MOX–Report No. 05/2014

Free Form Deformation Techniques Applied to 3D Shape Optimization Problems

ROZZA, G.; KOSHAKJI, A.; QUARTERONI, A.

Free Form Deformation techniques applied to 3D shape optimization problems

Anwar Koshakji[1], Alfio Quarteroni[1,2], Gianluigi Rozza[3]

[1] *MOX - Laboratory for Modeling and Scientific Computing,*
Department of Mathematics "F. Brioschi", Politecnico di Milano, Italy
anwar.koshakji@mail.polimi.it

[2] *EPFL, MATHICSE-CMCS, Lausanne, Switzerland*
alfio.quarteroni@epfl.ch

[3] *SISSA mathLab, International School for Advanced Studies, Trieste, Italy*
gianluigi.rozza@sissa.it

Communicated by Alessandro Iafrati

Abstract

The purpose of this work is to analyse and study an efficient parametrization technique for a 3D shape optimization problem. After a brief review of the techniques and approaches already available in literature, we recall the Free Form Deformation parametrization, a technique which proved to be efficient and at the same time versatile, allowing to manage complex shapes even with few parameters. We tested and studied the FFD technique by establishing a path, from the geometry definition, to the method implementation, and finally to the simulation and to the optimization of the shape. In particular, we have studied a bulb and a rudder of a race sailing boat as model applications, where we have tested a complete procedure from Computer-Aided-Design to build the geometrical model to discretization and mesh generation.

Keywords: `Free Form Deformation, Shape optimization, Viscous flows.`

AMS subject classification: `05A16, 65N38, 78M50.`

1. Introduction and motivations.

An important problem in computational science and engineering is to solve partial differential equations in domains involving arbitrary shapes, more particularly in shape optimization [1,2]. In an optimization context, one needs to solve the same equations several times, however in general this procedure may be very expensive for the computational viewpoint and repetitive with several iterations. This calls for an improvement of the approach to the problem, which depends on the selected discretiza-

tion model (e.g. *Finite Elements method* [3]) and on the description and parametrization of the domain geometry and to introduce shape perturbation/deformation.

Let us consider a domain and apply a perturbation to it. Some questions may arise: what would the range of reachable shapes be? Which level of geometrical complexity could be handled? How much would that cost in computational terms? Is there an easy and intuitive method to use? Developing such a strategy can be useful in many applications in shape optimization due to the high flexibility and complexity required for this kind of problems and because of the high number of iterations that can be needed to reach the convergence. Examples of practical uses can be found, for instance, in Aeronautics, for example in the shape optimization of an airfoil, or an entire wing, when attempting to obtain a drag reduction or an efficiency improvement, obeying to some specific optimization laws [1,2,4–8]. Other fields of interest are, e.g., the study of aorto-coronaric bypass configuration [9,10], the optimization of hulls and appendages in naval engineering [11–13], etc.

The aim of this work is to find a simple and powerful method for the management of a large variety of complex and smooth tridimensional deformations, and test it in order to deform some sample geometries. The method should be flexible to describe a wide range of shapes with minimum geometrical constraints. Applications will be made to deform the geometry of a rudder and that of a bulb, two appendages of a sailing yacht, and insert it in a shape optimization process.

The way to describe shape perturbations can be divided in two categories: variational or parametric. In this work we considered only the parametric shape variation: the perturbed domain is a function of a finite number of real parameters. There are several methodologies that will briefly be described below.

Some of the most typical approaches for parametric domains [14], and the motivations that have led us to choose the most appropriate one, are the following:

Basis shapes approach: By using this approach there is a well-chosen set of shape perturbations, which is used to model the geometry through a suitable linear combination.

The shape changes can be expressed as

$$(1) \qquad R = r + \sum_i v_i U_i,$$

where R is the design shape, r is the baseline shape, v_i is the design variable vector and U_i is the design perturbation based on several proposed shapes.

The main problem related to this approach is the definition of a set of basis shapes that are not linearly dependent. Some suggestion for the correct selection of shaped can be found in [15], where Karhunen-Loeve Expansion is adopted for the determination of a basis of independent base shapes. Once independency of the base shapes is obtained, variety of the possible shapes could be quite large. Internal subdivision and structural design in general needs typically to be re-designed once the external shape is changing, and specific design parameters are classically devoted to the definition of the inner structures. As a consequence, basis shapes approach (also called morphing) is also suitable for Multidisciplinary Design Optimization [14].

Discrete approach: This is the most intuitive approach, based on the use of the coordinates of the boundary points as design variables. This method is easy to implement, and the available shapes are limited only by the number of the boundary points. However, it is difficult to maintain a smooth geometry, and to do this, the number of the design variables becomes very large, which leads to a high computational cost and a difficult optimization problem to solve [16]. This approach is commonly applied only when an adjoint formulation of the flow solver is available. Otherwise, the number of design variable is absolutely not feasible for any nearly-real application.

Polynomial and Spline approach: This approach is based on the use of polynomial and spline representations for shape parametrization. This can greatly reduce the total number of design variables. The control of the shape is handled by few special points, called *control points*, which by modifying their positions, the value of the polynomial which describes the curve changes with them. Those methods are very popular in CAD and in design applications in general. According to the different mathematical properties that distinguish one from the other, these are called Bezier, B-splines and NURBS (Non-Uniform Rational B-Spline) curves [17,18]. Those types of curves are well suited for shape optimization, as shown in several works [1,4,5,14,19,20]. Some definitions of such curves are limited (that is not all the geometries can be represented) but they are connected with each other in such a way that one definition goes beyond the limit imposed by the other. NURBS curves are the most general, followed by Rational Bezier curves, which are slightly more limited, then by B-spline curves and finally by Bezier curves.

2. Free Form Deformation.

The problem of defining a solid geometric model of an object bounded by a complex surface has long been identified as an important research problem [21]. *Free Form Deformation* (*FFD*) is a versatile parametrization technique that was originally used with solid modeling system [22]. More recently it has been proposed in a variety of contexts, for example for the parametrization of airfoils and wings in a shape optimization context for potential flows [1], thermal flows [23] and viscous flows [24], for instance for cardiovascular devices [25]. A growing interest in FFD is characterizing naval field [26–28].

While other commonly used techniques directly manipulate the geometrical object at hands, FFD deforms a lattice that is built around the object itself, and consequently, manipulates the whole space in which the object is embedded. Here are some examples found in literature [2,22]. The lattice has the topology of a cube when deforming 3D objects or a rectangle when deforming two-dimensional objects.

One of the advantages concern the choice of the parameters, which is up to the user. Experience [1] shows that FFD is a low dimensional parametrization that gives a good accuracy even with few parameters and it has a good sensitivity. It operates on the whole space that embeds the deformed objects, through the definition of a reference domain and by moving some suitable control points. This allows the user to manipulate the control points of trivariate Bezier[a] volumes.

FFD can treat surfaces of any formulation or degree and it is independent from the domain or the mesh which is used for its discretization, and it features a good trade-off between generality and simplicity.

A distinguishing aspect of this method is that, by deforming the whole volume around (or inside) the object, the computational grids are also being automatically deformed with the object, which is a valuable characteristic for automated design optimization procedures.

FFD can be applied locally or globally and preserves the shape smoothness (and derivative continuity). In the next sections, after formulating FFD method, some examples of applications will be presented and some properties will be recalled [1,2,4,6,14,22,29].

Another benefit of using FFD is that the computation may be subdivided into an *offline* stage, which is more time consuming, and an *online* part, which can be computed several times once the product of the offline part is stored and it is much cheaper. This fact matches the need of the model reduction method, such as the *Reduced basis method*

---

[a]Also B-splines or NURBS can be used to produce the deformation volume [14].

