyper-surface parameters resulting from HERO				
Objective function		Φ_1 Eq. (16)	Φ_2 Eq. (17)	Φ_3 Eq. (18)
Auxiliary coefficients		/	$a = \frac{1}{3}$; $b = \frac{1}{3}$	$a = 0.5$
$\mathbf{p} = (p_1, ..., p_6)$		$(2, 2, 1, 2, 2, 2)$	$(1, 1, 2, 1, 1, 1)$	$(2, 1, 2, 1, 2, 2)$
$\mathbf{n} = (n_1, ..., n_6)$		$(5, 6, 6, 6, 6, 6)$	$(3, 1, 3, 2, 3, 3)$	$(4, 5, 6, 6, 6, 6)$
Overall CPs number		100 842	1 536	72 030
$\varepsilon_{\max}$	(ERASMUS)	0.0038	0.0713	0.0087
	(*fmincon*)	0.0036	0.0653	0.0080
$\varepsilon_{\text{mean}}$	(ERASMUS)	$2.2336e^{-5}$	0.0099	$8.9231e^{-5}$
	(*fmincon*)	$2.3744e^{-5}$	0.0071	$8.8004e^{-5}$

Table 7: Results comparison of HERO metamodelling strategy for the objective function formulations of Eqs. (16), (17) and (18).

of internal knots along this direction decreases. As a result, decreasing degrees gives a better objective function value.

Table 7 highlights the fact that the objective function choice has a real impact on the metamodel parameter values. The three formulation are characterised by advantages and drawbacks.

The metamodel obtained considering Eq. (16) is the easiest to use since no additional parameter has to be set in the objective function. It gives good results in terms of the relative errors (maximum and average ones), but the obtained metamodel is the one with the highest number of CPs.

Conversely, the objective function of Eq. (17) gives the user the possibility of balancing between approximation error, number of data to evaluate the metamodel (i.e. number of CPs) and on-line computation time, which is related to the degrees of the blending functions. However, when equal importance is given to these requirements, the obtained metamodel is the less accurate for the problem at hand. Indeed, the main drawback is that one cannot know *a priori* the value of the auxiliary parameters a and b appearing in Eq. (17).

Finally, the metamodel obtained with the objective function (18) appears to be the best compromise. The obtained metamodel shows a good balance between accuracy and number of CPs. In this case, the CNLPP during the first step of the optimisation process is subject to the constraint on the maximum relative error of Eq. (19). Indeed, the optimisation tool tries to minimise the resources of the metamodel (both number of CPs and degrees values) respecting a given accuracy (i.e. the constraint on the maximum approximation error).

These three metamodels have been used to optimise the panel RU and the results are discussed in the next Section.

7. Numerical results: stiffened panel optimisation

The optimisation of the RU of the composite stiffened panel has been carried out by means of the GA ERASMUS coupled with the ANSYS FE model and with the three metamodels generated and discussed above. The parameters of the GA ERASMUS to solve problem (41) are listed in Table 5, while the results are given in Table 8. It is noteworthy that the value of the first buckling load reported in Table 8, for each metamodel, is calculated by entering the optimum value of the design variables obtained at the end of the optimisation process into the FE model of the panel RU. The comparison between the first buckling load of the optimised configuration resulting from each metamodel and that provided by the FE model (for the same value of the design variables) is given in Table 9.

Parameters	Skin			Stringer			Mass [kg]	λ [N]
	n_S	ρ_{0S}	ρ_{1S}	n_B	ρ_{0B}	ρ_{1B}		
Reference	28	-	-	32	-	-	0.9194	466 830
Solution 1 (metamodel Φ_1)	32	-0.0255	0.0010	23	0.5809	0.6323	0.8719 (-5.17%)	493 509 (+5.71%)
Solution 2 (metamodel Φ_2)	32	0.0712	0.0069	22	0.6973	0.6560	0.8587 (-6.60%)	476 510 (+2.02%)
Solution 3 (metamodel Φ_3)	32	0.4175	0.0161	23	0.6434	0.6788	0.8719 (-5.17%)	490 560 (+5.08%)
Solution 4 (FE model)	32	-0.0349	0.0020	23	0.7602	0.8636	0.8719 (-5.17%)	498 332 (+6.75%)

Table 8: Optimised configurations of the panel RU obtained by ERASMUS coupled to the three metamodels and the FE model (the percentage difference between performances of the optimised solutions and the reference one is indicated in parentheses).

