UNIONE EUROPEA
Fondo Europeo di Sviluppo Regionale

Ministero dell'Università e della Ricerca

PON RICERCA E INNOVAZIONE 2014 - 2020

Consiglio Nazionale delle Ricerche

Cluster Trasporti

CTN01_00176_163601

TRIM
Tecnologia e Ricerca Industriale per la Mobilità Marina

TRIM
Tecnologia e Ricerca Industriale per la Mobilità Marina

Efficient shape parametrisation techniques

Sotto-Progetto	Efficienza	
Obiettivo Realizzativo	Sviluppo di metodi numerici innovativi per la parametrizzazione Validazione e test case	
Descrizione attività	Sviluppo di metodi numerici innovativi per la parametrizzazione Validazione e test case	
Tipo di documento	Rapporto Tecnico	
Codice del documento	SP.4-OR.10-D.1	
Data di emissione	30/09/2021	
Redazione	Andrea Mola, Marco Tezzele, Nicola Demo, Gianluigi Rozza	SISSA Scuola Internazionale Superiore di Studi Avanzati

Consiglio Nazionale delle Ricerche · FINCANTIERI · CETENA · SEASTEMA · AZIMUT BENETTI GROUP · PERINI NAVI · mare FVG · Navtec · LIBERTYlines · opus automazione · NEXTWORKS ENGINEERING FORWARD · SULTAN

SISSA
Scuola Internazionale Superiore di Studi Avanzati

Titolo documento	Efficient shape parametrisation techniques
Codice documento	SP.4-OR.10-D.1
Distribuzione	Pubblico

Rev.	Data	Pagine	Redazione	Responsabile
0	30/09/2021	2+30	Andrea Mola, Marco Tezzele, Nicola Demo, Gianluigi Rozza	Gianluigi Rozza

L'attività descritta nella presente pubblicazione è stata finanziata dal Progetto TRIM — Tecnologia e Ricerca Industriale per la Mobilità Marina — coordinato dal Consiglio Nazionale delle Ricerche e finanziato dal Ministero dell'Università e della Ricerca nell'ambito dell'iniziativa dei Cluster Tecnologici Nazionali.

Contents

Summary

We describe different approaches for shape parametrisation techniques based on free form deformation, radial basis functions and inverse distance weighting, allowing efficient geometrical morphing. We show for all these techniques several aspects to highlight their potential in targeted applications.

Tests are performed in study cases related with naval and nautical applications (hulls, propellers, etc). A particular attention is dedicated to the integration of shape morphing and parametrisation with CFD solvers, as well as mesh generations. Further efforts is given to the description of open source software libraries developed within the project.

1 Introduction

The present work focuses for the most part on problems modeled by parameterized partial differential equations. In many cases of interest, the input parameters of such PDE problems happen to be associated with the shape of the computational domain, or of one of its portions. For instance, virtual prototyping applications very often investigate on the influence that the shape of a given craft or artifact have on its predicted performance. In some cases, through repetitive resolution of PDE systems, such applications go as far as identifying the optimal shape of the craft or artifact under study, or quantifying the sensitivity of its performance to variations of its geometry. In these last cases, the virtual prototyping application pipelines will include optimization or uncertainty quantification tools. Naturally, such mathematical algorithms can only be fed with numbers. Thus, the main purpose of shape parameterization is that of mapping any geometric variation of the object under study to variations of numbers — the parameters — which then can be processed by any other mathematical algorithm used in the virtual prototyping pipeline.

To allow for completely automated virtual prototyping and optimization pipelines, it is very important that the shape parametrization algorithm is able to automatically obtain the deformed domain and computational grid once the shape parameter values are prescribed. Such constraint not only ensures that repeated solution of parameterized PDE problems is possible also when shape parameters are included among the inputs, but it also significantly limits the time required for meshing, which in such a framework typically represents the most time consuming activity in terms of human operator hours.

A further constraint for shape parameterization algorithms arises from their application in the context of model order reduction. As discussed with greater detail in several chapters of the present work, many modal decomposition strategies used in ROMs are based on a snapshot matrix which is obtained under the assumption that the PDE problem degrees of freedom number and order remains unaltered as input parameter values are modified. For such a reason, the shape parameterization algorithm must not alter the topology of the computational grid used for the PDE resolution. Thus, a simple approach consisting in creating a new mesh after the shape of the domain has been altered is not suited for ROM applications. A more comprehensive strategy which deforms the computational mesh along with the domain geometry is needed.

For this reason, in the present work we first analyze algorithms that are suited for the deformation of codimension one subsets of $\mathbb{R}^2$ or $\mathbb{R}^3$, which are used to define the geometry of computational domain through the description of their boundary representation. In two dimensions, codimension one subsets are typically described through the points of piecewise linear curves representation or by the control points and nodes of Non Uniform Rational Basis-Splines (NURBS) curves included in Computer Aided Design (CAD) files. In three dimensions the most common representations of codimension one subsets are given by the points and connectivity of surface triangulations, or by the control points and nodes of NURBS saved into CAD files. After this, we focus on the deformation of codimension zero computational domains in $\mathbb{R}^2$ or $\mathbb{R}^3$, which are

typically used to define computational grids. Both in two and three dimensions, these domains are typically described through the nodes and the connectivity of polygonal or polyhedral tessellations. It is important to stress that an additional advantage of deforming both the domain geometry and the computational grid in a consistent way, is that it allows for a strong interaction between design and simulation teams. In fact, the shape modifications validated by PDE based simulations will readily be available for production with no further processing.

In the present work, we first consider parameterization algorithms which attempt to come up with a general law to deform at once the whole two or three dimensional space in which the morphed object is embedded. Such general purpose algorithms typically operate extending or interpolating the displacement of a relatively small set of control points, to all the points of the two or three dimensional space. Thus, the displacements of such control points become the user prescribed shape parameters. The position of each point of the objects of interest is then displaced through interpolation of the control points displacement to obtain the modified shape. This paradigm is common to Free Form Deformation (FFD), originally proposed in [1], and to the application of Radial Basis Functions (RBF) interpolation [2] and Inverse Distance Weighting (IDW) interpolation [3] to shape parameterization. A considerable advantage that these algorithms present, is that they can be indifferently operate on the points of domain geometries or on the nodes of computational grids. For this reason they have been easily applied to CAD geometries, as for instance in [4], and to computational meshes as in [5]. On the other hand, one of the possible drawbacks of these tools is that the shape parameters identified are not necessarily related to a specific physical behavior of the object under study. For instance, a simple scaling parameter can affect the aspect ratio of a ship hull and have a clear effect on its hydrodynamic performance. On the other hand, the effect of the prescribed displacement of one of many shape control points on the hull performance is more difficult to establish. Ideally, in several engineering applications the shape parameter should be readily associated to qualitative variations in the designed object performance. Moreover, in some cases general purpose algorithms might violate engineering constraints on the shapes to be deformed. This is for instance the case of aircraft and ship propellers, which are objects with a highly engineered shape and strict design rules [6]. Thus, we also discuss cases in which the shape parameterization strategy is tailored to certain bodies to alter their shape preserving the geometric characteristics they have been designed to possess.

The content of this work is organized as follows. In Section 2 we introduce and discuss the mathematical background of different algorithms for shape parameterization. After general purpose methods such as free form deformation (FFD), radial basis functions (RBF) interpolation, and inverse distance weighting (IDW) interpolation are presented, we describe examples of different object specific deformation strategies. In Section 3, we discuss the application of all these strategies to deformation of both domain geometries and computational meshes. In such context, we also present applications of the proposed shape parameterization methodologies to industrial and real life problems. Finally, in Section 4 we provide some concluding remarks, as well as some interesting perspectives for the field of shape parameterization in the next few years.

2 Shape parametrisation techniques

In this section we are going to present a diverse range of possible shape parameterization tools. We divided such deformation techniques into two main categories based on their objective: the general purpose shape parameterizations and the object specific ones.

2.1 General purpose shape parameterization algorithms

In the first category we can group such morphing techniques which come with a general law to deform the whole 3D space at once. They usually make use of a set of control points to move in order to propagate the desired shape deformation.

Let us introduce a physical domain $\Omega \subset \mathbf{R}^3$, and a general parametric deformation map $\mathcal{M}(\cdot, \mu) : \mathbf{R}^3 \to \mathbf{R}^3$ acting on Ω, where $\mu \in \mathcal{P}$ represents the parameters vector characterizing the morphing.

In the following subsections we are going to present the specific definition of $\mathcal{M}$ for three non-affine techniques such as free form deformation (FFD), radial basis functions (RBF) interpolation, and inverse distance weighting (IDW) interpolation. For affine maps used in the model order reduction context refer to [7], while for a more general overview we suggest [8]. For an efficient implementation of the methods presented below we recommend the open source Python package called PyGeM [9], which also integrates easily with industrial CAD files.

2.1.1 Free form deformation

Free form deformation is a well-known morphing technique for both local and global deformations of a given object. Originally presented in [1], in the last decades it has gained popularity in the context of parametric PDEs and model order reduction for shape optimization studies [10, 11, 12, 13, 14], optimal flow control problems [15], sailing boats [16], in naval engineering [4, 17, 18, 19, 20, 21] and CFD applications [22], and in isogeometric analysis [23, 24], to cite a few. Since the deformation induced by the FFD does not depend on the topology of the object to be morphed, it is very versatile and non-intrusive, especially for complex geometries or in industrial contexts [25, 26, 27].