(RB) [23–25,30–32]. Without giving details, RB method is based on an offline and on an online part, as well. So the two methods can be connected, remarkably improving the computational time and the efficiency of the computation.

2.1. *Formulation.*

A suitable illustrative example is that of a rectangle (in 2D) or a parallelepiped (in 3D) of transparent, flexible plastic in which an object or several objects are embedded, which are intended to be deformed [2]. Here we follow the tridimensional case.

After defining a reference domain Ω_0 and a subset that we wish to perturb $D_0 \subset \Omega_0$, a differentiable and invertible map is introduced $\Psi :$ $(x_1, x_2, x_3) \to (s, t, p)$, so that $\Psi : (D) \to (0, 1) \times (0, 1) \times (0, 1)$. The FFD is defined in the reference coordinates (s, t, p) of the unit cube. Let us select a regular grid of unperturbed control points $\mathbf{P}^0_{l,m,n}$, where $l = 0, \dots, L$, $m = 0, \dots, M$ and $n = 0, \dots, N$ so that

$$(2) \qquad \mathbf{P}^0_{l,m,n} = \begin{bmatrix} l/L \\ m/M \\ n/N \end{bmatrix}.$$

A parameter vector $\boldsymbol{\mu}_{l,m,n}$ is introduced, whose dimension is $3 \times (L+1) \times (M+1) \times (N+1)$, because for each control point we consider the possibility to move in three different directions (s, t and p). In Figure 1 we can show how they are distributed.
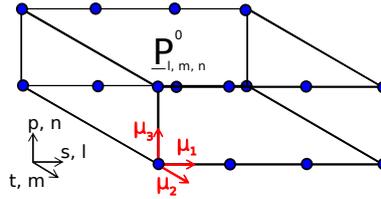

Figure 1. Unperturbed control points and parameters vector.

Each control point is perturbed by the corresponding value of the parameters vector:

$$(3) \qquad \mathbf{P}_{l,m,n}\left(\boldsymbol{\mu}_{l,m,n}\right) = \mathbf{P}^0_{l,m,n} + \boldsymbol{\mu}_{l,m,n}.$$

Then the parametric domain map is constructed $\mathbf{T} : D_0 \to D(\boldsymbol{\mu})$ as

$$(4) \qquad \mathbf{T}\left(\Psi\left(\boldsymbol{x}\right); \boldsymbol{\mu}\right) = \Psi^{-1}\left(\sum_{l=0}^{L}\sum_{m=0}^{M}\sum_{n=0}^{N} b^{L,M,N}_{l,m,n}\left(s, t, p\right) \mathbf{P}_{l,m,n}\left(\boldsymbol{\mu}_{l,m,n}\right)\right),$$

where

$$b_{l,m,n}^{L,M,N}(s,t,p) = b_l^L(s)\, b_m^M(t)\, b_n^N(p) = \dots$$

(5)
$$\dots = \binom{L}{l}\binom{M}{m}\binom{N}{n}(1-s)^{(L-l)}\, s^l\, (1-t)^{(M-m)}\, t^m$$
$$\cdot\, (1-p)^{(N-n)}\, p^n,$$

are tensor products of the 1-d Bernstein basis polynomials

(6)
$$b_l^L(s) = \binom{L}{l}(1-s)^{(L-l)}\, s^l,$$
$$b_m^M(t) = \binom{M}{m}(1-t)^{(M-m)}\, t^m,$$
$$b_n^N(p) = \binom{N}{n}(1-p)^{(N-n)}\, p^n,$$

defined on the unit square with local variables $(s,t,p) \in [0,1] \times [0,1] \times [0,1]$, and the function Ψ maps $(x_1, x_2, x_3) \mapsto (s,t,p)$.

As mentioned, in order to effectively calculate the global map $\mathbf{T}$ there are two parts: one *offline*, that is the precomputation of the transformation by the use of a symbolic expression, and the other is *online*, that is the evaluation of the function for the parameters and the coordinates of the real system. This second part is very cheap, even in the 3D case. So once the offline part is completed, that is the part which takes the majority of the cost, the map $\mathbf{T}$ is calculated, so it is enough to evaluate it. For optimization problems, which implies to reiterate the computation several times, it may be an excellent tool.

2.2. *FFD properties and practical issues.*

FFD is a method that involves all the domain, as a lattice created by Bernstein polynomials where all the internal objects are deformed and follow the deformation rule imposed by them. In fact, by just moving one control point in one direction implies the deformation of all the domain, assuring a smooth and continuous deformation, no matter how complex this could be.

Since Bernstein polynomials vanish on the boundary, all the deformation takes place only inside the boundary domain.

Another thing to highlight is that the use of FFD also reduces the number of shape parameters: in [23] it has been estimated that compared to a small perturbation approach by moving individual mesh nodes, a very substantial reduction (up to 2 orders of magnitude) in the number of geometric

parameters can be achieved. This is another strong point in favor of this method, which makes it still more complete and efficient at the same time.

Last but not least, FFD can be applied also locally, in order to deform just a part of the domain and to focus only on it. Following this idea, multiple FFD blocks can be chained together to enhance the deformation of an object, the control points where the two FFD blocks are joined cannot be deformed. FFD can therefore be adapted according to the type of problem at hands, yielding a very versatile method.

FFD may also allow to maintain the same mesh during the optimization process, due to the fact that the smooth deformation involves all the domain where the FFD lattice is defined, including the mesh points. This is another important feature, because one does not need to remesh for every iteration, but the new mesh follows the deformation. In other words, since FFD is a technique to deform the space, it can be used to deform the mesh and the shape simultaneously [6]. However, one should proceed carefully, in order to ensure that a smooth deformation is imposed, which does not generate overlapping effects in the mesh. The smoothness of the deformation is ensured inside the lattice, since deformation is ruled by Bernstein polynomials. That is why we have chosen to operate just with the internal point of the FFD domain, as we have mentioned.

Another feature that can be taken into consideration is the possibility of implementing a rotation of the FFD lattice. This could be helpful for example in case the object to be deformed is rotated over a certain angle or it is necessary to deform this object in order to maintain a symmetry not aligned with the orthogonal axes. The mentioned rotation can be obtained by applying an additional rotational matrix $\mathbf{R}$ to compute the global map $\mathbf{T}$.

In order to achieve the purposes of this work and to apply an efficient method of deformation and use it in an automatic shape optimization process, it is necessary to have the appropriate tools to work with and have efficient interactions and communications among them. So there is the necessity to create an initial geometry, define the problem we want to solve and implement by the FFD method. For this reason, many options and softwares have been used and explored, and it has been necessary to study a connection between them, so that they can interact and communicate with proper interfaces and consistent data[b].

---

[b]A CAD software (Computer-Aided Design) has been necessary for the setting up and design of the models and their geometries that we want to optimize. For this purpose, many CAD softwares are available, such as *Rhino* [33], *AutoCad* [34]. *SOLID-WORKS* [35], used in this work, is the starting point, where all the initial design decisions are taken. Secondly, the geometries are exported in a suitable format and imported into

2.3. *Geometrical models.*

A complete underwater part of a sailboat[c], with all its appendages, is shown in Figure 2.
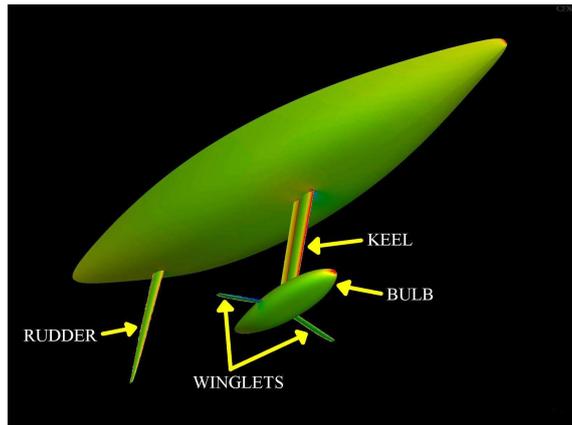

Figure 2.   Bulb and appendages of a sailboat.