When the metamodel is interfaced with the ERASMUS algorithm, the whole optimisation process requires approximatively 2 minutes instead of two weeks when ERASMUS is directly coupled with the ANSYS FE model (for parameters defined in Table 5). However, this time does not include the TPs database generation. When the time required to generate the TPs database is taken into account, the use of the metamodel based on NURBS hyper-surfaces allows to save about two days to obtain an optimised solution.

As it can be seen from Table 8, the optimised solution of the panel RU has improved performances, in terms of both mass and buckling strength when compared to the reference solution, regardless of the mathematical formulation used to generate the metamodel. Indeed, weight saving is almost 5.20% for all solutions, except for the metamodel using objective function Φ_2 of Eq. (17), which is characterised by a weight saving of about 6.6%. However, as it can been inferred from Table 9, this metamodel is characterised by an accuracy lower than the others in terms of first buckling load assessment. In particular, although at the end of the optimisation process the constraint on the buckling load of Eq. (41) is met, the real value of the buckling load provided by the FE model for this configuration is lower than that resulting from the metamodel (but still higher than the reference counterpart, see Tables 2 and 8).

It is noteworthy that the optimised configuration resulting from the metamodel where the objective function Φ_1 is considered (solution 1 in Table 8) is very close to the optimised

Solution	First-buckling load λ [N]	
Solution 1	Metamodel	494 210 (+5.87%)
	ANSYS FE	493 509 (+5.71%)
Solution 2	Metamodel	491 230 (+5.23%)
	ANSYS FE	476 510 (+2.07%)
Solution 3	Metamodel	491 390 (+5.26%)
	ANSYS FE	490 560 (+5.08%)

Table 9: Comparison of of the first buckling load resulting from the three metamodels and the original FE model for solutions listed in Table 8. The percentage difference between the buckling load of the optimised solutions and the reference one is indicated in parentheses.

solution found by directly coupling the FE model of the RU to the ERASMUS algorithm (solution 4 in Table 8). This is a quite expected result because, as stated in Section 6, the metamodel related to the objective function of Eq. (16) is the one characterised by the higher accuracy. Indeed, solutions 1 and 4 shows the same kind of orthotropy in terms of the dimensionless PPs characterising both stringer and skin macroscopic behaviour. Conversely, the optimised configuration provided by the metamodel generated by using the objective function of Eq. (18) (solution 3 in Table 8) is characterised by equivalent properties (in terms of mass and first buckling load) when compared to solutions 1 and 4. Nevertheless, the elastic behaviour of the skin is completely different because the optimum value of ρ_{0S} is positive, which means that the orthotropy type is completely different with respect to the counterpart characterising solutions 1 and 4. This result confirms the non-convex nature of the CNLPP of Eq. (41).

8. Conclusions

An original metamodelling technique based on Non-Uniform Rational Basis Spline (NURBS) hyper-surfaces has been presented in this paper. The generation of a suitable metamodel is stated as a constrained non-linear programming problem (CNLPP). More specifically, the metamodel is obtained as a result of a constrained hyper-surface fitting problem: the goal is to find the optimal values of the parameters tuning the shape of the NURBS hyper-surface in order to fit a given set of target points (TPs). In this background, the metamodel constitutive parameters are determined by means of an original hybrid optimisation strategy, which combines the features of an enhanced meta-heuristic algorithm and of a deterministic one to perform the solution search.

The proposed approach is very general: neither simplifying hypotheses nor empirical rules are used to select *a priori* the parameters governing the behaviour of the metamodel. Indeed, the optimal number of parameters and their optimal values are simultaneously determined by the hybrid optimisation strategy (and according to user's defined accuracy), making the proposed approach problem-independent.
In particular, a special genetic algorithm, called ERASMUS (EvolutionaRy Algorithm for optimiSation of ModUlar Systems), has been used to find the optimal value of both discrete and continuous variables involved in the definition of the NURBS hyper-surface. However, when the NURBS entity discrete parameters are included in the vector of design variables, the resulting CNLPP is defined over a domain of variable dimension. By means of its peculiar features (i.e. genetic operators allowing for the simultaneous evolution of species and individuals) the ERASMUS algorithm is able to deal with this special class of CNLPP. Moreover, the proposed approach aims at being as efficient as possible because it is able

of automatically determining if a B-Spline or a NURBS hyper-surfaces is needed to fit a given set of TPs.