FFD deforms the physical domain Ω by propagating the displacements of a lattice of points, called FFD control points, placed in Ω. The map $\mathcal{M}$ is the composition of three different geometric deformations, that is

$$\mathcal{M}(\cdot, \mu) := \psi^{-1} \circ \hat{T} \circ \psi(\cdot, \mu), \qquad \mu \in \mathcal{P}. \tag{1}$$

Figure 1 depicts such maps on a simple bidimensional example. The function ψ represents the affine mapping from the physical domain Ω to its reference configuration $\hat{\Omega}$. The FFD control points P are deformed and the morphing is propagated to all the points within the lattice, usually by using Bernstein polynomials but also other choices

are possible. The displacements of such control points along the cardinal directions define the parameters vector μ. The map $\hat{T}(\cdot,\mu) : \hat{\Omega} \to \hat{\Omega}(\mu)$ for a general 3-dimensional case with a lattice of $L \times M \times N$ control points is then defined as follows:

$$\hat{T}(x,\mu) = \sum_{l=1}^{L}\sum_{m=1}^{M}\sum_{n=1}^{N} b_{lmn}(x) P_{lmn}(\mu_{lmn}) \qquad \forall x \in \hat{\Omega},\ \mu \in \mathcal{P}, \tag{2}$$

where b_{lmn} stands for the Bernstein polynomials of degree l, m, n in each direction, and $P_{lmn}(\mu_{lmn}) := P_{lmn} + \mu_{lmn}$, with P_{lmn} indicating the coordinates of the FFD control point with indices l, m, n. By exploiting the good properties of such polynomials it is also possible to impose continuity constraints by adding control points. This is particularly useful at the interface between the unmorphed part of the geometry and the lattice of control points, for which usually physical constraints have to be satisfied.

In the classical offline-online framework we emphasize that the terms $b_{lmn}(x)$ can be computed in the offline phase, while in the online phase we just perform linear combinations of precomputed quantities depending on x and μ-dependent control points locations. Moreover since ψ and ψ^{-1} are affine maps the separation of the two phases remains.

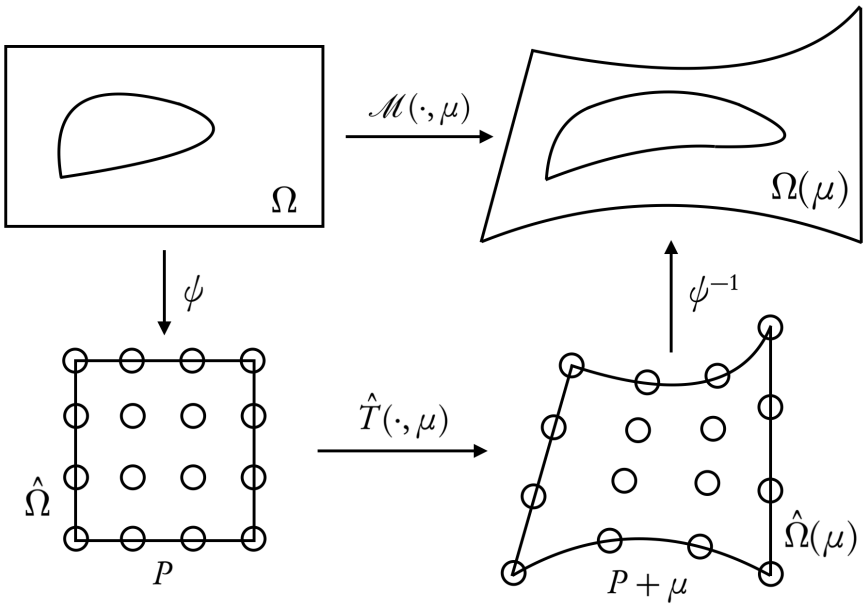

Figure 1: Sketch of the three deformation maps which compose the FFD morphing.

2.1.2 Radial basis functions interpolation

With radial basis function (RBF) [2] we intend any smooth real-valued function $\bar{\varphi} : \mathbf{R}^d \to \mathbf{R}$ whose value depends only on the distance between the input and some fixed point. This means that it exists $\varphi : \mathbf{R}^+ \to \mathbf{R}$ such that $\bar{\varphi}(x) = \varphi(\|x - c\|)$, where $\|\cdot\|$ stands for the Euclidean distance and c is some fixed point.

To illustrate the many possible choices among all the RBF, in Figure 2 we consider a simple univariate example where we approximate the function $f(x) = 0.5\sin(4\pi x) + \exp(x\cos(2\pi x))$ using 6 different type of radial basis functions which are:

- multi-quadratic biharmonic splines (also called multiquadric) [28], defined as $\varphi(r) = \sqrt{1 + (r\varepsilon)^2}$;
- inverted multi-quadratic biharmonic splines (abbreviated in *inverse* in Figure 2) [2], defined as $\varphi(r) = \frac{1}{\sqrt{1+(r\varepsilon)^2}}$;
- Gaussian [2], defined as $\varphi(r) = e^{-(r\varepsilon)^2}$;
- cubic and quintic, defined as $\varphi(r) = r^3$ and $\varphi(r) = r^5$, respectively;
- thin plate [29], defined as $\varphi(r) = r^2\ln(r)$.

We remark that the list does not cover all the possible RBF. Figure 3 reports the interpolation error over the entire domain for the different RBF described above.

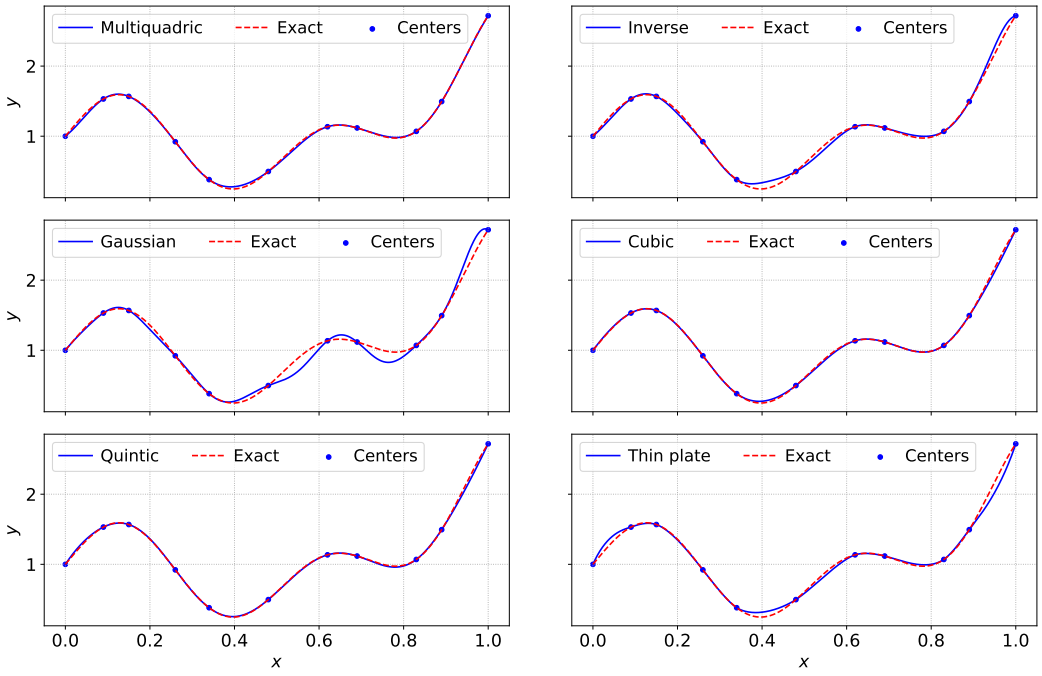

Figure 2: Interpolations corresponding to 6 different RBF. The exact function is depicted with a red dashed line. The parameter ε is equal to the inverse of the average distance between the sampling points.

We follow [30, 31] for a brief presentation of the radial basis functions interpolation method, in order to define $\mathcal{M}(\cdot, \mu)$ for such technique. Let $\{x_{C_i}\}_{i=1}^{N_C}$ be the control points placed over the surface of the object we want to parametrize. We can induce a deformation by moving these control points and interpolating them with a new surface.

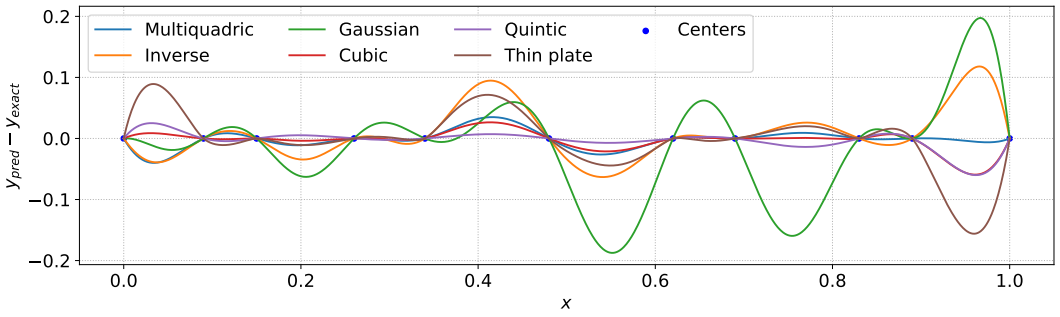

Figure 3: Interpolation error for different types of radial basis functions. Refer to Figure 2 for the actual interpolations.

The parameters vector μ is composed by the displacements of such RBF control points. Let φ be the selected radial basis function, and $\{\gamma_i\}_{i=1}^{N_C}$ be the weights associated to the RBF centered in $\{x_{C_i}\}_{i=1}^{N_C}$, respectively. We can define the deformation map $\mathcal{M}$ for the RBF interpolation technique as

$$\mathcal{M}(x,\mu) := s(x,\mu) + \sum_{i=1}^{N_C} \gamma_i(\mu)\varphi(\|x - x_{C_i}\|) \qquad \forall x \in \Omega,\ \mu \in \mathcal{P}, \tag{3}$$

where $s(\cdot,\mu)$ is a polynomial term of degree 1, which means $s(x,\mu) = S(\mu)x + c(\mu)$. The total unknowns are $d + d^2$ for the polynomial term, and $d \times N_C$ for the weights, where $d = 2$, or 3, depending on the dimension of Ω. Regarding the γ_i, let $\{y_{C_i}(\mu)\}_{i=1}^{N_C}$ be the new control points obtained from x_{C_i} after applying the displacements defined by μ. The interpolation contraints we impose read as follows

$$\mathcal{M}(x_{C_i},\mu) = y_{C_i}(\mu) \qquad \forall i \in [1,\dots,N_C]. \tag{4}$$

For the remaining unknowns due to the polynomial term, we impose the conservation of the total force and momentum (see [2]) as

$$\sum_{i=1}^{N_C} \gamma_i(\mu) = 0, \qquad \sum_{i=1}^{N_C} \gamma_i(\mu) x_{C_i}^{(j)} = 0 \quad \forall j \in [1,\dots,d], \tag{5}$$

where $x^{(j)}$ denotes the j-th component of x.