In particular, in this work we have focused on the shape optimization of two: the bulb and the rudder, which have both been created using SOLID-WORKS. First in Figure 3(a) the bulb is presented, which has a much less complex shape than the shape of the rudder (Figure 3(b)). It is just an initial surface, which will be given as an initial guess in the optimization process. Keel and winglets (see Figure 2) are not present because usually a shape optimization of a bulb under a uniform flow includes just the geometry of the bulb itself[d] [12,36].

In order to geometrically create the rudder, a mean spanwise airfoil NACA 63012 [37], a root cord of 0.5 *m* and a total length of 3.02 *m* have been used.

---

the equation solver program. *COMSOL Multiphysics* has been used to define the problem and to solve it at every iteration of the optimization process.

[c]Image courtesy of CMCS (Chair of Modelling and Scientific Computing - EPFL - Lausanne).

[d]Winglets are largely influencing the performances in oblique flow, as well as they are also important in the symmetric case, but a simplification is required in order not to increase too much the computational cost. We think that this is a reasonable choice for the preliminary investigation, but not the best practical approach, since the real problem includes winglets and fin.
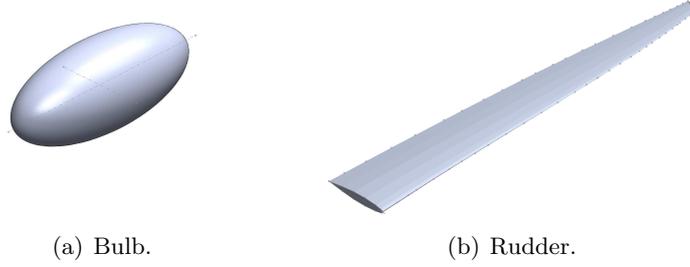
(a) Bulb. (b) Rudder.

Figure 3.   Initial geometrical models.

3. Mathematical and numerical formulation of the model problems.
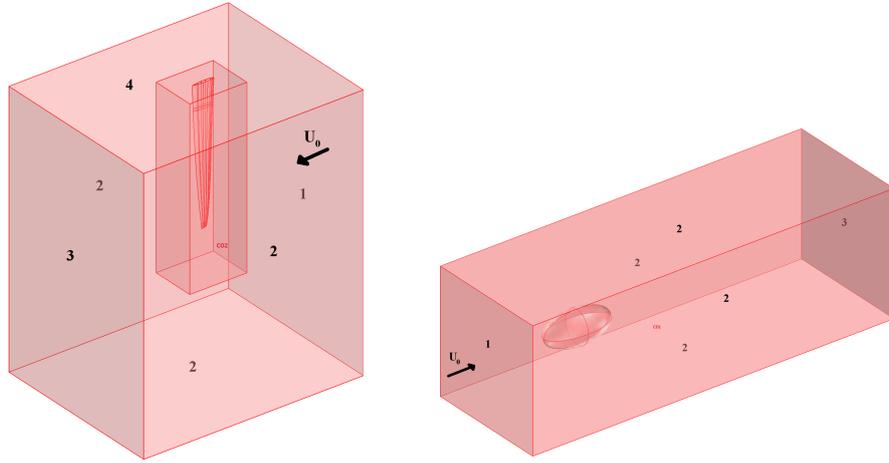
3.1. *Problem definition.*

Given a domain $\Omega \subset \mathbb{R}^3$ (the complementary of the region occupied by the sailing boat), the equations considered are the incompressible steady *Navier-Stokes equations*, for a viscous Newtonian fluid [3,38–40]:

$$(7) \quad \begin{cases} (\mathbf{u} \cdot \nabla)\,\mathbf{u} + \nabla p - \nabla \cdot \left[\nu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \right] = \mathbf{f}, & \mathbf{x} \in \Omega \\ \nabla \cdot \mathbf{u} = 0, & \mathbf{x} \in \Omega, \end{cases}$$

where ρ is the fluid density, which is constant, $\mathbf{u}$ is the velocity field of the fluid, p is the pressure divided by the density, ν is the kinematic viscosity, $\nu = \frac{\mu}{\rho}$, where μ is the dynamic viscosity and $\mathbf{f}$ is a forcing term per mass unit. The first equation of the system is the *momentum equation*, the second is the conservation of mass equation, which is known also as *continuity equation*. In Figures 4(a) and 4(b) we show the boundary conditions and on which face of the domain they have been imposed. The numbers on the surface correspond to a cruising condition:

1. Inlet: uniform velocity $\mathbf{u} = -U_0 \mathbf{n}$;
2. Open boundary: normal stress $\left[-p\mathbf{I} + \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \right] \mathbf{n} = 0$;
3. Outlet: pressure $p = p_0$ and no viscous stress $\mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \mathbf{n} = \mathbf{0}$;
4. Wall: no slip $\mathbf{u} = \mathbf{0}$;

where $\mathbf{n}$ is the normal to the face of the domain considered, $\mathbf{u}$ is the velocity vector and p is the pressure. As previously introduced, μ is the dynamic viscosity, U_0 and p_0 are the values of the velocity and of the pressure in the unperturbed field, respectively.

(a) Boundary conditions for the rudder problem.

(b) Boundary conditions for the bulb problem.

Figure 4.   Domain definition and boundary conditions.
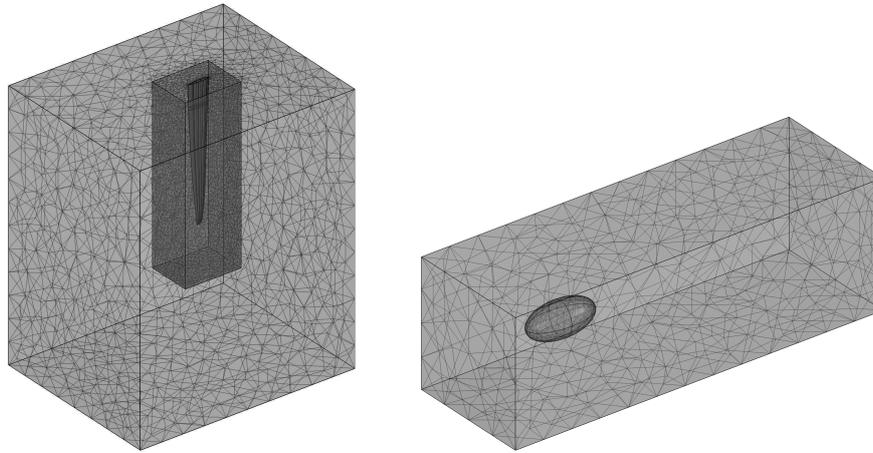
(a) The rudder and the mesh of the domain (number of elements: 222138).

(b) The bulb and the mesh of the domain (number of elements: 8904).

Figure 5.   Domain definition and boundary conditions.

By writing the weak formulation and using the Galerkin finite element method, we obtain:

$$\begin{cases} \mathbf{A}\mathbf{U} + \mathcal{N}\left(\mathbf{U}\right) + \mathbf{B}^T\mathbf{P} = \mathbf{F} \\ \qquad\qquad\quad \mathbf{B}\mathbf{U} = 0, \end{cases}$$

where all matrices depend on appropriate test functions.

