Geometric Modelling and Deformation for Shape Optimization of Ship Hulls and Appendages

Elisa Berrini, Bernard Mourrain, Yann Roux, Mathieu Durand, Guillaume
Fontaine

HAL Id: hal-01373249
https://hal.science/hal-01373249v4

Geometric modelling and deformation for shape optimisation of ship hulls and appendages

Elisa Berrini[a,b], Bernard Mourrain[a], Yann Roux[b,c], Mathieu Durand[c], Guillaume Fontaine[c]

[a] *Université Côte d'Azur, INRIA Sophia-Antipolis Méditerranée, 2004 route des Lucioles, 06902 Sophia-Antipolis, France*
[b] *MyCFD, 29 Avenue des frères Roustan, 06220 Golfe-Juan, France*
[c] *K-Epsilon, WTC Bat E, 1300 Route des Crêtes, 06560 Valbonne, France*

Abstract

The precise control of geometric models plays an important role in many domains such as Computer Aided geometric Design and numerical simulation. For shape optimisation in Computational Fluid Dynamics, the choice of control parameters and the way to deform a shape are critical. In this paper, we describe a skeleton-based representation of shapes adapted for CFD simulation and automatic shape optimisation. Instead of using the control points of a classical B-spline representation, we control the geometry in terms of architectural parameters. We assure valid shapes with a strong shape consistency control. Deformations of the geometry are performed by solving optimisation problems on the skeleton. Finally, a surface reconstruction method is proposed to evaluate the shape's performances with CFD solvers. We illustrate the approach on two problems: the foil of an AC45 racing sail boat and the bulbous bow of a fishing trawler. For each case, we obtained a set of shape deformations and then we evaluated and analysed the performances of the different shapes with CFD computations.

Keywords: computers in design, hydrodynamics (hull form), design (vessels)

1. Introduction

Automatic shape optimisation is a growing field of study, with applications in various industrial sectors. As the performance of a flow-exposed object can be obtained accurately with CFD (Computational Fluid Dynamics), small changes in design can be captured and analysed. Based on these performance analysis capabilities, optimisation strategies can then be applied to deform the geometric model in order to improve the physical behaviour and the performances of the model.

Fig.1 shows the core of an optimisation loop, which will repeatedly run numerical simulations and deform the geometry for an automatic search of an optimal shape.
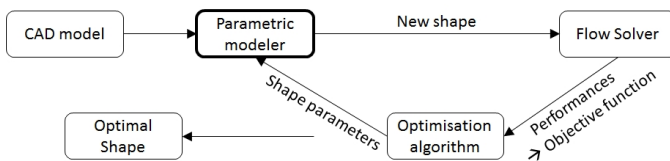

Figure 1: Automatic shape optimisation loop

Different types of tools need to be linked together to perform such automatic shape optimisation with aerodynamic or hydrodynamic criteria: a parametric modeller, a meshing tool, a flow solver and an optimisation algorithm [1, 2, 3], see Fig.1. Recent technological progresses allow to quasi-automatically run the meshing tool, the CFD solver and the post-processing of the relevant results of the computation. Then optimisation algorithms such as EGO (Efficient Global optimisation) [4, 5, 6] demonstrate their efficiency to solve problems with a large number of degrees of freedom and where the objective function values are difficult and costly to evaluate.

However, less efforts have been dedicated to the development of efficient parametric modellers. These components deform the object according to the optimisation algorithm output. Their role is critical in the way the space of possible shapes is explored. To be compatible with the current capabilities of the optimisation tools, the parametric modeller has to modify the shape of the object using a reduced number of parameters. It should provide a precise control of the shape, while allowing to generate a wide range of admissible shapes.

In this paper, we propose a new approach to shape deformation for parametric modellers with the purpose of being integrated into an automatic shape optimisation loop with a CFD solver.

The methodology presented here has the ability to generate valid shapes from an architectural point of view thanks to a novel shape consistency control based on architectural parameters. We focus on reducing the number of degrees of freedom of the deformation problem. We also focus on being independent from the CAD (Computer Aided-Design) software used by representing objects with a skeleton generated from the CAD model. Finally, we propose a methodology to link shape representation, shape deformation and numerical simulation.

Email addresses: elisa.berrini@inria.fr / elisa@mycfd.com (Elisa Berrini), Bernard.Mourrain@inria.fr (Bernard Mourrain), yann@k-epsilon.com (Yann Roux)

The motivation of working with architectural parameters is lead by the intuitiveness for an architect to control a shape by such expert parameters instead of control points.