We emphasize that the RBF contributions $\varphi(\|x - x_{C_i}\|)$ can be precomputed in the offline phase, while for a given μ in the online phase we just need to solve the $d \times N_C + d + d^2$ linear system and perform the matrix vector product in Equation 3.

For an application of the RBF interpolation technique used to morph a carotid artery we mention [32]. For aeronautics fluid-structure interaction problems see [33]. There are also extensions of the method such as the rescaled localized radial basis function interpolation method for nonconforming meshes [34], or the combination with non-intrusive model order reduction [35, 36, 37].

2.1.3 Inverse distance weighting interpolation

Inverse distance weighting is an interpolation method originally proposed in [3]. Let $\{(x_{C_i}, f(x_{C_i})\}_{i=1}^{N_C}$, where $x_{C_i} \in \mathbf{R}^d, f(x_{C_i}) \in \mathbf{R}$ for $i = 1, \dots, N_C$, be the set of the input-output pairs obtained evaluating $f : \mathbf{R}^d \to \mathbf{R}$. The IDW interpolant of f is defined as

$$\mathfrak{I}_{\text{IDW}}[f](x) = \sum_{i=1}^{N_C} w_i(x) f(x_{C_i}), \qquad x \in \mathbf{R}^d, \tag{6}$$

such that $\mathfrak{I}_{\text{IDW}}[f](x_{C_i}) = f(x_{C_i})$ for $i = 1, \dots, N_C$. The functions $w_i : \mathbf{R}^d \to \mathbf{R}$ are defined as

$$w_i(x) = \begin{cases} \frac{\|x - x_{C_i}\|^{-p}}{\sum_{j=1}^{N_C} \|x - x_{C_j}\|^{-p}} & \text{if } \; x \neq x_{C_i}, \\ 1 & \text{if } \; x = x_{C_i}, \end{cases} \qquad \text{for } \; i = 1, \dots, N_C, \tag{7}$$

where the power $-p$ controls the rate influence of the i-th input point and $\|\cdot\|$ refers to Euclidean norm. Such functions require to compute all the distances between the point to interpolate x and the input data $\{x_{C_i}\}_{i=1}^{N_C}$. Assuming the weight function becomes negligible after a certain distance ϵ, we can rewrite Equation 7 into

$$w_i(x) = \begin{cases} \frac{d(x, x_{C_i})^{-p}}{\sum_{j=1}^{N_C} d(x, x_{C_j})^{-p}} & \text{if } \; d(x, x_{C_i}) \in [0, \epsilon)\,, \\ 1 & \text{if } \; d(x, x_{C_i}) = 0, \\ 0 & \text{if } \; d(x, x_{C_i}) \in [\epsilon, +\infty], \end{cases} \qquad \text{for } \; i = 1, \dots, N_C, \tag{8}$$

where for sake of notation we define the distance function $d : \mathbf{R}^d \times \mathbf{R}^d \to \mathbf{R}^+$ as $d(x, y) = \|x - y\|$.

Such interpolation can be adopted as efficient tool for shape parameterization problems. Among the all contributions in literature, we suggest [38] for a parametric computational grid by means of IDW, [33] for application of IDW into fluid-structure interaction problems and [39] for an example of domain parameterization in a reduced order modeling framework.

The main idea for shape parameterization using interpolation techniques is to extend the deformation we impose to a limited set of control points by interpolating their displacements. We define such parameter-dependent displacement at point x_{C_i} as $f_\mu(x_{C_i}) \equiv f(x_{C_i}, \mu) \in \mathbf{R}$, for $i = 1 \dots, N_C$. Interpolating the displacements with IDW, we obtain

$$\mathfrak{I}_{\text{IDW}}[f_\mu](x) = \sum_{i=1}^{N_C} w_i(x) f(x_{C_i}, \mu), \qquad x \in \mathbf{R}^d, \; \mu \in \mathcal{P}. \tag{9}$$

We remark that we have considered until now scalar displacement, but IDW extension to vector functions is trivial: assuming all components are independent, we simply apply IDW to each component. Thus, considering a d-dimensional vector as displacements, we can derive the deformation map $\mathcal{M}(\cdot, \mu) : \mathbf{R}^d \to \mathbf{R}^d$ from the former equation:

$$\mathcal{M}(\cdot, \mu) := \sum_{i=1}^{N_C} w_i(\cdot) f(x_{C_i}, \mu), \qquad x \in \mathbf{R}^d, \; \mu \in \mathcal{P}. \tag{10}$$

Equation 10 shows the decoupling between the spatial dependent terms ($w_i(\cdot)$) and the parameter dependent ones ($f(x_{C_i}, \mu)$). This implies that for any x, we can precompute (and store) the weight functions, requiring to compute only the parameter dependent term for any new evaluation of $\mathcal{M}(\cdot, \mu)$ in the online phase.

2.2 Object specific shape parameterization algorithms: some examples

The general purpose parameterization techniques just described, are ideal in cases in which the artifacts or objects to be deformed are not bound to preserve local or global geometric properties throughout the deformation process. In the engineering field, several objects or their components are often designed for the purpose of carrying out a distinct task or to deliver a specific performance. Hence, the parameter-to-shape relationship must be built so that all the deformations considered result in shapes which would not impair the possibility the component to carry its task or provide the required performance.

There are several examples of this situation, which can also be used to illustrate different possible constraints the geometry can be subjected to. For instance, the aerodynamic design of an aircraft wing tip fuel tank or external drop fuel tank will probably require that any shape considered must be characterized by a certain volume. A similar situation is encountered in the design of the ballast-filled bulb at the bottom of the keel of a sailing yacht. To provide the adequate restoring moment, the bulb weight is fixed, and if the ballast material is not changed this results in a volume constraint.

RBF, IDW, and FFD methodologies in the form described in the previous sections, are not suited to automatically produce shapes that retain a fixed volume. Of course, it is in principle possible to complement such algorithms with the addition of a supplementary parameter which can be used to scale each RBF or FFD deformation in order to obtain the prescribed volume. Such parameter, however, might in some case turn out to be not completely independent with respect to the original one, which must be accounted for in the overall design process.

In addition, there are several situations in which general purpose algorithms cannot be employed without a more invasive and complex adaptation. As an example, we can refer to the design of ship or yacht hulls. Typically, the lateral walls of a hull are constrained to be vertical or at least straight, as this results in easier management and arrangement of internal passengers or cargo areas. Compared to a global constraint, such as the aforementioned fixed volume one, this local constraint is more difficult to fulfill. Both RBF and FFD would in fact require significant modifications if they have to be effective for the generation of parameterized hulls with vertical or straight side walls. An even more important case of shapes that present local constraints limiting the effectiveness of general purpose shape parameterization algorithms is that of aircraft wings and propeller or turbo machinery blades. These objects are originally designed to have airfoil sections with standard and tabulated shapes having well documented fluid dynamic performance. As remarkably efficient estimates of the overall wing or blade performance can be obtained from its airfoil section ones, it is very important

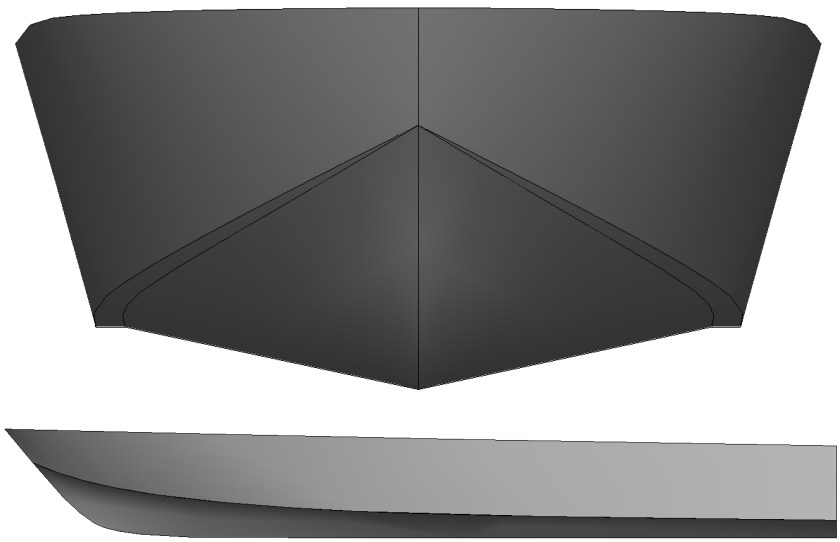

Figure 4: Front view (on top) and side view (on bottom) of a planing hull featuring a double chine line and substantially straight walls above the chine line.

that any parameterization algorithm applied on these artifacts does not lead to sections with non tabulated shape or performance. It is clear that also in this case RBF and FFD as they have been described cannot generate shapes that grant this property.

In light of all the problems illustrated by these examples, in this section we will discuss shape parameterization algorithms adapted to specific design problems. In particular, making use of some of the aforementioned design cases (i.e. planing hull yachts and naval propeller blades) we will illustrate possible approaches that can be used when general purpose shape parameterization methods cannot be directly employed.

2.2.1 Planing Yacht Hulls

We now focus on the case of the planing hull yacht illustrated in Figure 4. The geometry of this kind of hull is typically characterized by a V-shaped lower part, surmounted by a lateral wall which is typically straight and — except in the bow region — presents a constant angle with respect to the vertical direction. The V-shaped bottom part of the hull and its lateral walls are typically divided by a sharp edge named chine. The example presented in Figure 4 presents a double chine line which delimits a narrow chine surface.

In the hydrodynamic design process of this kind of hulls, engineers typically want to investigate how shape modifications affect performance in terms of lift, drag and stability (or seakeeping). It is quite clear that deforming the V-shaped bottom part of the hull, which is in contact with the water throughout the navigation, will have the highest impact on the overall hydrodynamic behavior of the hull. For this reason,

design engineers are mainly interested in modifying the lower part of the hull and — at a lower extent — the chine surface, while they want to keep the top part of the hull substantially non deformed. Under these requirements, FFD can hardly be used in this context, as its smooth shape functions would not allow to sharply separate what happens above and below the chine lines. As for RBF, it is still possible to employ its control points displacements to impose the desired deformation on the bottom part of the hull, and a somewhat rigid displacement on the top part, so as to accommodate for the bottom part deformation. Yet, this opens the problem of identifying the RBF control points displacements that would result in the desired deformations of the bottom and top hull regions. So, rather than solving the latter problem, it is convenient to deal with the more intuitive problem of developing a parameterized law to displace any point on the hull surface and generate admissible overall deformations. As we will see in section 3.3, RBF is still used in this context as an extremely valid tool to extend the hull surface deformations to the surrounding volume and adapt the volumetric computational meshes and carry out parameterized fluid dynamic simulations.