3.1.1. *Cost functionals.*

We have chosen to minimize the following functionals: either

$$
(8) \qquad J_b\left(\boldsymbol{\mu}\right) = \frac{D\left(\boldsymbol{\mu}\right)}{D_0},
$$

or

$$
(9) \qquad J_r\left(\boldsymbol{\mu}\right) = \frac{1}{2}\left(\frac{D\left(\boldsymbol{\mu}\right)}{D_0} + \frac{E_0}{E\left(\boldsymbol{\mu}\right)}\right),
$$

where J_b is the functional for the bulb, J_r is the functional for the rudder, $E = L/D$ is the efficiency where L and D are, respectively, the lift of the rudder and the drag force obtained by solving the Navier-Stokes equations, and D_0 and E_0 are reference quantities (generally the values of the first step). D and L are obtained by making a pressure integration on the rudder surface (for the bulb it is done only for the drag) in a streamwise and spanwise direction, respectively.

3.2. *Optimization algorithm.*

The next step is the description of the iterative optimization algorithm. In literature many optimization methods have been proposed [16,41]. Among them, the most common ones are the gradient-like method [42,43], genetic algorithms [44]. Gradient-like methods require the gradient of the scalar cost function and constraints (dependent variables), respecting the shape design (independent) variables. The problem of these kinds of methods is that they may converge to local optimum and not to the global one. Besides, Genetic Algorithms (GAs) have proven their strength against local limits, however they may require a very high number of configurations evaluation to converge. In our case, the cost of GA methods is still not affordable, thus we will focus on the gradient methods, preferring also a deterministic approach to the problem[e].

For a smooth constrained problem, let g and h be vector functions representing all inequality and equality constraints respectively. Our opti-

---

[e]The built-in MATLAB function *fmincon* comes to our purpose [45]. It is a function based on gradient-like method, which finds a constrained minimum of a scalar function of several variables, starting from an initial estimate.

mization process can be written as

$$\min_{\boldsymbol{\mu}} J\left(\boldsymbol{\mu}\right),$$

(10)
$$\text{subject to } g\left(\boldsymbol{\mu}\right) \leq 0,$$

$$h\left(\boldsymbol{\mu}\right) = 0,$$

where $\boldsymbol{\mu}$ is the set of FFD parameters defined in Section 2.1, the design variables which are correlated with the control points displacement. A priori, one has to decide which parameters to choose, according to the direction of the deformation, and this is absolutely arbitrary.

Then, we define the Lagrangian function as

(11) $$L\left(\boldsymbol{\mu}, \boldsymbol{\lambda}\right) = J\left(\boldsymbol{\mu}\right) + \sum \lambda_{g,i} g_i\left(\boldsymbol{\mu}\right) + \sum \lambda_{h,i} h_i\left(\boldsymbol{\mu}\right),$$

where $\boldsymbol{\lambda}$ is the Lagrange multiplier vector of $\boldsymbol{\lambda}_g$ and $\boldsymbol{\lambda}_h$. Its length is the total number of constraints. The Hessian of this function is shown below

(12) $$W = \nabla_{\mu\mu}^2 L\left(\boldsymbol{\mu}, \boldsymbol{\lambda}\right) = \nabla^2 J\left(\boldsymbol{\mu}\right) + \sum \lambda_{g,i} \nabla^2 g_i\left(\boldsymbol{\mu}\right) + \sum \lambda_{h,i} \nabla^2 h_i\left(\boldsymbol{\mu}\right),$$

where $\nabla_{\mu\mu}^2$ is the Laplacian in respect to vector $\boldsymbol{\mu}$. The function *fmincon* uses a *Sequential Quadratic Programming* (SQP) method, that is one of the most popular and robust algorithms for nonlinear continuous optimization [46,47], and it is appropriate for small or large problems. The method solves a series of subproblems designed to minimize a quadratic model of the objective function using a linearization of the constraints. A non-linear program in which the objective function is quadratic and the constraints are linear is called a Quadratic Program (QP). An SOP method solves a QP at each iteration. In particular, if the problem is unconstrained, then the method reduces to *Newton's method* [48] to find a point where the gradient of the objective vanishes. If the problem only equality constraints, then features the method is equivalent to applying *Newton's method* to the first-order optimality conditions (or *Karush-Kuhn-Tucker* (KKT) conditions [46]) of the problem.

In order to define the k-th subproblem, both the inequality and equality constraints have to be linearized. If $\mathbf{p} = \boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k$, where k is the iteration counter, we obtain the local subproblem

$$\min \frac{1}{2}\mathbf{p}^T W_k \mathbf{p} + \nabla J_k^T \mathbf{p},$$

(13)
$$\text{subject to } \nabla h_i\left(\boldsymbol{\mu}_k\right)^T \mathbf{p} + h_i\left(\boldsymbol{\mu}_k\right) = 0,$$

$$\nabla g_i\left(\boldsymbol{\mu}_k\right)^T \mathbf{p} + g_i\left(\boldsymbol{\mu}_k\right) \geq 0,$$

A QP method is now used to solve this problem [46].

3.3. *Solvers.*

For the (symmetric) bulb problem a direct solver (PARDISO [49]) has been used, while the iterative solver BiCGStab [43] has been used for the rudder.

All the operations involved in the optimization process are summarized in Figure 6, starting from the definition of the model problem and the design variables, through the choice of the cost functional and the iterations of the optimization process, till the final optimized shape.

Note that during this optimization procedure, there is no need to regrid after deformation, since the deformation applied from the FFD techniques involves not only the geometry but also the mesh itself. When limiting to small deformation fields, the mesh continues to maintain its validity.
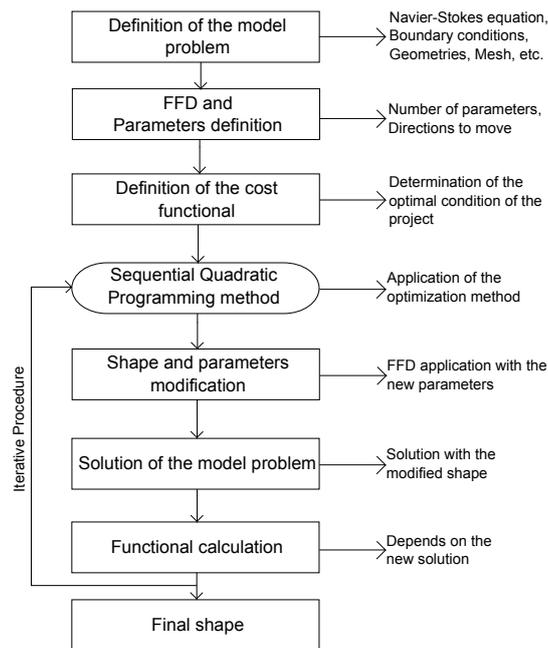

Figure 6.   Scheme for the shape optimization process.

4. Simulations and results.

4.1. *Bulb.*

First we present the results concerning the bulb. In the simulation, the cost functional described in previous section has been used. A FFD into a bounding box has been applied (see Figures 7 and 8). The total number of control points is 343 ($L = 6$, $M = 6$ and $N = 6$, referred to x, y, and z direction respectively), however the ones actively involved in the simulation are only 12, corresponding to the use of 20 parameters.

The following constraints have been imposed:

- Concerning the volume V we allow a variation up to 20% of its initial value. This is to avoid the most obvious condition of minimum resistance, when the bulb degenerates to a point in the space.
- In order to maintain the symmetry along the z direction additional constraints have been imposed to displacements of control points indicated as A and B in Figure 8, such that $\mu_A = -\mu_B$. Instead, no constraints to the control points have been imposed to maintain the symmetry along the y direction, to test whether the result still remains symmetric without explicitly imposing the symmetry (for consistency and generalization purposes).
- The parameters vary between $[-80\%, 80\%]$ (expressed in percentage of the chord) to maintain the deformation contained and to avoid mesh degeneracy.