The effectiveness of the metamodel based on NURBS hyper-surfaces is tested in the framework of a meaningful design problem: the least-weight design of a composite stiffened panel subject to constraints of different nature, i.e. on the first buckling load and feasibility requirements.
Three different formulations of the CNLPP have been proposed to generate the metamodel based on NURBS hyper-surfaces. The first one is completely automatic and does not need the user intervention. The other two formulations allows the user to define the relative importance between accuracy, amount of information (related to the number of control points) and processing time (related to the basis function degrees). Results show that the first formulation outperforms the other two in terms of accuracy, whilst the third one represents the best compromise between accuracy and computational effort. Moreover, the second formulation represents the best solution in terms of memory saving (the resulting metamodel is the one characterised by the least amount of information) but the relative accuracy is worse than that related to the other formulations.

This behaviour is confirmed by the results obtained when the metamodels are used for design purposes. The optimised configurations show performances better than those characterising the reference solution (in terms of both mass and first buckling load). As expected, the metamodel generated considering the second formulation is characterised by the lowest accuracy (which remains still acceptable for design purposes). The time required to found a solution is about two minutes (regardless of the considered formulation) which is significantly lower than that required by directly coupling the finite element model with the ERASMUS algorithm (about two weeks to get a solution). However, if the time required to generate the TPs database is included in the processing time of the metamodel, the time saving reduces to only two days.

As far as prospects of this study are concerned, the proposed metamodelling strategy lacks of an efficient sampling technique. As a matter of fact, if not properly set, the sampling strategy may result in a set of TPs not suited to generate a pertinent metamodel for the problem at hand. A good sampling strategy should generate a TPs database made of the least amount of information, sufficient to ensure a good level of accuracy, and should provide information in the most critical regions of the metamodel domain. These aspects constitute a true challenge and research is ongoing in order to develop efficient sampling methods which take into account the topology of the set of TPs, i.e. presence of disjoint regions, non-convex domains, etc.

Data availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

References

[1] F. Chinesta, R. Keunings, A. Leygue, The Proper Generalized Decomposition for Advanced Numerical Simulations, SpringerBriefs in Applied Sciences and Technology, Springer International Publishing, Cham, 2014.
URL http://link.springer.com/10.1007/978-3-319-02865-1

[2] G. G. Wang, S. Shan, Review of Metamodeling Techniques in Support of Engineering Design Optimization, Journal of Mechanical Design 129 (4) (2007) 370–380.

URL http://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?articleid=1449318

[3] C. Bisagni, L. Lanzi, Post-buckling optimisation of composite stiffened panels using neural networks, Composite Structures 58 (2) (2002) 237–247.
URL http://www.sciencedirect.com/science/article/pii/S0263822302000533

[4] D. G. Krige, A statistical approach to some basic mine valuation problems on the Witwatersrand, Journal of the Southern African Institute of Mining and Metallurgy 52 (6) (1951) 119–139.
URL https://journals.co.za/content/saimm/52/6/AJA0038223X_4792

[5] M. D. Buhmann, Radial basis functions, Acta Numerica 2000 9 (2000) 1–38.
URL http://journals.cambridge.org/abstract_S0962492900000015

[6] E. Viennet, Réseaux à fonctions de base radiales, Apprentissage connexionniste (2006) 105.
URL https://hal.archives-ouvertes.fr/hal-00085092/

[7] B. YEGNANARAYANA, ARTIFICIAL NEURAL NETWORKS, PHI Learning Pvt. Ltd., 2009.