The parameterized law to displace any point on the hull is in this case developed taking advantage of the main design features of the object at hand. As suggested in Figure 5, the transverse vertical sections present the same topology for most of the hull length. Each section is in fact characterized by a bottom vertex which coincides with the section plane intersection with the keel line, and by two vertices resulting from the section plane intersection with the chine lines. So, the parameterization strategy devised acts modifying each section with the same strategy based upon these easily identifiable points. More specifically, the V angle of the hull bottom is altered by rigidly rotating around the keel line intersection and with the desired angle θ, all the section points located between the internal chine line and the keel line intersections. This rotation results in a net displacement of the internal chine line intersection point. Such displacement is used to rigidly translate the section of the chine and wall surfaces, and ensure that the top part of the section remains connected with the rotating bottom. With the same principle, once these two steps are completed, it is possible to add rotations to the chine surface and, if desired, to the wall surface. Finally, a further parameterized function can be introduced to modulate the sectional rotations introduced along the hull longitudinal coordinate.

In summary, the present example suggests that when a highly engineered object such as a planing yacht hull is to be parameterized fulfilling some specific shape constraints, a viable possibility is that of exploiting the main topological features of the original geometry. Such topological features are often a result of the characteristics that designers want their object to possess. So, as is the case of the keel and chine lines for a planing hull, they can be used to build a parameterization law that it useful to engineers.

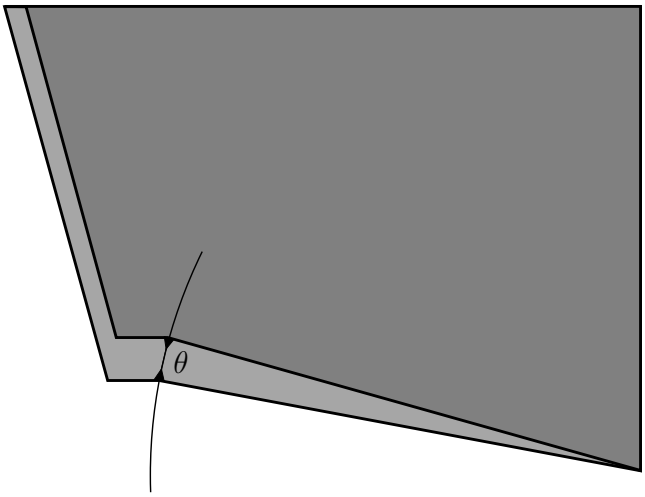

Figure 5: Vertical section of a planing hull perpendicular to its longitudinal axis. The bottom vertex of the section is the intersection of the section plane with the keel line. The intersections of the two chine lines are also visible on the bottom left side of the section. The parameterization strategy first step consists in a rigid rotation of the bottom part of the section around the keel intersection. This is combined with a rigid translation of the chine and wall surface sections needed to keep the top part of the section in contact with the displaced bottom part.

2.2.2 Propellers

As previously mentioned, aircraft wings and propeller or turbo machinery blades represent further examples of objects for which RBF, IDW, and FFD are not ideal shape parameterization algorithms. This is mainly because such objects are designed starting from a set of two dimensional airfoil sections, which are placed in the three dimensional space according to some strict design rule. If we consider for instance a propeller blade, a specific design table first reports the list of airfoil section shapes to be used in its generation. In such design table, each airfoil section, originally described in the $\mathbb{R}^2$ space, is also complemented with a set of parameters (radius, chord length, pitch angle, skew angle and rake) which indicates how the airfoil section should be ultimately placed in the $\mathbb{R}^3$ space. As depicted in the left part of Figure 6, the airfoil sections are first laid on cylinders with axes coinciding with the propeller one. Thus, each airfoil section is associated to a specific radius. The chord length parameter represents the distance between the leading and trailing edges of the airfoil, projected on the section cylinder. The skew and pitch angles represent instead the rotations prescribed for each section around the blade axis and the entire propeller axis, respectively. Finally, the rake is the translation along the propeller axis prescribed for each section. For more details about propeller blade design the interested reader is referred to [6].

It should at this point appear clear that the design procedure described automatically leads to parameterized propeller blades. In fact, the value of chord length, pitch angle, skew angle at each radius corresponding to the airfoil sections available, are indeed parameters of the generated propeller blade. Moreover, several different families of

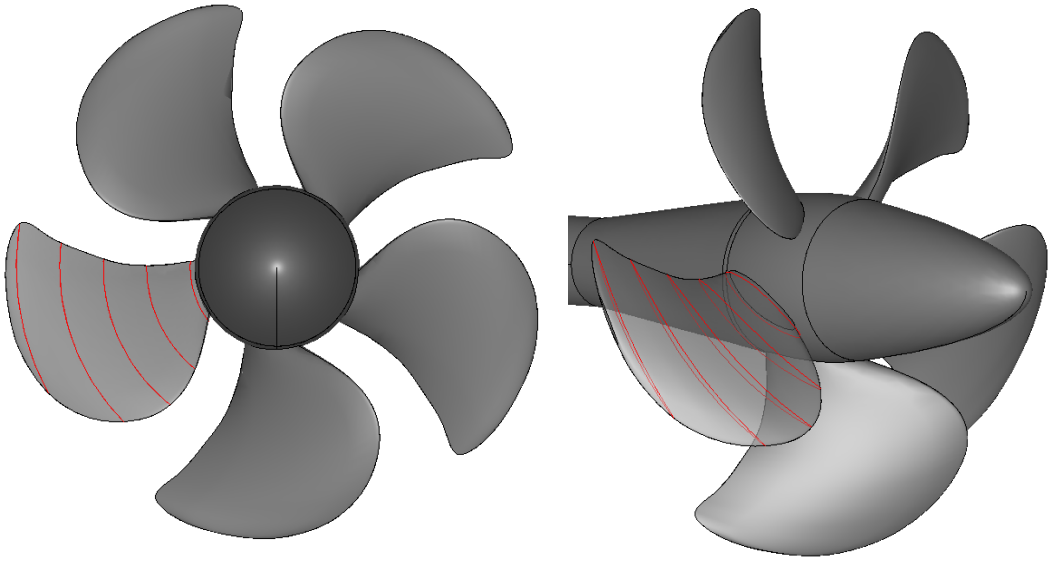

Figure 6: Front view (on the left) and lateral view (on the right) of a ship propeller. In both images, one of the five propeller blades surface is transparent, and a set of airfoil cylindrical sections is indicated in red to indicate how the blade surface is generated as the envelope surface of the single sections.

parameterized airfoil sections are available across the engineering community. For instance, the NACA airfoils [40] are parameterized based on quantities such as maximum thickness, maximum deflection of the camber line and location along the chord of the maximum deflection point.

Despite the previous description suggests that the geometry of propeller blades — as well as turbo machinery blades and aircraft wings — possesses an inherent parameterization, it is still possible to modify such parameterization to have better control on the finalized shape. As depicted in Figure 7 for the case of the pitch radial distribution it is possible to make use of B-splines to interpolate the blade parameters variation along the radial coordinates. This results in the possibility of switching the shape parameters to the control points of the single chord length, pitch, skew and rake radial distribution curves. Given the smoothness of B-splines, this approach results in a smoother parameter dependence, especially considering that the CAD modelers which are used to build the three dimensional surfaces passing through the airfoil sections, are also based on B-splines and NURBS surfaces. A further advantage of this approach is that if the two dimensional airfoil parameters are also interpolated with this strategy, the number of airfoil sections can be arbitrarily increased to provide the CAD modeler with more sections and have more control on the generated three dimensional surfaces. For an efficient selection of such parameters using the active subspaces method please refer to [41].

Both the standard generation of the three dimensional propeller blade from its design table, and from B-spline interpolated radial distributions of airfoil parameters, have

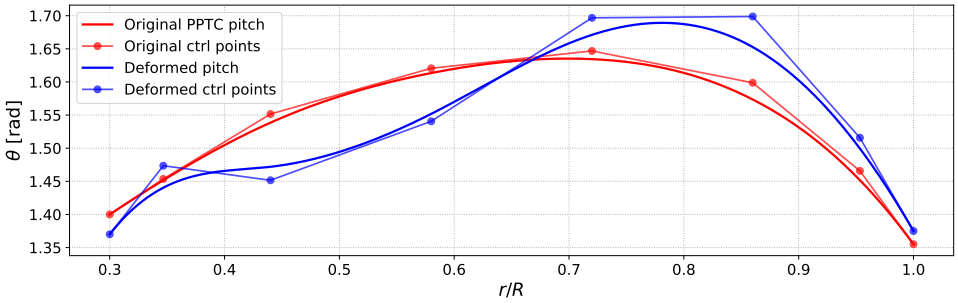

Figure 7: The airfoil sections pitch as a function of relative radial coordinate. The plot shows both an initial pitch distribution (red continuous line) and the control points associated with its B-spline interpolation (red piecewise linear line). A new, smooth, pitch radial distribution (blue continuous line) is then obtained through a displaced set of pitch control points (blue piecewise linear line).

been implemented in the Python package BladeX [42]. Given sectional values of airfoil section, radii, chord length, pitch and skew angle and rake, BladeX allows for direct generation of CAD representations of the corresponding blade surface. In addition, the software supports the B-spline representation of the airfoil sections parameters variations along the relative radial coordinate, which as said can be used to re-parameterize the blade shape, or to enhance the smoothness of the resulting CAD surface.

3 Tests and applications

The parameterization strategies described in the previous sections have been applied to different design problems. In the present section, we will describe a wide spectrum of shape parameterization examples, with particular focus on practical details such as the file format containing the geometric information of the considered objects. In addition, we will distinguish between cases in which the parameterization tools only act on the surfaces describing the object of interest [4, 41], or are instead used to deform a three dimensional computational grid [43]. In the former cases, the deformation of a codimension[1] 1, $\mathbb{R}^2$ surface embedded in $\mathbb{R}^3$ is studied. In the latter case, the complete morphing of a codimension 0 computational domain $\Omega \subset \mathbb{R}^3$ is instead carried out.