In Figure 8 the displacements of the control points chosen as parameters for the optimization are shown.
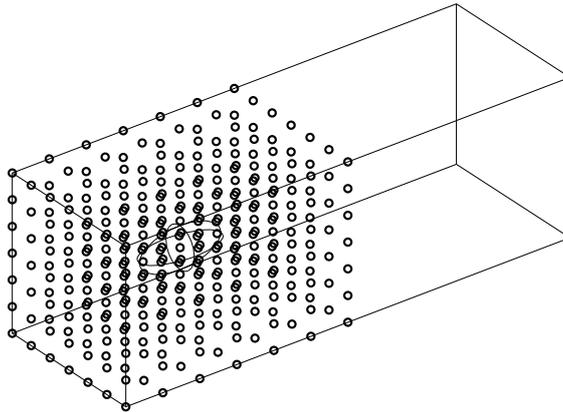

Figure 7.   Domain where the bulb is inserted and definition of the FFD bounding box.
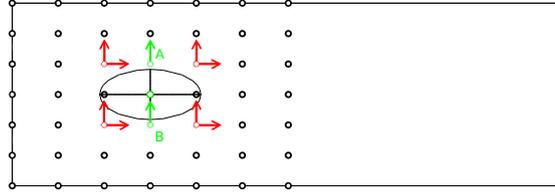
Figure 8.   Lateral view of the domain and the FFD bounding box.

Results after the optimization process follow. For a better view we present just the flow field belonging to the XY plane, being a symmetric flow. In Figure 9 the initial pressure field around the undeformed bulb is shown, while in Figure 10 we report the pressure field obtained after the shape optimization.
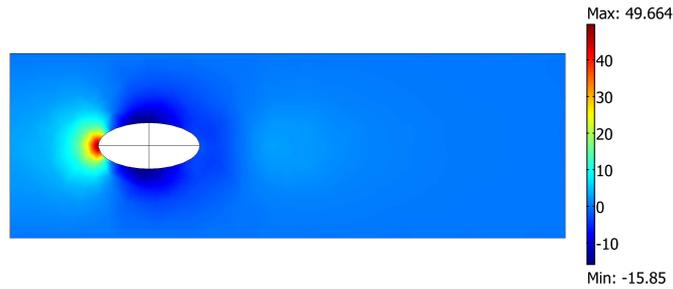

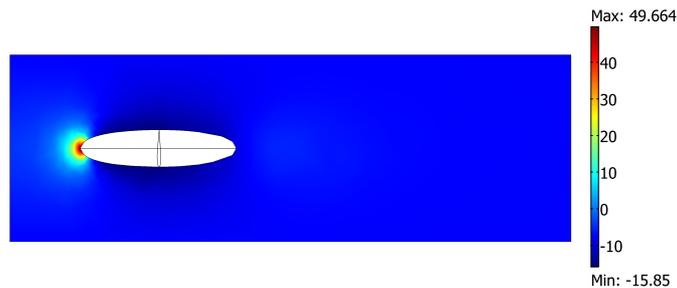Figure 9.   Pressure field of the initial shape of the bulb in plane XY (Pa).

Figure 10.   Pressure field around the optimized bulb in plane XY (Pa).

In Figure 11 the initial velocity field around the initial shape of the bulb is shown, while in Figure 12 velocity field obtained after the shape optimization is represented.

As it can be noticed, the optimization succeeded to reduce the wake past the bulb. In fact, the drag force D is composed by two contributions: one given by skin friction force and the other given by the pressure force [38].
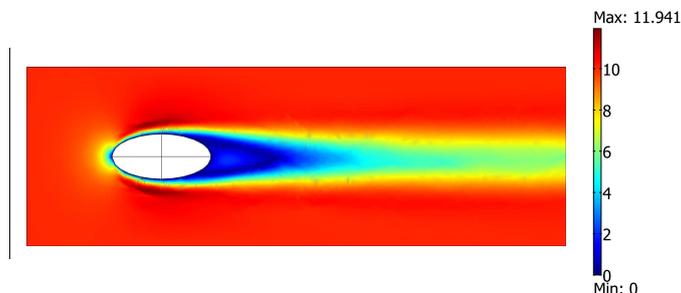
Figure 11.   Velocity field of the initial shape of the bulb in plane XY (m/s).
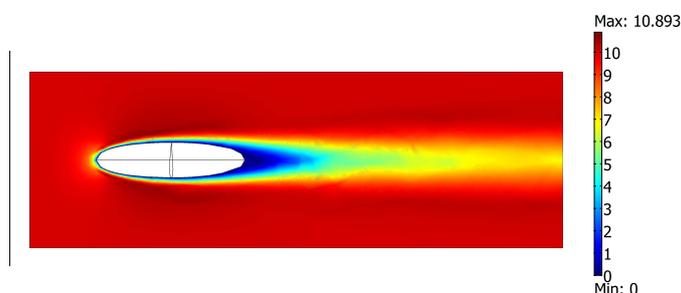

Figure 12.   Velocity field around the optimized bulb in plane XY (m/s).

Depending on the shape of the object, besides of course on the Reynolds number, one contribution becomes more important than the other one. In the case of a bulb, or a blunt body, and in presence of a sufficient high Re, as in our case the major contribution derives from the pressure force. By trying to contain this contribution, the frontal area of the bulb is reduced, as we expected, and it becomes more and more similar to an airfoil, where the skin friction drag, or the viscous one, is predominant.

In Table 1 the values of the parameters and of the cost functional J obtained as results, which are defined in Section 3.1.1, are reported. To recall, D_0, V_0 and J_0 are the drag, the volume and the cost functional, respectively, referred to the undeformed bulb, while D, V and J are the ones obtained at the end of the optimization. The percentage gain of drag decrease is denoted with $\%\Delta$.

Table 1.   Value obtained before and after the shape optimization for the bulb.

D_0 [N]	D [N]	V_0 $\left[m^3\right]$	V $\left[m^3\right]$	J_0	J	$\%\Delta$
1.389	1.005	0.7156	0.5725	1	0.724	27.6

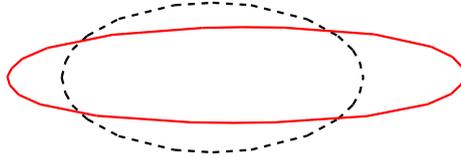In Figure 13 the initial and final shape are compared.

Figure 13.   Visualization of the initial and final shape of the bulb.

As it can be observed, there is an important reduction of the drag with the new shape, a gain of 27.6% respect to the initial shape. The final volume is the 80% of the initial one, and this indicates that the optimization has stopped because it has reached the limit of volume reduction.

4.2. *The rudder.*

We now show the results for the rudder. The simulations have been more difficult, due to the fact that the rudder has a more complex and refined geometry. It has also been rotated by an angle of $\alpha = 5°$ with respect to the flow field around the z axis. Also in this case, a local FFD has been applied, as shown in Figures 14 and 15.

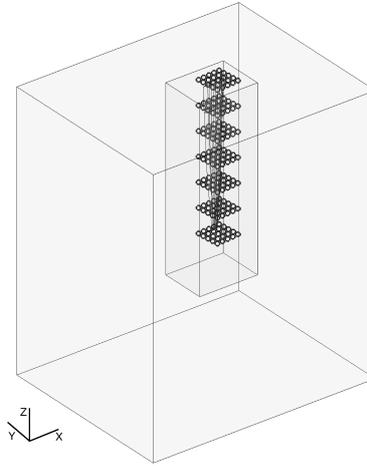Figure 14.   The rudder and the subdomains where the rudder is placed, with the lattice of the rotated FFD bounding box.