[8] V. Capecchi, M. Buscema, P. Contucci, B. D'Amore (Eds.), Applications of Mathematics in Models, Artificial Neural Networks and Arts, Springer Netherlands, Dordrecht, 2010.
URL http://link.springer.com/10.1007/978-90-481-8581-8

[9] G. Berkooz, P. Holmes, J. L. Lumley, The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows, Annual Review of Fluid Mechanics 25 (1) (1993) 539–575.
URL https://doi.org/10.1146/annurev.fl.25.010193.002543

[10] A. Chatterjee, An introduction to the proper orthogonal decomposition, Current Science 78 (7) (2000) 808–817.
URL http://www.jstor.org/stable/24103957

[11] Y. Zhao, W. Lu, C. Xiao, A Kriging surrogate model coupled in simulation–optimization approach for identifying release history of groundwater sources, Journal of Contaminant Hydrology 185-186 (2016) 51–60.
URL http://www.sciencedirect.com/science/article/pii/S0169772216300043

[12] A. A. Giunta, Aircraft Multidisciplinary Design Optimization using Design of Experiments Theory and Response Surface Modeling Methods (1997).
URL https://vtechworks.lib.vt.edu/handle/10919/30613

[13] M. J. Sasena, Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations (2002) 237.

[14] P. Chandrashekarappa, R. Duvigneau, Radial Basis Functions and Kriging Metamodels for Aerodynamic Optimization, report, INRIA (2006).
URL https://hal.inria.fr/inria-00137602/document

[15] H. Shi, Y. Gao, X. Wang, Optimization of injection molding process parameters using integrated artificial neural network model and expected improvement function method, The International Journal of Advanced Manufacturing Technology 48 (9-12) (2010) 955–962.
URL https://link.springer.com/article/10.1007/s00170-009-2346-7

[16] C. J. Turner, HyPerModels: hyperdimensional performance models for engineering design, PhD Thesis (2005).
URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.392.1513&rep=rep1&type=pdf

[17] C. J. Turner, R. H. Crawford, N-Dimensional Nonuniform Rational B-Splines for Metamodeling, Journal of Computing and Information Science in Engineering 9 (3) (2009) 031002.
URL http://ComputingEngineering.asmedigitalcollection.asme.org/article.aspx?articleid=1401568

[18] J. Steuben, J. Michopoulos, A. Iliopoulos, C. Turner, Inverse characterization of composite materials via surrogate modeling, Composite Structures 132 (2015) 694–708.
URL http://www.sciencedirect.com/science/article/pii/S0263822315003943

[19] 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.
URL https://link.springer.com/article/10.1007/s10999-017-9396-z

[20] 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 (2018) 225–251.
URL https://link.springer.com/article/10.1007/s10957-017-1192-2

[21] M. Montemurro, A. Catapano, A New Paradigm for the Optimum Design of Variable Angle Tow Laminates, in: A. Frediani, B. Mohammadi, O. Pironneau, V. Cipolla (Eds.), Variational Analysis and Aerospace Engineering: Mathematical Challenges for the Aerospace of the Future, Springer Optimization and Its Applications, Springer International Publishing, Cham, 2016, pp. 375–400.
URL https://doi.org/10.1007/978-3-319-45680-5_14

[22] 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.
URL http://www.sciencedirect.com/science/article/pii/S0263822316320712

[23] M. Montemurro, A. Catapano, A general B-Spline surfaces theoretical framework for optimisation of variable angle-tow laminates, Composite Structures 209 (2018) 561–578.
URL http://www.sciencedirect.com/science/article/pii/S0263822318324334

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

[25] G. Costa, M. Montemurro, J. Pailhès, N. Perry, Maximum length scale requirement in a topology optimisation method based on NURBS hyper-surfaces 68 (1) (2019) 153–156.
URL https://hal.archives-ouvertes.fr/hal-02277399

[26] 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.
URL http://www.sciencedirect.com/science/article/pii/S0045782519302968

[27] T. Rodriguez, M. Montemurro, P. Le Texier, J. Pailhès, Structural displacement requirement in a topology optimization algorithm based on isogeometric entities, Journal of Optimization Theory and Applications 184 (1) (2020) 250–276.
URL https://doi.org/10.1007/s10957-019-01622-8

[28] L. Piegl, W. Tiller, The NURBS Book, Monograph in Visual Communication, Springer-Verlag Berlin Heidelberg, 1995.
URL 10.1007/978-3-642-97385-7