3.1 Geometric model parameterization

The present subsection focuses on the deformation of an object or artifact shape through parameterization of the surfaces that delimit it. We point in fact out that most Computer-Aided Design (CAD) tools used throughout the industrial manufacturing community, identify the shape of three dimensional objects through the mathematical description of its codimension 1 boundaries. For this reason, we first discuss applications in which surfaces in $\mathbb{R}^3$ are specified by means of both triangulations and parametric equations.

3.1.1 Triangulated surfaces

Triangulated surfaces describe a codimension 1 surface through a finite non overlapping triangulation obtained by discretizing the original surface manifold. Such objects are particular widespread in many computational and simulation contexts, since their discrete nature. Contrary to CAD surfaces (that will be extensively discussed in next subsection), triangulated surfaces are indeed defined only by the coordinates of the triangles vertices and their topological connections, resulting easier to be handled. Usually stored into STereo Lithography interface (STL) files [44], triangulated surface are widely adopted within the meshing phase. In several applications, the surface deformation is preferred to computational mesh deformation, since it allows to apply the meshing procedure after surface morphing, ensuring a better control on the quality of final mesh. It is possible to note that such approach requires to create a new grid for all the deformations, increasing the computational effort needed.

Given the generic parametric deformation map $\mathcal{M}(\cdot, \mu)$, its application to triangulated surface is obtained by applying such mapping to all the nodes of the triangulation. Thus, the topology is maintained unaltered. Figure 8 shows a demonstration of surface

[1] The codimension of a subspace U of a vector space V is the dimension of the quotient space V/U. It is indicated by the symbol $\mathrm{codim}_V U$ and is equal to the dimension of the orthogonal complement of subspace U in V, namely

$$\mathrm{dim}U + \mathrm{codim}_V U = \mathrm{dim}V.$$

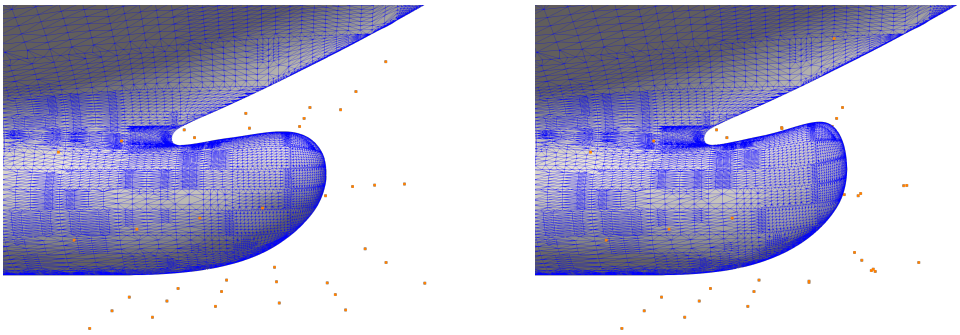

Figure 8: FFD application to the STL triangulation representing the bulbous bow of a cruise ship hull. The left panel, which represents the original bow geometry, also displays the FFD undeformed control point lattice. On the right, the visible displacement of the control point results in a deformed STL geometry.

deformation applied to the bulbous bow of a naval hull by means of FFD for a shape optimization problem [14] where reduced order modeling was involved to reduce the computational load. Here the control points are spanned around the bulb of the ship, in order to induce a deformation only in such region and reducing then the wave resistance of the ship (in a calm-water environment). As we can note from the picture, the surface derivative is kept continuous, an important feature for many industrial applications.

3.1.2 CAD surfaces

We now describe some examples of the application of the shape parameterization strategies discussed in Section 2 to Non-uniform rational basis spline (NURBS) surfaces [45]. NURBS are an important component of numerous industry and computer graphics wide CAD standards such as IGES [46], STEP [47] and ACIS.

A full mathematical description of NURBS and their application to analytic geometry is beyond the scope of the present work (we refer the interested reader to [48, 49]). Yet, for our purposes it is at least important to recall here that NURBS curves and surfaces are defined by their order, by a set of weighted control points and by a knot vector. Control points in particular are always either directly linked to the curve or surface, or behave as if they were connected by a spring to it. For this reason, they offer the possibility to modify the NURBS surface by altering the position of the control points. This very intuitive editing mechanism — which is among the features that made this mathematical instrument very popular in the CAD and computer animation industries — is here exploited to apply any parameterization strategy to NURBS surfaces. As illustrated in Figure 9, the deformation of a NURBS curve through FFD is carried out directly applying the parameterization algorithm to the NURBS control points, and then computing the resulting modified shape.

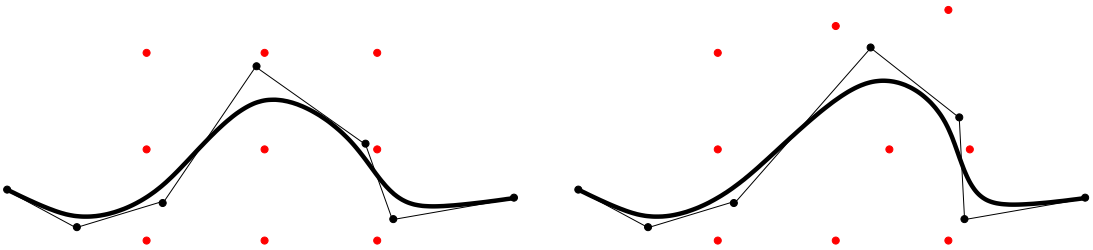

Figure 9: A simplified two dimensional example of the application of FFD to a NURBS curve. On the left, the procedure starts by placing a FFD lattice (red dots) on top of a NURBS curve. On the right, the displacement of the FFD control points is used to modify the position of the NURBS control points (black dots). The modified control polygon results in the morphed NURBS curve displayed on the right.

It is important to point out here that the deformed surface obtained moving the CAD control points is identical to the one obtained moving every surface point only if the transformation law applied is affine [48]. We also recall that most of the parameterization algorithms discussed in the present work do not, in general, lead to affine transformations. Thus, we must remark that deforming a CAD shape through displacement of its control points might result in slight differences compared to deforming it through a surface triangulation. We report that, for all the examples discussed in the present chapter, the difference between applying the same deformation to NURBS surfaces or to their triangulated equivalent is quite modest. Yet, we stress that this issue must be taken into account if consistency between NURBS and triangulated surfaces deformations must be enforced up to a very strict tolerance. Ideally, in the latter case it is possible to resort to least square fitting to generate a novel NURBS surface passing as close as possible to the deformed triangulation. We warn, however, that this procedure would still bring in the possibly significant source of error associated with the least square minimization.

Figures 10 and 11 depict an example of the application to CAD geometries of the parameterization algorithms described in Section 2 and presented in [17]. The shape considered is that of the DTMB 5415 hull [50], which was originally conceived for the preliminary design of a US Navy Combatant ship. FFD is here used in the bow regions to displace the control points of the hull NURBS surfaces and create a family of hulls which feature deformed sonar domes. The images display both the front and side views of the original DTMB 5415 IGES geometry (right and bottom), and of the IGES geometry of a specific modification in which the sonar dome has been enlarged and inclined downwards (left and top).

As can be appreciated in the pictures, the DTMB 5415 hull portrayed in Figures 10 and 11 is composed of three NURBS surfaces. Such faces border with each other through shared edges in which the shape is C^1 or C^0 continuous only up to a tolerance specified in the IGES file containing the hull data. We use such very common example to remark that an additional problem associated with deforming NURBS surfaces through displacement of their control points, is that it can introduce discontinuities in

shapes composed of several CAD patches. Despite all the parameterization strategies described in this chapter are designed to be continuous maps from $\mathbb{R}^3$ to $\mathbb{R}^3$, the control point distribution in two bordering faces can in principle be so different, that the deformation affects such two surfaces in a non equal way. This might potentially alter the continuity tolerance of the original CAD file, and result in small gaps, overlaps or normal vector jumps appearing between the NURBS surfaces which were originally smoothly joined. Such loss of C^1 and/or C^0 continuity is more pronounced when the deformation map gradients are high relative to the local NURBS control points density. Thus, a solution to mitigate the problem is that of adding control points to the NURBS surfaces. In the vast majority of cases considered, such procedure proved effective in obtaining small continuity tolerances in composite CAD surfaces deformations, although the increased amount of control points results in heavier CAD files [9].

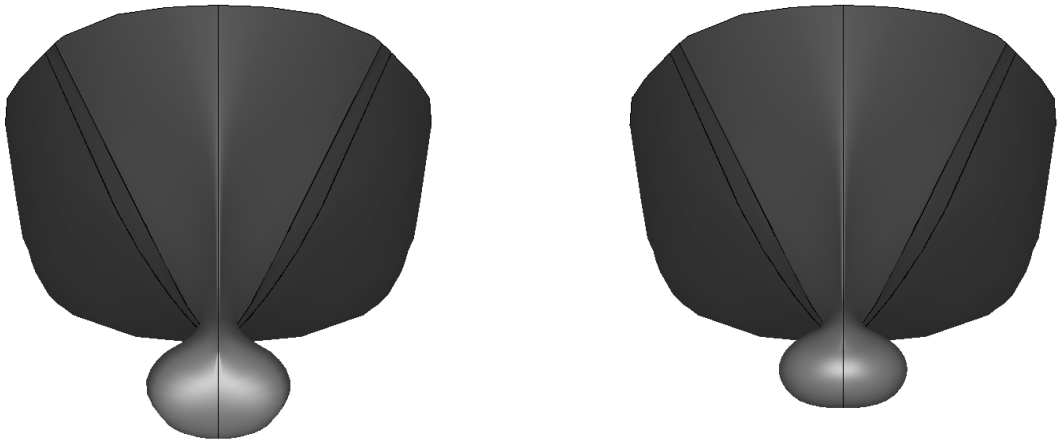

Figure 10: An application of the FFD algorithm to the NURBS surface specified by means of and IGES file. A front view of the DTMB 5415 navy combatant hull featuring a modified and enlarged bow is displayed on the left; the original hull surface is displayed on the right for reference.