As we anticipated in Section 2.3, the mean airfoil is a NACA 63012 profile, which is a symmetric one. To maintain this symmetry, a constraint on the displacements of the parameters has been imposed, such that all the displacements are proportionally connected along the z axis. Thus the constraints imposed are:

- As in the case of the bulb, the volume of the rudder cannot diminish more than 20% of its initial value (for demonstration purposes).
- $\mu_{i_{zk}} = \frac{k}{5}\mu_{i_{z5}}$, where i is the i-th parameter, k is the layer of each plane XY formed by the control points considered along the z axis. In this case we have considered 5 layers, so k can vary from 1 to 5, where 5 is the highest XY plane of control points taken into consideration (see Figure 15). So the displacements at the inferior layers will be proportional to the highest one.
- At every level k, it has been imposed that $\mu_A = -\mu_B$ (Figure 15) to maintain the symmetry in the plane of the airfoil.
- Since deformations are very delicate, so the range of the parameters is restricted to $[-200\%, 200\%]$ (expressed in percentage of the chord).

Since the rudder is rotated, it has been necessary to rotate also the local FFD, in order to respect the symmetry condition, as shown in Figure 15. The lattice has 175 control points ($L = 4$, $M = 4$ and $N = 6$, referred to x,
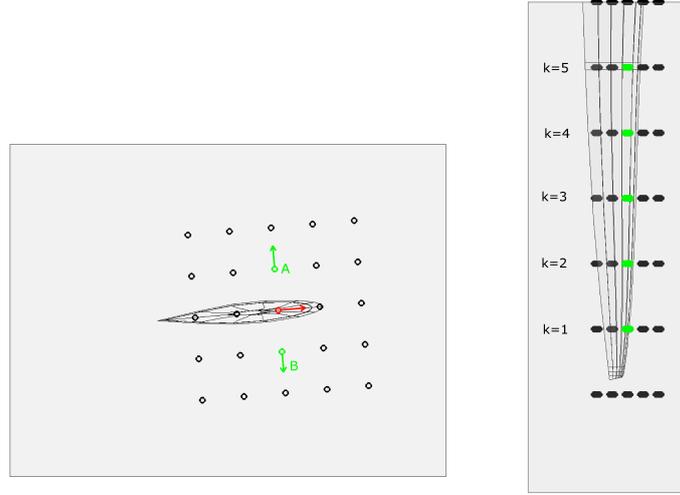

Figure 15.   An upside and a lateral view of the rudder in the XY plane and in the XZ plane respectively and the displacements considered.

y, and z direction respectively), and the ones chosen for the optimization and the parameters are 15, which are all indicated in Figure 15. In the next figures the shape of the rudder is shown from plane XZ, regarding the pressure (Figures 16(a) and 16(b)) and the velocity field (Figures 17(a) and 17(b)) before and after the optimization process. Results refer to the situation in the center of the domain, that is taken at $y/2$.

The shape is modified, however the deformation is not large enough to appreciate the entity of the variations. In Figure 18 we show an am-
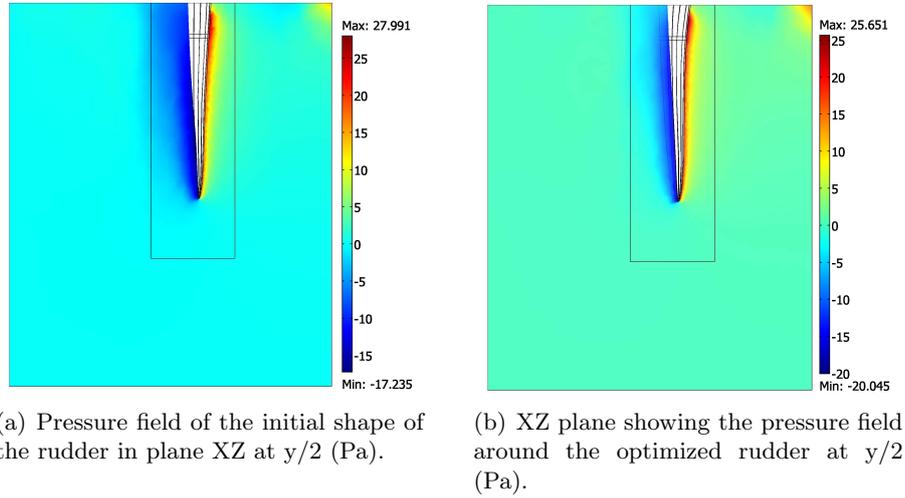
18

(a) Pressure field of the initial shape of the rudder in plane XZ at y/2 (Pa).

(b) XZ plane showing the pressure field around the optimized rudder at y/2 (Pa).

Figure 16. Pressure fields around the initial and the final shape of the rudder.

(a) Velocity field of the initial shape of the rudder in plane XZ at y/2 (m/s).

(b) XZ plane showing the velocity field around the optimized rudder at y/2 (m/s).
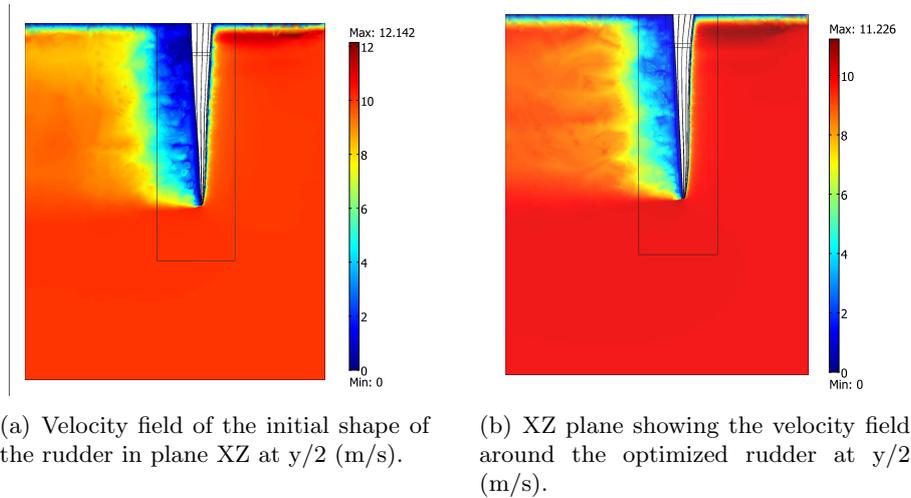
Figure 17. Velocity fields around the initial and the final shape of the rudder.

plification of twice the deformation obtained, just to give an idea of its magnitude. The values of the physical variables, which have been defined in Section 3.1.1, are shown in Table 2. D_0, E_0, V_0 and J_0 are the variables referred to the initial shape of the rudder, then D, E, V and J are the ones referred to the final optimized shape, which are the drag, the efficiency and the volume respectively (see Section 3.1.1). $\%\Delta_{\text{tot}}$ is the relative gain obtained with the new shape, which includes the contributions

Figure 18.   XZ plane with the amplified deformed rudder.

of the combination of drag D and efficiency E. The optimization process

Table 2.   Value obtained before and after the shape optimization for the rudder.

D_0 [N]	D [N]	E_0	E	V_0 [m^3]	V [m^3]	J_0	J	$\%\Delta_{tot}$
2.993	2.915	2.559	3.7464	0.0298	0.0207	1	0.829	17.1

has stopped after reaching the maximum range value of the parameters, which was imposed on the highest layer ($k = 5$). However, even if the deformation involved on the final form of the rudder is small, the wake after it looks smaller than the original one and there is a total gain of 17.1%, which corresponds to a 2.6% reduction in drag and a 31.7% gain in efficiency E, maintaining the construction symmetries.