[29] M. Montemurro, A contribution to the development of design strategies for the optimisation of lightweight structures, HDR thesis, Université de Bordeaux, France (2018).
URL http://hdl.handle.net/10985/15155

[30] Y. Audoux, M. Montemurro, J. Pailhès, Non-Uniform Rational Basis Spline hypersurfaces for metamodelling, Computer Methods in Applied Mechanics and Engineering 364 (2020 (in press)).
URL https://doi.org/10.1016/j.cma.2020.112918

[31] Y. Audoux, M. Montemurro, J. Pailhes, A surrogate model based on Non-Uniform Rational B-Splines hypersurfaces, Procedia CIRP 70 (2018) 463–468.
URL http://www.sciencedirect.com/science/article/pii/S2212827118304037

[32] M. Montemurro, A. Pagani, G. A. Fiordilino, J. Pailhès, E. Carrera, A general multi-scale two-level optimisation strategy for designing composite stiffened panels, Composite Structures 201 (2018) 968–979.
URL http://www.sciencedirect.com/science/article/pii/S0263822318311036

[33] P. Bézier, Courbes et surfaces, Hermès, Paris; Londres, 1986.

[34] C. de Boor, A Practical Guide to Splines., Mathematics of Computation 34 (149) (1980) 325.
URL https://www.jstor.org/stable/2006241?origin=crossref

[35] G. Farin, Curves and Surfaces for CAGD, Elsevier, 2002.
URL http://linkinghub.elsevier.com/retrieve/pii/B9781558607378X50005

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

[37] L. Cappelli, G. Balokas, M. Montemurro, F. Dau, L. Guillaumat, Multi-scale identification of the elastic properties variability for composite materials through a hybrid optimisation strategy, Composites Part B: Engineering 176 (2019) 107193.
URL http://www.sciencedirect.com/science/article/pii/S1359836819321766

[38] G. Bertolino, M. Montemurro, G. De Pasquale, Multi-scale shape optimisation of lattice structures: an evolutionary-based approach, International Journal on Interactive Design and Manufacturing 13 (4) (2019) 1565–1578.
URL https://doi.org/10.1007/s12008-019-00580-9

[39] M. Montemurro, M. I. Izzi, J. El-Yagoubi, D. Fanteria, Least-weight composite plates with unconventional stacking sequences: Design, analysis and experiments, Journal of Composite Materials 53 (16) (2019) 2209–2227.
URL https://doi.org/10.1177/0021998318824783

[40] E. Panettieri, M. Montemurro, A. Catapano, Blending constraints for composite laminates in polar parameters space, Composites Part B: Engineering 168 (2019) 448–457.
URL http://www.sciencedirect.com/science/article/pii/S1359836819304494

[41] L. Cappelli, M. Montemurro, F. Dau, L. Guillaumat, Characterisation of composite elastic properties by means of a multi-scale two-level inverse approach, Composite Structures 204 (2018) 767–777.
URL http://www.sciencedirect.com/science/article/pii/S0263822318318403

[42] M. Montemurro, The polar analysis of the Third-order Shear Deformation Theory of laminates, Composite Structures 131 (2015) 775–789.
URL http://www.sciencedirect.com/science/article/pii/S0263822315004791

[43] M. Montemurro, An extension of the polar method to the First-order Shear Deformation Theory of laminates, Composite Structures 127 (2015) 328–339.
URL http://www.sciencedirect.com/science/article/pii/S026382231500197X

[44] M. Montemurro, Corrigendum to "An extension of the polar method to the First-order Shear Deformation Theory of laminates" [Compos. Struct. 127 (2015) 328–339], Composite Structures 131 (2015) 1143–1144.
URL https://linkinghub.elsevier.com/retrieve/pii/S0263822315004651

[45] J. N. Reddy, Mechanics of Laminated Composite Plates and Shells: Theory and Analysis, Second Edition, CRC Press, 2004.

[46] P. Vannucci, Plane Anisotropy by the Polar Method*, Meccanica 40 (4) (2005) 437–454.
URL https://doi.org/10.1007/s11012-005-2132-z

[47] I. ANSYS, Mechanical APDL Modeling and Meshing Guide, 275 Technology Drive Canonsburg, PA 15317 (2013).