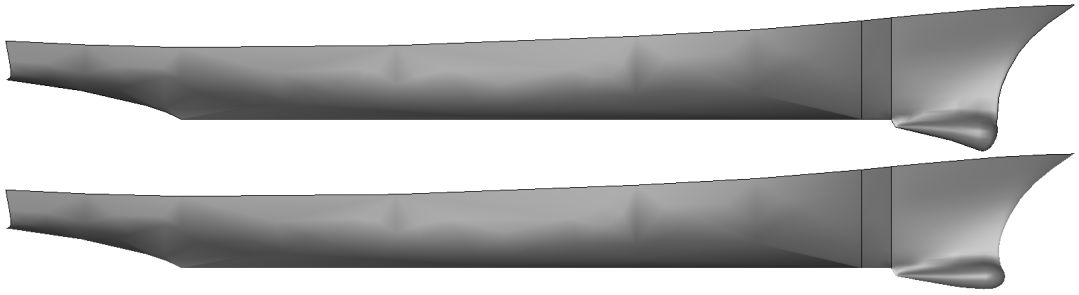

Figure 11: A side view of the IGES geometry of the DTMB 5415 navy combatant hull shown in Figure 10. The top picture depicts the CAD surface of the hull, modified via FFD. The bottom picture shows the original hull surface.

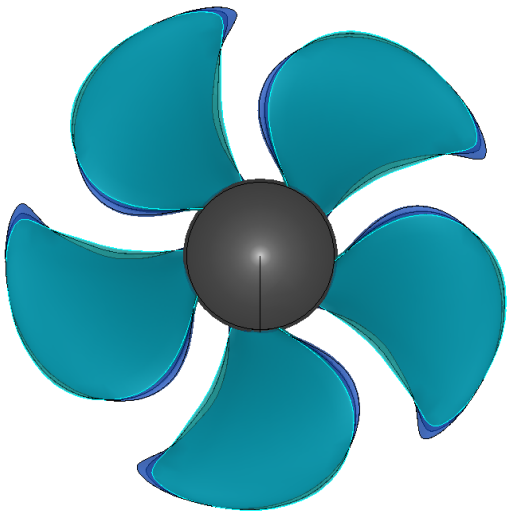

Figure 12: Representation of a family of ship propellers generated with bottom-up construction. The different propellers are obtained in this case through scaling of the skew angle curve. In particular, in the image an original propeller blade shape is presented along with two modifications resulting from scaling the skew curve by factors 0.8 and 1.1, respectively.

The ship propeller family shown in Figure 12 represents instead a different strategy to obtained parameterized IGES geometries. The algorithm for propeller design described in Section 2 has been used for the bottom-up generation of parameterized ship propeller blades. In this specific case, each blade is generated as the NURBS surface passing through 12 airfoil sections, each being defined by different curves in the XY plane, and different values of radial coordinate, chord length, pitch angle, skew angle and rake. The procedure implemented in the library BladeX [42] allows for the specification of the airfoil sections shape by means of a NACA 4 or 5 digits identifier or by means of a CSV file. Once the radial coordinate, chord length, pitch angle, skew angle and rake associated with every section are specified by the user, the airfoil sections points are placed in the three dimensional space, and a series of NURBS curves are generated, passing through all the points of each section. Finally, the NURBS surface passing through all the sectional curves is generated and joined to the propeller hub. The latter operations are carried out making use of functions of the CAD Python library PythonOCC [51]. In the example reported in Figure 12, the original blade shape is portrayed along with two variants generated altering the skew angle distribution along the radial coordinate. We recall that the skew angle designs the position of each radial blade section along the circumferential coordinate of the propeller disc plane. Thus, the blade generated scaling the skew values by a factors 1.1 appears more curved along the radial direction with respect to the original design. On the other hand, scaling the skew values by a factor 0.8 results in in a propeller with blades that appear straighter along the radial direction.

We finally point out that the specific deformation displayed in Figures 10 and 11 has been part of a family of 130 IGES hulls generated via FFD in the framework of a CFD simulation campaign presented in [17]. In such work, the geometric deformation

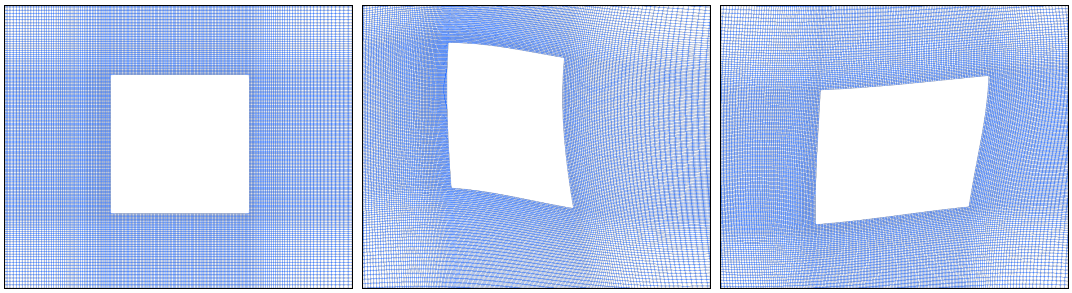

Figure 13: Examples of computational domain deformation using free form deformation: the first picture (left) shows the edges of the original mesh, while the others illustrate two deformed domains.

was concentrated in the region surrounding the bulbous bow. A further set of 130 hulls has been obtained deforming the full length of the hull, and has been used to carry out the CFD simulations presented in [4]. A family of 500 ship propellers as the one depicted in Figure 12 has been also designed in the framework of an optimization campaign presented in [41]. In all such works, active subspaces have been successfully used to detect possible redundancies in the geometric parameters introduced with FFD. On this note, we point out that active subspaces analysis is an extremely useful post processing tool to complement shape parameterization algorithms, as it allows for the identification of a reduced number of combined parameters which have most effect on the output of interest. In addition, it also can be used to evaluate the output sensitivity to variations of the original geometric parameters and eliminate inconsequential parameters.

3.2 Computational domain parameterization

In the present section, we turn our attention to codimension 0 domains, and their parameterization through the methodologies described in Section 2. Since these domains include subsets of $\mathbb{R}^3$ in 3D and of $\mathbb{R}^2$ in 2D, the examples will mainly refer to the morphing of volumetric computational grids for the solution of PDE problems arising from fluid dynamic or structural simulations.

3.2.1 FFD and RBF applied to computational grids

A first example, presented in Figure 13, illustrates the application of FFD for the deformation of the structured surface mesh surrounding a square in a 2D application. The images, which refer to the original configuration (left) and to two deformed domains (center and right), depict a close up of the grid region surrounding the square.

Figure 14 shows instead an application of RBF in the context of a CFD simulation campaign [52]. The pictures show two possible modifications of the Ahmed body obtained altering the inclination of the diagonal surface at the rear end body. The computational grid for the CFD simulations is deformed through the distribution of

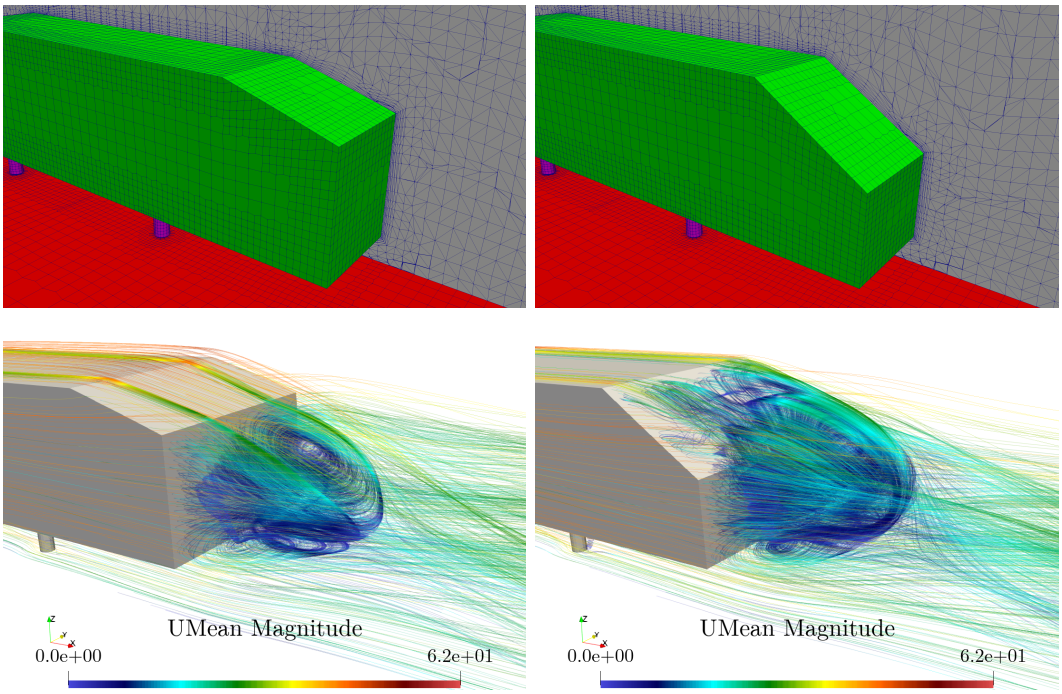

Figure 14: The application of RBF shape parameterization to the Ahmed body and the surrounding volumetric mesh for CFD simulations. In the test illustrated, the inclination of the diagonal surface at the rear end of the Ahmed body is modified to investigate its effect on aerodynamic performance. Two configurations, including a vertical cut of the volumetric mesh, are shown in the top images. For reference, path lines of mean velocity field resulting from the corresponding Unsteady RANS simulations are shown in the bottom pictures.

a series of control points on the boundaries of the 3D domain, and the displacement of a portion of those located on the modified diagonal surface. The vertical cut of the resulting volumetric grids are shown in the plots on the top. The corresponding bottom images display the path lines of the mean velocity field resulting from the Unsteady RANS simulations carried out with the deformed grid. As can be appreciated, the deformation strategy is able to alter the original computational mesh without impairing the convergence and effectiveness of the CFD simulations.