To sum up, we can conclude that FFD versatility suits well the optimization process, adapting without problems to the different kinds of geometries and constraints, improving the performance of the object taken into consideration and it may be considered a valid and efficient alternative approach with respect to more classical shape optimization methods. The major cost of the computational process is the solution of the Navier-Stokes equations, and it can become very high by the use of more FFD parameters, since the optimization process described in Section 3.2 is solved many times before converging. The use of some reduced order modelling techniques, such as reduced basis method [23,30,31], can help diminishing the computational cost of the solution of the Navier-Stokes equations and allow to pursue both geometrical and computational cost reduction, since FFD is able to manage shape optimization with a reduced number of parameters.

However, the test cases have been considered with the aim of describing a shape optimization design process of a generic CAD object by adapting

the model equations and the quantities appearing in the cost functional and constraints, which can be weighted differently according to the optimization that one wants to pursue. In Figure 19 the comparison between the two geometries is shown: the initial shape and the final one (black dashed line and red line, respectively).

Figure 19.   Rudder initial and final shape comparison.

5. Conclusions.

The FFD proved to be a powerful and efficient parametrization method that could be used in several applications, such as the shape optimization of a wing or an airfoil, a bypass conduct or a part of a sailing boat. In this work, FFD has been tested on 3D examples. We have considered two shape optimization processes dealing with a bulb and a rudder of a yacht, respectively. The FFD method has been applied around a bounding box in order to have a better sensitivity of the deformation around the object. Moreover, regarding the rudder, FFD has also been rotated/distorted in order to maintain the symmetry constraint of the deformation along its spanwise direction. One aspect that could be tested in order to improve the control of the deformation is to subdivide the domain into several FFD settings, so that we may have different deformations sets/regions.

A distinguishing feature of the FFD method is that the deformation involves also the mesh defined inside the lattice of points (bounding box), and for small and smooth deformation, there is no particular need to make

a new mesh at each iteration of the shape optimization problem. Thus FFD is mesh independent and also independent of geometry and even the PDE model to which it is applied. FFD can also be efficiently used in the preliminary design phase, for instance of a complete aircraft, more in general in a multidisciplinary shape optimization problem [30,50].

These features make FFD a very flexible and efficient method. The results presented in Section 4 show that, applying the Navier-Stokes equations to solve the flow in a cruising condition, a new optimized shape is obtained both for the bulb and for the rudder, with an improvement of the fluid dynamics performances and indexes related with state variables chosen properly to minimize/maximize, which is drag and a combination of drag and efficiency, respectively, obtaining as much as the 27.6% in drag reduction for the bulb and a 17.1% of improvement of the combination of drag reduction and efficiency for the rudder.

The choice of the degrees of freedom of the admissible deformations and the number of the parameters are all up to the user. One aspect that can make the object of further investigation is the setup of a method that, by identifying which shapes need to be deformed, allows the choice of control points to improve the shape optimization with the least number and maximize the efficiency of the deformation. This could reduce even more the computational costs needed for the shape optimization process, which is determined by the number of chosen variables. To reach our goal we have developed a platform by combining several capabilities already available in order to combine different tools for geometrical modelling, shape variation, numerical simulation and optimization [51].

Anyway, the cost of the optimization by solving Navier-Stokes equations discretized by finite elements at every iteration and for every design variables may become prohibitive. In this perspective, an aspect of interest could be to couple FFD method with reduced order modelling techniques. This will simplify the complexity problem and gain even more in efficiency of computational performance.

We underline that FFD is seen as an alternative method for shape optimization. See [51,52] for some classical results in shape optimization rebuilt with FFD and their comparison with classical techniques. For more advanced uses of FFD we recall some recent studies in [50]. In the aerodynamics field, an alternative parametrization named *MASSOUD* (*Multidisciplinary Aero/Struc Shaper Optimization Using Deformation*), a sort of evolution of the FFD, has been proposed by Samareh in [53], which modifies the FFD method in order to parameterize the shape perturbations rather than the geometry itself. This could lead to a generalized FFD approach.

REFERENCES

1. E. I. Amoiralis and I. K. Nikolos, Freeform Deformation Versus B-Spline Representation in Inverse Airfoil Design, *J. Comput. Inform. Sci. Eng.*, vol. 8, pp. 024001–1–024001–13, June 2008.

2. T. Lassila and G. Rozza, Parametric free-form shape design with PDE models and reduced basis method, *Comp. Meth. Appl. Mech. Eng.*, vol. 199, pp. 1583–1592, 2010.

3. A. Quarteroni, *Numerical Models for Differential Problems*, vol. 2 of *MS&A*. Milano: Springer, 2009.

4. M. Andreoli, A. Janka, and J. A. Désidéri, Free-form-deformation parameterization for multilevel 3D shape optimization in aerodynamics, *INRIA Research Report no. 5019*, November 2003.

5. J. A. Désidéri, R. Duvigneau, B. Abou El Majd, and Z. Tang, Algorithms for efficient shape optimization in aerodynamics and coupled disciplines, in *42nd AAAF Congress on Applied Aerodynamics*, (Sophia-Antipolis, France), March 2007.

6. R. Duvigneau, Adaptive parameterization using Free-form deformation for aerodynamic shape optimization, *INRIA Research Report RR-5949*, July 2006.

7. A. Jameson, Optimum Aerodynamic Design using CFD and Control Theory, in *AIAA Paper 95-1729*, (12th AIAA Computational Fluid Dynamics Conference), 1995.

8. A. Jameson, N. Pierce, and L. Martinelli, Optimum Aerodynamic Design using the Navier-Stokes Equations, *AIAA Paper 97-0101*, 1997.

9. A. Quarteroni and G. Rozza, Optimal control and shape optimization in aorto-coronaric bypass anastomoses, *Mathematical Models and Methods in Applied Sciences (M3AS)*, vol. 13, no. 12, pp. 1801–23, 2003.

10. G. Rozza, On Optimization, Control and Shape Design for an arterial bypass, *International Journal Numerical Methods for Fluids*, vol. 47, no. 10-11, pp. 1411–1419, 2005. Special issue for ICFD Conference, University of Oxford.

11. M. Lombardi, N. Parolini, G. Rozza, and A. Quarteroni, *Numerical simulation of sailing boats: dynamics, FSI, and shape optimization*, vol. 66, ch. 15, pp. 339–378. G. Buttazzo and A. Frediani ed., Optimization and its Applications series, Springer, 2012.

12. A. Manzoni, Ottimizzazione di forma per problemi di fluidodinamica: analisi teorica e metodi numerici, Master's thesis, Mathematical Eng., Politecnico di Milano, 2008.

13. N. Parolini and A. Quarteroni, Mathematical models and numerical simulations for the America's Cup, *Comp. Meth. Appl. Mech. Eng.*, vol. 194, pp. 1001–1026, 2005.

14. J. A. Samareh, A survey of shape parameterization techniques, in *CEAS AIAA ICASE NASA Langley International Forum on Aeroelasticity and Structural Dynamics*, June 1999.

15. M. Diez, D. Peri, F. Stern, and E. F. Campana, An Uncertainty Quantification approach to assess geometry-optimization research spaces through Karhunen-Loeve Expansion, *UQ '12: SIAM Conference on Uncertainty Quantification*, vol. SIAM Publications, 2012, Philadelphia (USA).

16. B. Mohammadi and O. Pironneau, *Applied Shape Optimization for Fluids*. Oxford: Oxford University Press, 2001.

17. R. A. Adams and C. Essex, *Calculus: a complete course*. Canada: Pearson Education, 7 ed., 2009.

18. C.-K. Shene, "CS3621 Introduction to Computing with Geometry Notes." Michigan Technological University, 1997-2008.

19. A. Buffa, G. Sangalli, and R. Vazquez, Isogeometric analysis in electromagnetics: B-splines approximation, *Comp. Meth. Appl. Mech. Eng.*, vol. 199, no. 17-20, pp. 1143–1152, 2010.

20. J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, *Isogeometric analysis: toward integration of CAD and FEA*. Jhon Wiley & Sons, October 2009.