Controlling shapes by architectural parameters allows reducing the number of degrees of freedom of the problem. They also introduce a physical and a design meaning into the optimisation process, allowing also to generate a majority of shapes that are valid. In comparison to fixing limits of variation for coordinates of control points, fixing the bounds to the architectural parameter variation is intuitive. Finally, architectural parameters are independent of each other, making the search of an optimal solution easier.

We propose a way to control shapes efficiently in terms of these architectural parameters, by controlling and deforming the skeleton curves in terms of these parameters.

The skeleton deformation is completed by a surface reconstruction step, to produce a smooth geometric model that can be used by the meshing and simulation tools. The approach allows us to be independent of the initial CAD representation and is not limited to a specific CAD software.

The generalizable concept of skeleton-based representation is well adapted to extend our tool to a large set a shapes e.g. hulls, appendages, propellers, wind turbine blades, airships.

In this paper, we illustrate application of the modeller on two applications: the AC45 foil used by racing yachts, and the bulbous bow of a trawler ship. For each case, we present the chosen hydrodynamic criteria to measure the performances, the shape parameter that we will modify and then we propose an analysis of the results.

2. Related work

In CAD software, the standard description used to describe shapes are B-Spline curves and surfaces [7]. A B-Spline curve of degree p is defined as :

$$C(t) = \sum_{i=0}^{n} B_{i,p}(t)c_i, \; t \in [0,1] \qquad (1)$$

where $c_i = (x_i, y_i, z_i)$ are the 3D control points, and $B_{i,p}(t)$ are the B-Spline basis functions.

For CFD computation, the object geometry is represented by a mesh. We present in the following paragraph existing methods based on both surface or mesh representations of shapes.

Deforming the control points of the NURBS representations used in CAD software to automatically generate new shapes is not appropriate. The number of control points to represent adequately the shape may be too large (3 degrees of freedom per control point) to be used in shape optimisation. Another obstacle is the complexity of the geometric models that can be trimmed, or subdivided into numerous patches that cannot be deformed in a structured way or that are simply not clean enough for CFD computations.

Shape deformation of ships forms for automatic shape optimisation is a relatively recent approach. However, deformation techniques have been highly developed in other application fields, such as 3D animation and movies.

Free Form Deformation FFD and morphing are classical methods created for 3D animation purposes, and they have been applied to shape optimisation for ships. Closely related to the FFD method, deformation techniques that enclose a shape in a mesh cage linked with barycentric coordinates have been proposed [8]. Naval applications with morphing can be found in [9, 10] and applications with FFD can be found in [11, 12, 13].

FFD and morphing are usually applied to meshes and not to a continuous geometry, thus limiting deformation because the meshes can be subject to degeneration. FFD methods can be very efficient with a small number of degrees of freedom to control the whole shape of the object. However, in order to perform local deformation, the only way is to increase the number of control points by refining the areas of interest. Moreover, FFD does not take into account any architectural parameters when deforming an object, leading to non-realistic results.

Morphing is limited to known bounds of shape variations. The exploration of the space of possible optimal shapes is extremely reduced.

For 3D animation, another common technique for controlling shape are skeleton-based mesh deformation techniques [14]. We can also find deformation techniques with subdivision surfaces [15], for example by using energy minimization techniques to find the best position of mesh elements to match the user's manipulations.

For applications with a direct interaction with the physical characteristics of the object, physically driven deformation methods exist. In this type of methods, the shape represents the domain where Partial Differential Equations (PDE) are solved. The domain can be either a mesh or a level set function. The results of the PDE are used in a cost function to determine parts of the domain needing to be deformed to optimise its value. Applications can be found for meshes [16], subdivision surfaces [17] and on level set function [18]. Generally in such applications, solving the PDE is not excessively costly. In shipbuilding, methods based on shape gradients focuses on minimizing an energy function obtained by solving the Navier-Stokes equations [19, 20].

Engineering dedicated CAD software can also provide parametric design features, allowing the user to build parametrized models such as *Catia^{TM}* or *Grasshopper* for Rhinoceros 3D^{TM} or *CAESES* from Friendship System^{TM}. These method allows to generate shapes easily, but all of the parameters defined on the design are lost when saving the model in a standard file exchange format such as IGES or STEP. This represents a limitation for automatic linking with solvers (CFD, structural analysis, etc.) or optimisation algorithms.

These type of parametrized models have been combined with isogeometric flow solvers for ship hull optimisation, for instance in [21, 22].