3.3 The use of RBF to extend boundary deformation to internal grid nodes

In this section, we discuss a further interesting and useful application of RBF to propagate the boundary deformation to the internal nodes of volumetric grids for PDEs based simulations. Most of the algorithms designed to modify the shape of specific artifacts — such as the planing hulls and propellers as described Section 2.2 — result in the deformation of co-dimension one manifolds. Thus, if these deformation methodologies must be used in a pipeline involving volumetric PDEs resolution, they must be necessarily complemented by an adequate strategy to extend the boundary

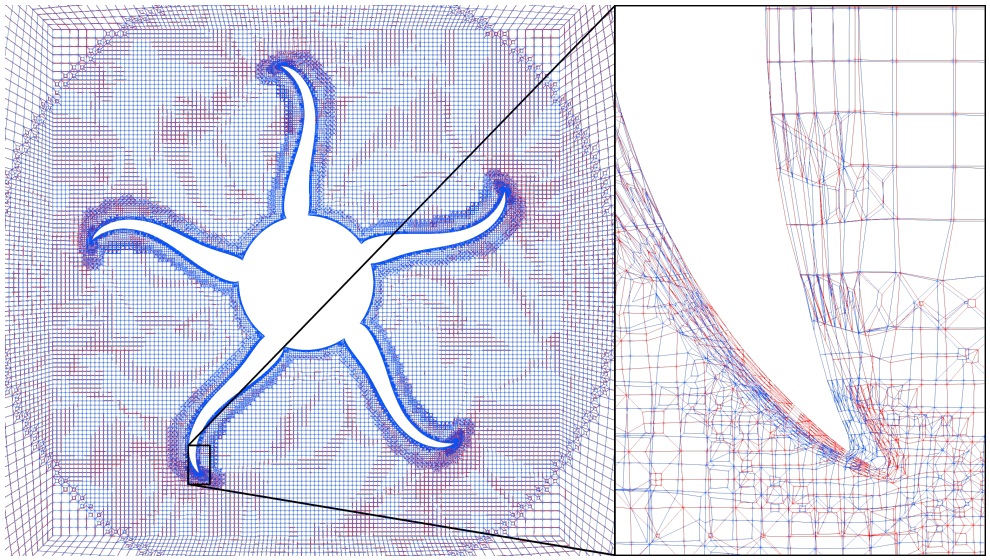

Figure 15: Examples of computational domain deformation using free-form deformation: the first picture (left) shows the edges of the original mesh, while the others illustrate two deformed domains.

displacements to the whole, co-dimension zero, computational domain.

Figure 15 shows the deformation of the CFD computational mesh past a PPTC ship propeller, following a variation of the blades pitch in the tip region obtained with the strategy described in Section 2.2.2. In the plot, the blue lines represent the edges of a cut of the volumetric mesh obtained by means of a plane perpendicular to the propeller axis. The red lines refer instead to the edges of the deformed mesh in correspondence with the same sectional plane. The close up of the blade tip region displayed on the right, show how the deformation of the blade surface is smoothly propagated throughout the internal nodes of the volumetric mesh, avoiding the presence of inverted or highly skewed cells.

Finally, we want to stress that for practical reasons the RBF extension of to volumetric meshes is often convenient for surfaces originally deformed with FFD. As previously discussed, in principle FFD allows for simultaneous deformation of both boundary surface and volumetric grid. Yet, when deforming volumetric meshes, the aim is that of distributing the deformations over a high number of cells, rather than concentrating all the displacements in a very confined region in which cells can get distorted or even inverted. But because FFD only affects points located within the control points lattice, this means that the latter must extend for a bigger volume. In addition, to maximize the volumetric mesh quality, the user must include more control points in the lattice to make sure that different deformation magnitudes are imposed in regions close to the deformed surface and far from it. Such manual control over the local mesh deformation can often become quite cumbersome.

For such reasons, in several cases it will be convenient to only employ FFD to deform

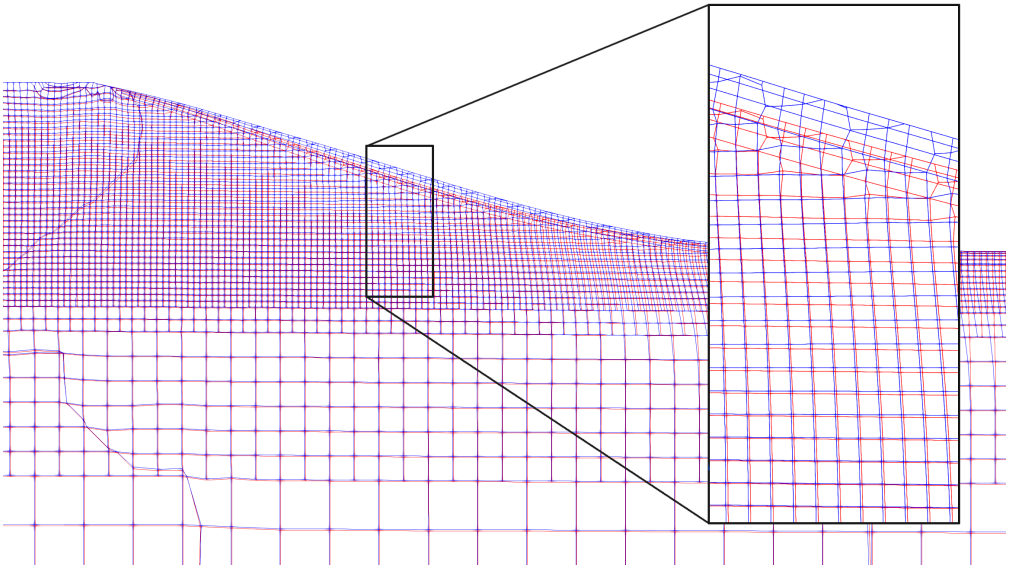

Figure 16: A sectional view of a volumetric grid deformation carried out by means of RBF in a naval engineering problem. The blue lines denote the original volumetric mesh edges, while the red lines indicate the deformed configuration.

the boundary surface mesh, or a triangulated format of its geometry. Then, after such a surface mesh has been modified through FFD, it is possible to resort to RBF to propagate the boundary displacements to the internal nodes of the volumetric mesh for CFD simulations. In a broader sense, RBF is an interpolation algorithm, and is here used to estimate the nodes displacement function on every point of a domain, based on the function values prescribed on the boundary. In the framework of RBF, this is done by linear combinations of radial bases used to approximate in any point of the domain a function with values prescribed only in a finite number of points. In the case of interest, the displacement field function is prescribed only on the points of the boundary surface, and is interpolated in the positions corresponding to every node of the volumetric mesh. The application of such procedure in a naval engineering test case (see [5] for further detail) is illustrated in Figure 16. The illustration displays a sectional view of the original (blue lines) and deformed (red lines) volumetric mesh around the surface of a ship hull. As documented in the original publication, the combined FFD/RBF strategy has been use to successfully generate 200 shape variations of a container ship hull, and accordingly deform the volumetric grids to carry out an optimization campaign based on the OpenFOAM CFD solver library. The output of the checkMesh utility of the OpenFOAM library confirms that the mesh deformation is not significantly altering both the minimum face area and minimum cell volume, while it only leads to a modest — 7.2% at most — increase of average mesh non-orthogonality index. Such an increase, however, is well below level that can significantly affect the result of the CFD simulations.

4 Concluding remarks

In this chapter we presented a diverse range of possible non-affine shape morphing and parameterization techniques. In the general purpose group we introduced free form deformation, radial basis function interpolation, and inverse distance weighting interpolation. These techniques are characterized by the fact that they can be applied to any object or part of it, without any limitations. Then we presented more specific deformation methods which are necessary when we want to morph highly engineered artifacts while preserving some characteristics of these objects. We focused on naval engineering applications, where deformations of a marine propeller and a side-wall of a planing motor yacht are presented. These artifacts have also to be actually manufactured so general purpose parameterizations can easily generate object which are very costly to produce or infeasible.

In the second part of the chapter we presented methodologies to guarantee that any modification of the geometry is reflected in the computational mesh. The advantage of these techniques is twofold: first the shapes validated by simulations will also be available for manufacturing; second we can easily apply non-intrusive reduced order methods to the database of solutions since we preserve the same number of degrees of freedom.

Future studies will address the possibility to apply the free form deformation technique while satisfying some constraints. For example we could want to preserve the volume of the deformed object, or we could want to impose the presence of a second object near the one to morph.

References

[1] Sederberg T. W. and Parry S. R. Free-form deformation of solid geometric models. *ACM SIGGRAPH Computer Graphics*, 20(4):151–160, 1986.

[2] Buhmann M. D. *Radial basis functions: theory and implementations*, volume 12 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, 2003.

[3] Shepard D. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of 1968 23rd ACM National Conference*, pages 517–524. Association for Computing Machinery, 1968.

[4] Tezzele M., Salmoiraghi F., Mola A., and Rozza G. Dimension reduction in heterogeneous parametric spaces with application to naval engineering shape design problems. *Advanced Modeling and Simulation in Engineering Sciences*, 5:25, 2018.

[5] Demo N., Tezzele M., Mola A., and Rozza G. Hull shape design optimization with parameter space and model reductions, and self-learning mesh morphing. *Journal of Marine Science and Engineering*, 9(2):185, 2021.

[6] Carlton J. S. *Marine Propellers and Propulsion*, chapter 6: Propeller performance characteristics, pages 88–135. Butterworth-Heinemann, Oxford, second edition, 2007.

[7] Rozza G., Huynh D. B. P., and Patera A. T. Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations. *Archives of Computational Methods in Engineering*, 15:229–275, 2008.

[8] Rozza G., Hess M., Stabile G., Tezzele M., and Ballarin F. Basic Ideas and Tools for Projection-Based Model Reduction of Parametric Partial Differential Equations. In Benner P., Grivet-Talocia S., Quarteroni A., Rozza G., Schilders W. H. A., and Luis Miguel Silveira, editors, *Model Order Reduction*, volume 2, chapter 1, pages 1–47. De Gruyter, Berlin, Boston, 2020.

[9] Tezzele M., Demo N., Mola A., and Rozza G. PyGeM: Python Geometrical Morphing. *Software Impacts*, 7:100047, 2021.

[10] Samareh J. Aerodynamic shape optimization based on free-form deformation. In *10th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 4630, 2004.