21. M. Botsch and L. Kobbelt, An intuitive framework for real-time freeform modeling, *ACM Trans. on Graphics*, vol. 23, no. 3, pp. 630–634, 2004. Proc. ACM SIGGRAPH.

22. T. W. Sederberg and S. R. Parry, Free-form deformation of solid geometric models, in *Proceedings of SIGGRAPH - Special Interest Group on GRAPHics and Interactive Techniques*, vol. 20, pp. 151–159, August 1986.

23. G. Rozza, T. Lassila, A. Manzoni, E. Ronquist (ed.), and J. Hesthaven (ed.), Reduced basis approximation for shape optimization in thermal flows with a parametrized polynomial geometric map, in *Spectral and High Order Methods for Partial Differential Equations, Lectures Notes in Comp. Science and Engineering* (S. Heildeberg, ed.), vol. 76, pp. 307–315, 2010. Selected papers from the ICOSAHOM 09 Conference, NTU Trondheim, Norway, 22-26 June 2009.

24. G. Rozza, A. Manzoni, J. Pereira (ed.), and A. Sequeira (ed.), Model order reduction by geometrical parametrization for shape optimization in computational fluid dynamics, in *Proceedings of ECCOMAS 2010 CFD Conference*, (Lisbon, Portugal), June 2010.

25. A. Manzoni, A. Quarteroni, and G. Rozza, Shape optimization for viscous flows by reduced basis method and free form deformation, *International Journal for Numerical Methods in Fluids*, vol. 70, no. 5, pp. 646–670, 2012.

26. D. Peri, Conformal Free Form Deformation for the Optimisation of Complex Geometries, *Ship Technology Research*, vol. 59, no. 1, January 2012, ISSN 0937-7255.

27. D. Peri and E. F. Campana, Global optimization for safety and comfort, *COMPIT 2005*, pp. 477–486, Hamburg, 2005.

28. A. Manson and G. Thomas, Stochastic Optimisation of IACC Yachts, *COMPIT 2007*, 2007, Cortona, Italy.

29. J. Wang and T. Jiang, Nonrigid registration of brain MRI using NURBS, *Journal Pattern Recognition Letters*, vol. 28, no. 2, 2007.

30. T. Lassila, A. Quarteroni, and G. Rozza, A reduced model with parametric coupling for fluid-structure interaction problems, *SIAM Journal of Scientific Computing*, vol. 34, pp. A1187–A1213, 2012.

31. G. Rozza, An introduction to reduced basis method for parametrized PDEs, in *Applied and Industrial Mathematics in Italy* (W. Scientific, ed.), vol. 3 of *Series on Advances in Mathematics for Applied Sciences*, (Vol. 82, pp. 508-519, Singapore), 2009. Proceedings of SIMAI Conference, Italian Society for Applied and Industrial Mathematics, Rome, Italy, 14-18 September 2008.

32. G. Rozza, D. B. P. Huynh, C. N. Nguyen, and A. T. Patera, Real-time reliable simulation of heat transfer phenomena, in *ASME - American Society of Mechanical Engineers - Heat Transfer Summer Conference Proceedings*, (S. Francisco, CA, US), July 2009, Paper HT 2009-8812.

33. Available from: http://www.rhino3d.com.

34. Available from: http://www.autodesk.it/adsk/servlet/pc/index? siteID=457036&id=14626681.

35. Available from: http://help.solidworks.com.

36. C. Fassardi and K. Hochkirch, Sailboat design by response surface optimization, in *High Performance yacht Design Conference*, (Auckland, New Zealand), 2006.

37. I. H. Abbot and A. E. V. Doenhoff, *Theory of Wing Sections*. New York: Dover Publications Inc., 1959.

38. J. D. Anderson Jr., *Fundamentals of Aerodynamics*. McGraw-Hill, 2001.

39. A. Baron, "Fluid dynamics." Course material, www.aero.polimi.it, Politecnico di Milano, 2001.

40. F. M. White, *Fluid Mechanics*. McGraw-Hill, 3 ed., 1994.

41. A. Jameson and L. Martinelli, *Aerodynamic shape optimization techniques based on control theory*, vol. 1739/2000 of *Lecture Notes in Mathematics*. Computational Mathematics Driven by Industrial Problems, Springer Berlin / Heidelberg, 2000.

42. M. Avriel, *Nonlinear Programming: Analysis and Methods*. Dover Pub-

lications, September 2003.

43. H. A. v. d. Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, vol. 13, no. 2, pp. 631–633, 1992.

44. W. Banzhaf, P. K. R. E. Nordin, and F. D. Francone, *Genetic Programming - An introduction*. San Francisco, CA: Morgan Kaufmann, 1998.

45. Available from: http://www.mathworks.com/help/techdoc/index.html.

46. J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

47. J. F. Bonnans, J. C. Gilbert, C. Lemarechal, and C. A. Sagastizabal, *Numerical Optimization: Theoretical and Pratical Aspects*. Springer, 2 ed., 2003.

48. A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*. Milano: Springer, 2007.

49. Available from: http://www.pardiso-project.org.

50. D. Forti, Comparison of shape parametrization techniques for fluid-structure interaction problems, Master's thesis, Aerospace Eng., Politecnico di Milano, 2012.

51. F. Ballarin, Ottimizzazione di forma per flussi viscosi tridimensionali in geometrie cardiovascolari, Master's thesis, Mathematical Eng., Politecnico di Milano, 2011.

52. F. Ballarin, A. Manzoni, G. Rozza, and S. Salsa, Shape optimization by Free-Form Deformation: existence results and numerical solution for Stokes flows, *submitted to Journal of Scientific Computing*, 2013.

53. J. A. Samareh, A Novel Shape Parameterization Approach, in *Tech. Rep. NASA-TM-1999-209116*, March 1999.

MOX Technical Reports, last issues

**Dipartimento di Matematica "F. Brioschi",
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)**

04/2014 PALAMARA, S.; VERGARA, C.; CATANZARITI, D.; FAGGIANO, E.;
CENTONZE, M.; PANGRAZZI, C.; MAINES, M.; QUARTERONI, A.
*Patient-specific generation of the Purkinje network driven by clinical
measurements: The case of pathological propagations*

05/2014 ROZZA, G.; KOSHAKJI, A.; QUARTERONI, A.
*Free Form Deformation Techniques Applied to 3D Shape Optimization
Problems*

03/2014 KASHIWABARA, T.; COLCIAGO, C.M.; DEDE, L.; QUARTERONI, A.
*Well-posedness, regulariy, and convergence analysis of the Finite Ele-
ment approximation of a Generalized Robin boundary value problem*

02/2014 ANTONIETTI, P.F.; SARTI, M.; VERANI, M.
Multigrid algorithms for high order discontinuous Galerkin methods

01/2014 SECCHI, P.; VANTINI, S.; ZANINI, P.
*Hierarchical Independent Component Analysis: a multi-resolution non-
orthogonal data-driven basis*

67/2013 CANUTO, C.; SIMONCINI, V.; VERANI, M.
*On the decay of the inverse of matrices that are sum of Kronecker
products*

66/2013 TRICERRI, P.; DEDE ,L; QUARTERONI, A.; SEQUEIRA, A.
*Numerical validation of isotropic and transversely isotropic constitutive
models for healthy and unhealthy cerebral arterial tissues*

65/2013 AMBROSI, D.; CIARLETTA, P.
Plasticity in passive cell mechanics

64/2013 CIARLETTA, P; AMBROSI, D.; MAUGIN, G.A.
*Mass transport in morphogenetic processes: a second gradient theory
for volumetric growth and material remodeling*

63/2013 PIGOLI, D.; MENAFOGLIO, A.; SECCHI, P.
Kriging prediction for manifold-valued random field