Specific software have been developed during the last decades for ship applications. One of the most widespread is *CAE-SES*, allowing the user to modify imported geometries using advanced geometrical parameters that can be modified by hand or automatically with a CFD optimisation loop [23]. The shape can be controlled with curves or surfaces around the object called *Lackenby shift transformations*. Points of the object are linked to the curves or surfaces, and follow its deformations.

Similarly, a ship dedicated tool *Bataos* [24] allows to modify the shape of sections of the hull by multiplying or adding predefined functions to the control points of the B-Spline curve describing the section.

These tools are based on geometrical control of shapes. Architectural parameters are computed on the deformed geometry and can be included as constraints, but they do not directly control the shape modification.

3. Shape parametrization

Our goal is to develop a generic methodology to deform shapes with architectural constraints. To achieve this objective, we use a twofold parametrization of the shape that allows us to describe a large class of objects in the same way. We base our method on a generic skeleton concept to describe the geometry, completed by specific architectural parameters according to the studied shape.

3.1. Geometrical parametrization

Our motivation for using a skeleton based representation of the geometry comes from two considerations:

1. Lines plan are used by naval architects to define the external shape of the hull, for which consistent shapes must be obtained once deformed.
2. Classical and efficient techniques in 3D animation are based on the deformation of medial axis curves associated to a shape [14].

By combining these two types of representations, we aim at applying generic deformation algorithms while controlling the architectural consistency.

We consider the skeleton as a set of curves composed of a generating curve and section curves. Each section curve needs to be identified on the generating curve: a local coordinate system, with an origin and a rotation, allows us to know its position and orientation. We are going to describe more precisely this skeleton based representation in the next section and how the architectural parameters are associated to the geometry in the following sections.

3.1.1. Generating curve and section curves

In our skeleton concept, the generating curve describes the general shape of the object, whereas sections describe more precisely the outlines of the object around the generating curve, similarly to the architect's line plan.

The generating curve needs to describe the prominent features of the object. It is defined to be lying on the geometry and connects all the section curves. It is not necessarily planar, but symmetry considerations of the object allow to describe it as a planar curve in most cases.

Section curves are computed as the intersection curves between the studied object and a family of planes. To each section curve, we associate a point on the generating curve, a local coordinate system, an origin and a rotation which allows to know the position and the orientation of the section curve. The cutting planes are defined to be normal to the tangent vector of the generating curve at the corresponding point adjusted with the rotation associated to the section.

For practical purposes, we represent the generating curve and the section curves as B-splines curves with a given number of control points. We further simplify the representation by choosing a finite subset of the section curves, associated with a finite sampling of the generating curve. Fig.2 illustrates skeletons obtained with Rhinoceros 3D$^{\text{TM}}$.

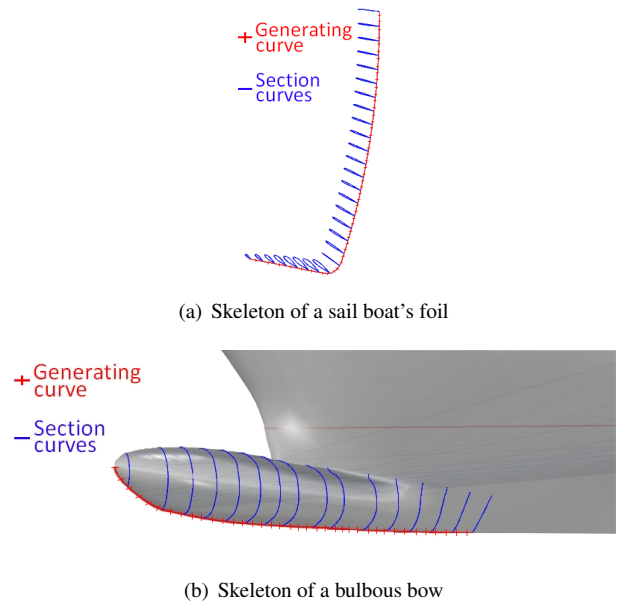(a) Skeleton of a sail boat's foil

(b) Skeleton of a bulbous bow

Figure 2: Examples of skeletons

This leads to a representation of the geometry in terms of a finite set of control points. We denote by $\mathbf{c}_g$ the control points of the generating curve and by $\mathbf{c}_i$ the control points of the ith sampled section curve for $i = 1, \ldots, N$.

We illustrate in the next paragraphs the method to obtain the skeleton on two different models.