[11] Rozza G., Koshakji A., and Quarteroni A. Free Form Deformation techniques applied to 3D shape optimization problems. *Communications in Applied and Industrial Mathematics*, 4, 2013.

[12] Ballarin F., Manzoni A., Rozza G., and Salsa S. Shape optimization by free-form deformation: existence results and numerical solution for stokes flows. *Journal of Scientific Computing*, 60(3):537–563, 2014.

[13] Salmoiraghi F., Scardigli A., Telib H., and Rozza G. Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation. *International Journal of Computational Fluid Dynamics*, 32(4–5):233–247, 2018.

[14] Demo N., Tezzele M., Gustin G., Lavini G., and Rozza G. Shape optimization by means of proper orthogonal decomposition and dynamic mode decomposition. In *Technology and Science for the Ships of the Future: Proceedings of NAV 2018: 19th International Conference on Ship & Maritime Research*, pages 212–219. IOS Press, 2018.

[15] Rozza G., Manzoni A., and Negri F. Reduction strategies for pde-constrained optimization problems in haemodynamics. In *Proceedings of the 6th European Congress on Computational Methods in Applied Sciences and Engineering*, pages 1748–1769. Vienna Technical University, 2012.

[16] Lombardi M., Parolini N., Quarteroni A., and Rozza G. Numerical simulation of sailing boats: Dynamics, FSI, and shape optimization. In Buttazzo G. and Frediani A., editors, *Variational Analysis and Aerospace Engineering: Mathematical Challenges for Aerospace Design*, pages 339–377. Springer, 2012.

[17] Demo N., Tezzele M., Mola A., and Rozza G. An efficient shape parametrisation by free-form deformation enhanced by active subspace for hull hydrodynamic ship design problems in open source environment. In *Proceedings of ISOPE 2018: The 28th International Ocean and Polar Engineering Conference*, volume 3, pages 565–572, 2018.

[18] Diez M., Campana E. F., and Stern F. Design-space dimensionality reduction in shape optimization by Karhunen–Loève expansion. *Computer Methods in Applied Mechanics and Engineering*, 283:1525–1544, 2015.

[19] Tezzele M., Demo N., Gadalla M., Mola A., and Rozza G. Model order reduction by means of active subspaces and dynamic mode decomposition for parametric hull shape design hydrodynamics. In *Technology and Science for the Ships of the Future: Proceedings of NAV 2018: 19th International Conference on Ship & Maritime Research*, pages 569–576. IOS Press, 2018.

[20] Tezzele M., Demo N., and Rozza G. Shape optimization through proper orthogonal decomposition with interpolation and dynamic mode decomposition enhanced by active subspaces. In Bensow R. and Ringsberg J., editors, *Proceedings of MARINE 2019: VIII International Conference on Computational Methods in Marine Engineering*, pages 122–133, 2019.

[21] Demo N., Tezzele M., Mola A., and Rozza G. A complete data-driven framework for the efficient solution of parametric shape design and optimisation in naval engineering problems. In Bensow R. and Ringsberg J., editors, *Proceedings of MARINE 2019: VIII International Conference on Computational Methods in Marine Engineering*, pages 111–121, 2019.

[22] Demo N., Tezzele M., and Rozza G. A non-intrusive approach for reconstruction of POD modal coefficients through active subspaces. *Comptes Rendus Mécanique de l'Académie des Sciences, DataBEST 2019 Special Issue*, 347(11):873–881, 2019.

[23] Salmoiraghi F., Ballarin F., Heltai L., and Rozza G. Isogeometric analysis-based reduced order modelling for incompressible linear viscous flows in parametrized shapes. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1):21, 2016.

[24] Garotta F., Demo N., Tezzele M., Carraturo M., Reali A., and Rozza G. Reduced Order Isogeometric Analysis Approach for PDEs in Parametrized Domains. In D'Elia M., Gunzburger M., and Rozza G., editors, *Quantification of Uncertainty: Improving Efficiency and Technology: QUIET selected contributions*, volume 137 of *Lecture Notes in Computational Science and Engineering*, pages 153–170. Springer International Publishing, Cham, 2020.

[25] Rozza G., Malik M. H., Demo N., Tezzele M., Girfoglio M., Stabile G., and Mola A. Advances in Reduced Order Methods for Parametric Industrial Problems in Computational Fluid Dynamics. In Owen R., de Borst R., Reese J., and Pearce C., editors, *ECCOMAS ECFD 7 – Proceedings of 6th European Conference on Computational Mechanics (ECCM 6) and 7th European Conference on Computational Fluid Dynamics (ECFD 7)*, pages 59–76, 2018.

[26] Salmoiraghi F., Ballarin F., Corsi G., Mola A., Tezzele M., and Rozza G. Advances in geometrical parametrization and reduced order models and methods for computational fluid dynamics problems in applied sciences and engineering: overview and perspectives. In Papadrakakis M., Papadopoulos V., Stefanou G., and Plevris V., editors, *Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering,* Crete, Greece, 5-10 June 2016, volume 1, pages 1013–1031. ECCOMAS, 2016.

[27] Tezzele M., Demo N., Mola A., and Rozza G. An integrated data-driven computational pipeline with model order reduction for industrial and applied mathematics. *Special Volume ECMI, In Press*, 2020.

[28] Sandwell D. T. Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data. *Geophysical Research Letters*, 14(2):139–142, 1987.

[29] Duchon J. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.

[30] Morris A. M., Allen C. B., and Rendall T. C. S. CFD-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation. *International Journal for Numerical Methods in Fluids*, 58(8):827–860, 2008.

[31] Manzoni A., Quarteroni A., and Rozza G. Model reduction techniques for fast blood flow simulation in parametrized geometries. *International journal for numerical methods in biomedical engineering*, 28(6–7):604–625, 2012.

[32] Tezzele M., Ballarin F., and Rozza G. Combined parameter and model reduction of cardiovascular problems by means of active subspaces and POD-Galerkin methods. In Boffi D., Pavarino L. F., Rozza G., Scacchi S., and Vergara C., editors, *Mathematical and Numerical Modeling of the Cardiovascular System and Applications*, volume 16 of *SEMA-SIMAI Series*, pages 185–207. Springer International Publishing, 2018.

[33] Forti D. and Rozza G. Efficient geometrical parametrisation techniques of interfaces for reduced-order modelling: application to fluid-structure interaction coupling problems. *International Journal of Computational Fluid Dynamics*, 28(3–4):158–169, 2014.

[34] Deparis S., Forti D., and Quarteroni A. A rescaled localized radial basis function interpolation on non-cartesian and nonconforming grids. *SIAM Journal on Scientific Computing*, 36(6):A2745–A2762, 2014.

[35] Tezzele M., Demo N., Stabile G., Mola A., and Rozza G. Enhancing CFD predictions in shape design problems by model and parameter space reduction. *Advanced Modeling and Simulation in Engineering Sciences*, 7:40, 2020.

[36] Georgaka S., Stabile G., Star K., Rozza G., and Bluck M. J. A hybrid reduced order method for modelling turbulent heat transfer problems. *Computers & Fluids*, 208:104615, 2020.

[37] Demo N., Tezzele M., and Rozza G. A supervised learning approach involving active subspaces for an efficient genetic algorithm in high-dimensional optimization problems. *arXiv preprint arXiv:2006.07282*, 2020.

[38] Witteveen J. A. S. and Bijl H. Explicit mesh deformation using Inverse Distance Weighting interpolation. In *19th AIAA Computational Fluid Dynamics*. AIAA, 2009.

[39] Ballarin F., D'Amario A., Perotto S., and Rozza G. A POD-Selective Inverse Distance Weighting method for fast parametrized shape morphing. *International Journal for Numerical Methods in Engineering*, 117(8):860–884, 2018.

[40] Abbott I. H. and von Doenhoff A. E. *Theory of Wing Sections, Including a Summary of Airfoil Data*. Dover Books on Aeronautical Engineering Series. Dover Publications, 1959.

[41] Mola A., Tezzele M., Gadalla M., Valdenazzi F., Grassi D., Padovan R., and Rozza G. Efficient reduction in shape parameter space dimension for ship propeller blade design. In Bensow R. and Ringsberg J., editors, *Proceedings of MARINE 2019: VIII International Conference on Computational Methods in Marine Engineering*, pages 201–212, 2019.

[42] Gadalla M., Tezzele M., Mola A., and Rozza G. BladeX: Python Blade Morphing. *The Journal of Open Source Software*, 4(34):1203, 2019.

[43] Ballarin F., Faggiano E., Ippolito S., Manzoni A., Quarteroni A., Rozza G., and Scrofani R. Fast simulations of patient-specific haemodynamics of coronary artery bypass grafts based on a POD-Galerkin method and a vascular shape parametrization. *Journal of Computational Physics*, 315(C):609–628, 2016.

[44] STL (stereolithography) File Format Family. https://www.loc.gov/preservation/digital/formats/fdd/fdd000505.shtml.

[45] Schoenberg I. J. Spline functions and the problem of raduation. *Proceedings of the National Academy of Sciences*, 52(4):947–950, 1964.

[46] Kennicott P.R., IGES/PDES Organization, and U.S. Product Data Association. *Initial Graphics Exchange Specification, IGES 5.3*. U.S. Product Data Association, 1996.

[47] STEP-file, ISO 10303-21. https://www.loc.gov/preservation/digital/formats/fdd/fdd000448.shtml.

[48] Rogers D. F. Chapter 7 – rational b-spline surfaces. In Rogers D. F., editor, *An Introduction to NURBS*, The Morgan Kaufmann Series in Computer Graphics, pages 209–262. Morgan Kaufmann, San Francisco, 2001.

[49] Rogers D. F. and Adams J. A. *Mathematical Elements for Computer Graphics*. McGraw-Hill, Inc., USA, second edition, 1989.

[50] A. Moelgaard. PMM-tests with a model of a frigate class DDG-51. Technical Report 01, PMM-tests with a model of a frigate class DDG-51, 2000.

[51] Paviot T. pythonOCC, 3D CAD/CAE/PLM development framework for the Python programming language, PythonOCC – 3D CAD Python, 2017.

[52] Zancanaro M., Mrosek M., Stabile G., Othmer C., and Rozza G. A hybrid neural network reduced order model for turbulent and geometrically parametrized flows. *Fluids*, 6(8):296, 2021.