To construct a skeleton-based representation from an initial geometric model, we first choose a relevant generating curve

according to the model. For airfoil based shapes, the trailing edges is an ideal choice, as is the keeline for a hull.

To obtain the section curves, we compute the intersection between the object and the set of planes defined according to the tangent of the generating curve. If a non-null rotation is associated to the section, the cutting plane is first transformed according to this rotation. The planes are sampled along the generating curve, following a chord length or a curvature based distribution. At this stage, we obtained a first skeleton from the model. The number of control points or the quality (continuity, smoothness, etc.) of the curves depends on the original design.

Then, we reconstruct new B-Spline curves with a fitting process [25] from a point cloud sampled on the current sections and the generating curve. We use a small number of control points (e.g. ≤ 10) to represent these curves, that are smoothed and cleaned. In the applications that we have considered this is usually enough to ensure a good level of approximation. The average normalized distance between the intersection curves and the B-spline section curves is kept under 10^{-5} m.

3.1.2. Local coordinate systems for the section curves

Section curves are identified on the generating curve thanks to a local coordinate system. Each local coordinate system has its origin defined from a point on the generating curve, allowing to locate the section in 3D space.

The first axis U is defined into the section plane, its direction is imposed by a main feature of the section, as the leading edge for an airfoil section, or the maximum height for a bulbous bow section.
The second axis V is represented by the tangent of the generating curve $T_{\sigma(t)}$ at the origin point of the local coordinate system. In most cases, the first and second axes are orthogonal by construction, but some sections representing special features of the geometry, as the extremities of the foil, are not defined in the plane P_T orthogonal to $T_{\sigma(t)}$. Thus U and V are not orthogonal to each other.

Let R_T be the rotation that transform U such as $U \in P_T$. We apply the inverse of the rotation R_T to $T_{\sigma(t)}$ in order to obtain the second axis V orthogonal to U. R_T is associated to the section.

Then the third axis W is computed as the cross product of the first two axes.

The implicit definition of the second axis V allows the local coordinate system to move when the generating curve is modified, computing the new orientation of the section automatically. The translation and rotation matrices that turns the original tangent to the new one is applied to the other axis U and W and to the section control points. Therefore modifying the generating curve induces affine transformations on the section curves, given by the modification of the local coordinate system.
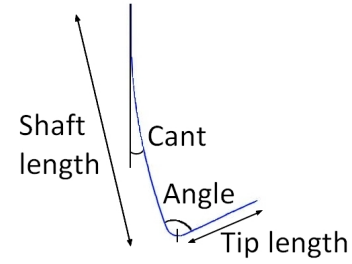
3.2. Architectural parameters

Architectural parameters describe the main characteristics of the object. They are chosen according to the design practice and effects on the object performance. Our goal is to control the shape of the studied object through the architectural parameters value.
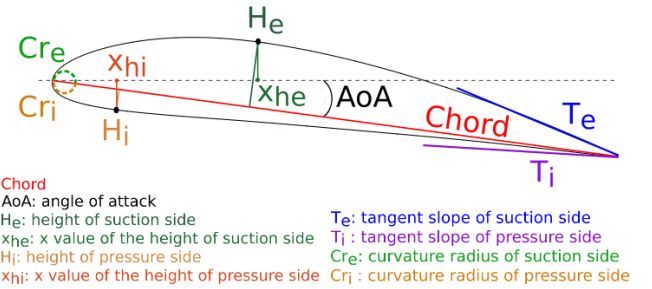
We associate different parameters to the generating curve and the section curves in order to control the whole shape.

For example, the main characteristics of an L-shaped sail boat foil are the length of the two parts, the angle between them and the angle of the entire foil called Cant. Then each airfoil profile section has particular features, such as chord length, thickness, angle of attack, etc. [26].

For a bulbous bow, the main features are the length, the angle, the height and thickness [27]. We illustrate those parameters in Fig.3 and Fig.4. The generating curve begins where the bulbous bow start to influence the hull shape. The length parameter is the total length of the generating curve. The angle parameter measures the angle between the x-axis and the extremity of the bulb. Variations of these parameters are shown in Fig.18. New types of parameters can be implemented easily to enrich the model, such as the sectional areas curve, the volume of the bulb, etc.

(a) Generating curve parameters

Chord
AoA: angle of attack
H_e: height of suction side
x_{he}: x value of the height of suction side
H_i: height of pressure side
x_{hi}: x value of the height of pressure side
T_e: tangent slope of suction side
T_i: tangent slope of pressure side
Cr_e: curvature radius of suction side
Cr_i: curvature radius of pressure side

(b) Section parameters (airfoil parameters)

Figure 3: Foil parameters

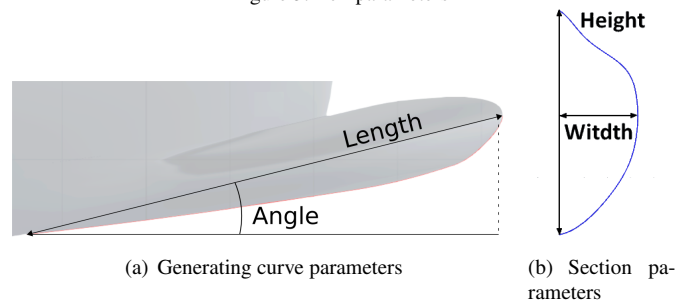

(a) Generating curve parameters

(b) Section parameters

Figure 4: Bulbous bow parameters

3.3. Observer function

We call ϕ, the observer function that computes the set of architectural parameters P on a given geometry G: $\phi : G \longrightarrow P$. These parameters can be real values such as the length of a foil or functions of the generating curve parameter, such as the twist angle of a profile defined at each point of the generating curve. For a given geometry $\sigma \in G$, the architectural parameters $\phi(\sigma)$ can thus belong to an infinite dimensional space since it can contain functions which represents values along the generating curve.

In practice, these functions will be represented with a B-Spline curves passing through the section parameter values according to their position on the generating curve. The B-Spline curves belong to a finite dimensional space with a small number of control points. These are the parameters that we will use to control the shape.

An illustration is shown in Fig.5, where the observer function made of 13 control points represents the chord length distribution of 28 section curves.

From this consideration, managing the B-Spline instead of each section parameters represents two main advantages. First, we reduce drastically the number of parameters that control the shape of the object and that are used in an optimisation loop. Secondly, the modification of a B-Spline curve can ensure a smooth distribution of the parameters, preserving the fairness of the object. The observer function can be split into a part for the generating curve and a part for the section curves, as different set of parameters can be defined on each type of curves.
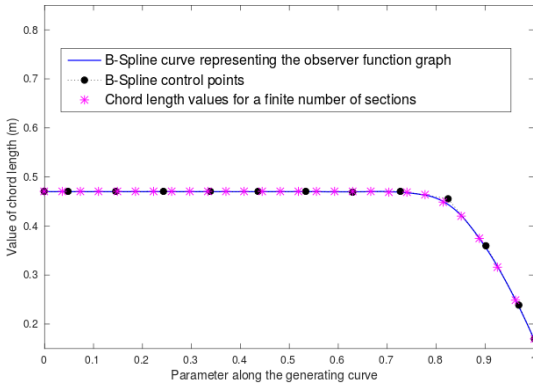

Figure 5: Distribution of the chord length parameter along the generating curve of a foil, approximated with a B-spline curve called *observer function*

4. Shape deformation

This section explains our strategy for computing a smooth shape corresponding to given architectural parameters. We described the problem as a non-linear constrained optimisation problem that can be applied on the generating curve or the section curves independently.
We start by presenting the problem, then we propose an optimisation algorithm to solve it numerically.

4.1. Problem setting

Our goal is to find the shape of G that matches a given set of architectural parameters in P.

The observer function $\phi : G \longrightarrow P$ is defining the parameters associated to a shape. To control the shape of the object through the parameters value, we need to find a shape corresponding to given parameters. In other words, we need to compute: $\phi^{-1} : P \longrightarrow G$.

As the shape in G is described by a skeleton made of B-Spline curves, we propose a method that computes new values of the coordinates of B-Spline curves control points until the new skeleton parameters reaches the target ones. The new coordinates of the B-Spline control points are the solution of a minimisation system that we construct with four terms. The discretised geometry, represented by a finite number of section curves and a generating curve is called ξ. The generating curve parameterized by $t \in [0, 1]$ is denoted ξ_g. Its controlled coefficients are $\mathbf{c}_g$. The ith sampled section curve corresponding to the parameters t_i on ξ_g is denoted ξ_i for $i = 1, \ldots, N$. It is parameterized by $s \in [0, 1]$ and its control points are $\mathbf{c}_i = (\mathbf{c}_{i,0}, \ldots, \mathbf{c}_{i,M})$. In the following paragraphs, ξ^0 denotes the initial geometry, that is the initial generating curve and section curves.

Parameters value
The first term measures the distance of the current parameters values $\phi(\xi)$ to the target ones V:

$$E_{param} = \|\phi(\xi) - V\|^2 \tag{2}$$

As we assume that the observer function can be split into a part for the generating curve and a part for the section curves, this error term is the sum of an error term for the generating curve and error terms for the sections. We denote by $E_{param,i} = \|\phi_i(\xi_i) - V_i\|^2$ and $E_{param,g} = \|\phi_g(\xi_g) - V_g\|^2$ the error term corresponding to the ith section curve and the generating curve respectively.

Shape consistency control
The second term is introduced to ensure consistency control by measuring the distance of the current generating or section curve to the original one. The consistency with the initial geometry is measured after applying a linear transformation which allows to match some parameters of the target curve ξ. These transformations include a scaling of the initial curve to match a given length or a rotation to match a given angle. In addition to these basics transformations, we also consider other explicit deformations which depend on the parameters V, such as a non-linear scaling of the height of a profile. These transformations of the initial curve, denoted D_V, are explicitly computed from the geometry ξ_i (or ξ_g). Transforming the initial curve by D_V helps matching the target parameters. The transformed geometry $D_V(\xi_i^0)$ (or $D_V(\xi_g^0)$) is used as the starting point of the optimisation algorithm.

We define:

$$E_{shape,i} = \|\xi_i - D_V(\xi_i^0)\|^2 \tag{3}$$

Similarly, for the generating curve we have $E_{shape,g} = \|\xi_g - D_V(\xi_g^0)\|^2$.

Architectural constraints

The third term allows taking into account specific constraints F for the studied object, usually position or tangency constraints. These constraints are defined for each section $\xi_i, i = 1, \ldots, N$ and are not necessarily the same for all sections. For example, an airfoil has a smooth connection between the suction and pressure faces thanks to a tangency constraint: the tangent at the leading edge has to be orthogonal to the chord vector:

$$F_1 : \frac{\partial \xi_i}{\partial s} \cdot \overrightarrow{chord} = 0.$$

For a bulbous bow, as we parametrized a half bulbous bow, we have to ensure that the sections end at $Y = 0$ and that the tangent at the extremity are preserved:

$$F_0 : Y(\xi_i(1)) = 0 \quad F_1 : \frac{\partial \xi_i}{\partial s}(v) - \frac{\partial \xi_0}{\partial s}(v) = 0 \quad v \in \{0, 1\}$$

Regularization

The last term controls the overall smoothness of the shape by introducing stiffness between successive control points $\mathbf{c}_{i,j}$. We add correction terms to control respectively C^1 and C^2 properties of control points.

$$H_0(\mathbf{c}) = \sum_j \|\Delta \mathbf{c}_j\|^2 \qquad \Delta \mathbf{c}_j = \mathbf{c}_j - \mathbf{c}_{j-1}$$

$$H_1(\mathbf{c}) = \sum_j \|\Delta^2 \mathbf{c}_j\|^2 \qquad \Delta^2 \mathbf{c}_j = \mathbf{c}_{j+1} - 2\mathbf{c}_j + \mathbf{c}_{j-1}$$

Finally, the proposed minimisation system is described as follows.

$$\min_{\mathbf{c}_i} E_{param,i} + \varepsilon E_{shape,i} + \sum_k \lambda_k F_k^2(\mathbf{c}_i) + \sum_{l=0}^1 \mu_l H_l(\mathbf{c}_i) \quad (4)$$

As we decoupled it into a minimization system for each section curve ξ_i and for the generating curve ξ_g, an optimisation problem similar to Eq.4 is solved for the generating curve ξ_g.

In these formulations, ε is a weight allowing to balance the influence of the shape control term. In fact, if this term is too high, the system will converge to a solution too close to the initial curv, and will have difficulty to respect the target parameters. ε can be seen as a penalty coefficient, but we chose to decrease it at each iteration because in our particular case the initial guess, the original curve, needs to be degraded to match new architectural parameters. The coefficients λ_i weighing the shape constraints and μ_i weighing the correction matrices are both very small, usually around 10^{-4}.

4.2. Numerical solution

The definition of the problem is well adapted to Sequential Quadratic Programming (SQP) [28]. SQP algorithm solves the minimisation problem by generating successive quadratic problems that approximate the cost function by a quadratic function obtained from a Taylor expansion of order 2. We use finite difference to compute the gradients of the system. We start with an initial value of ε and the original curve as the starting point of the algorithm, then we decrease ε at each iteration and start the SQP again with the last computed curve. The algorithm stops when the value of the objective function reaches a fixed threshold.

Fig.6 illustrates the deformation process. This inner problem, solved by SQP, has a relatively large number of degrees of freedom as it modifies the coordinates of control points. But with our methodology based on architectural parameters, the shape optimisation outer loop (see Fig.1) controls only a few number of degrees of freedom.
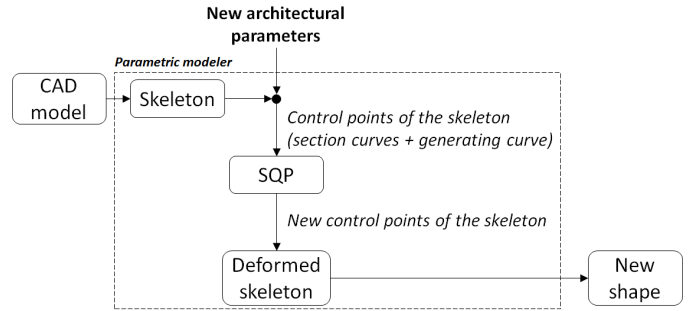

Figure 6: Deformation process

Example

Fig.2(a) illustrate the skeleton we obtained for the AC45 foil. The original model is made of 22164 control points. Our skeleton representation is made of only 578 points, $560 = 2 \times 28 \times 10$ points for the sections and 18 points for the generating curve.

The associated deformation process can be decoupled into 29 independent sub-problems. 28 sub-problems representing the suction and pressure faces of each sections are controlled by 6 architectural parameters. Each sub-problem has 16 degrees of freedom, as the extremal points of the sections are predetermined by chord and angle of attack. The last sub-problem, representing the generating curve, is controlled by 4 architectural parameters and has 36 degrees of freedom.

For the bulbous bow, whose skeleton is illustrated in Fig.2(b), the total number of control points for the half-hull is 2574, and 57 are directly controlling the bulbous bow shape. Our skeleton representation is made of 180 points, $160 = 16 \times 10$ points for the sections and 20 points for the generating curve. We choose to define more control points on the shape of the bulb than in the original model, as we look for a precise control of the shape of the bulbous bow. The number degrees of freedom used by

the shape optimisation problem is smaller as they are defined by the architectural parameters.

The associated deformation process can be decoupled into 11 independent sub-problems. 10 sub-problems representing the sections are controlled by 2 architectural parameters. Each sub-problem has 16 degrees of freedom, as the extremal points of the sections are pre-determined by the height. The last sub-problem, representing the generating curve, is controlled by 2 architectural parameters and has 40 degrees of freedom.

Moreover, the total number of variables controlling the sections can be reduced with *observer functions* described in Section 3.3.

An illustration of the deformation of a sail boat section is shown in Fig.7(b). The three parameters of the section are described by Fig.7(a). An observer function describes the repartition of the parameters along the sections, but only one section of the skeleton is considered in this example, made of 8 control points. The aim of the deformation is to reduce the value of the *Radius* parameters, while *Width* and *Length* are fixed. Considering this objective, the extremal points of the sections are fixed. The problem solved has 12 degrees of freedom and converges in 15 iterations, with 20 iterations of SQP each.

With a four-cores HP Probook-450 with a Intel® Core™ i7-4702MQ CPU 2.20GHZ, RAM 8.00 GB, the total time to perform this section deformation is 1.2 seconds.

5. Surface reconstruction

The optimisation method outputs deformed sections and generating curves, corresponding to the skeleton of a new shape. To evaluate the shape's performances with a CFD solver, we first need to reconstruct the 3D surface wrapping the deformed skeleton.

Building a new surface allows to obtain a cleaned-up model for the meshing tool. The quality of the obtained surfaces is ensured by the high precision of the implemented fitting process and the control of continuity between patches.

Classical techniques such as *lofting* [7] are a relevant choice for objects that can be represented with only one surface such as foils.

For more complex objects, multi-patch surfaces are required. In such cases, a particular attention has to be given to the continuity between them: for our application, patches have to be at least C^1. We developed a technique based on *form finding* [29] to reconstruct suitable surfaces. We expose this technique in the following paragraphs.

5.1. Problem setting

Given the section and generating curves of the new deformed skeleton, we construct a surface which contains these curves and satisfies tangency conditions on the boundaries of the surface.
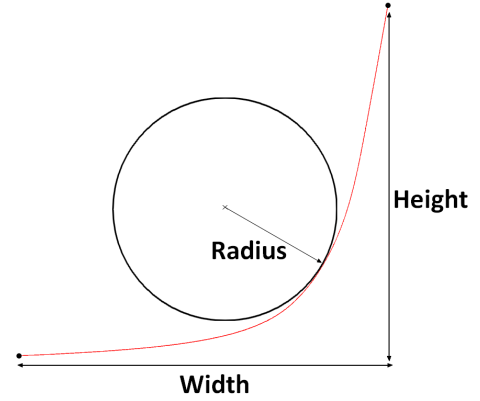
(a) Ship section parameters
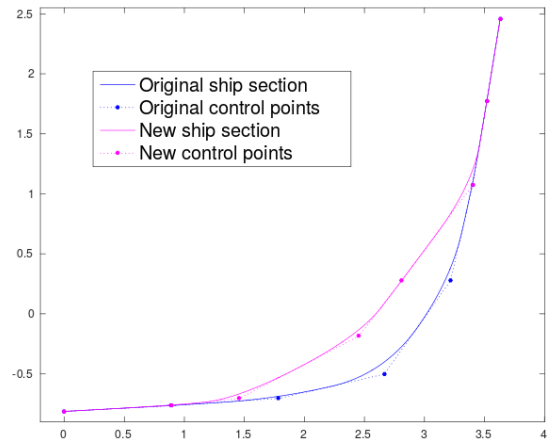
(b) Deformation of the curvature radius of a sail boat section

Figure 7: Ship section parameters and deformation

Like for the skeleton deformation, we compute the surface by solving an optimisation problem, where the control points of the surface c_{ij} are the unknowns. To define this optimisation problem, we discretize the problem by sampling the curves. We obtain a point set on which we will wrap the surface. The surface is computed by fitting techniques [30], taking into account smoothing and tangency constraints on the boundary of the surface to ensure the continuity between adjacent surfaces in the geometric model. As explained in [29], the constraints we use are quadratic in the control point coordinates c_i. We describe them in the following paragraphs.

First, we define for each point of the point set a coordinate mapping:

$$\mathbb{R}^3 \longrightarrow [0,1] \times [0,1]$$
$$P_l \longrightarrow (u_l, v_l), \qquad\qquad l = 0, ..., N_P$$

The mapping defines for each point P_l of the point set, the parameters u_l, v_l of the surface σ where $\sigma(u_l, v_l) = P_l$ will be verified approximately.

Surface fitting
This constraint ensure that the surface σ passes thought the points P_l:

$$E_{fitting} : \sum_l \|\sigma(u_l, v_l) - P_l\|^2 = 0, \ l = 0, ..., N_P \quad (5)$$

$$\text{with } \sigma(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} c_{ij} B_i(u) B_j(v)$$

Tangency constraint with fix parts of the object

Let n_l be the normal at P_l of the fix surface adjacent to a surface we want to reconstruct. In the u direction, the continuity constraint is expressed by:

$$E_{fixT} : < \sigma_u(u_l, v_l) \cdot n_l >^2 = 0, \ l = 0, ..., N_P \quad (6)$$

$$\text{where } \sigma_u(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} c_{ij} B_i'(u) B_j(v)$$

We have similar constraints in the v direction.

Tangency constraint with mobile parts

At the N_n points P_l on the frontier with other reconstructed surfaces, the values of the normals n_k of both surfaces are new unknowns satisfying an equality constraints. In the u direction, the continuity constraint is expressed by:

$$\begin{cases} E_{mobileT1} : < \sigma_{1\,u}(u_{1,l}, v_{1,l}) \cdot n_k >^2 = 0 \\ E_{mobileT2} : < \sigma_{2\,u}(u_{2,l}, v_{2,l}) \cdot n_k >^2 = 0 \end{cases} \quad (7)$$

$l = 0, ..., N_P, \ k = 0, ..., N_n$ (and similarly for the constraints in the v direction).

Moreover, the normal vector n_k must satisfy:

$$E_{mobile\ normals} : < n_k \cdot n_k >^2 = 1, \ k = 0, ..., N_n \quad (8)$$

Notice that these constraints require an initial value of σ_1, σ_2 and n_k.

Regularization

A regularization energy term can also be introduced for the surfaces, to improve the "fairness" of the surface. It is a quadratic function of the unknowns control coefficients $c_{i,j}$, similar to the regularization term for curves used in Section 4.1. We do not detail it here (see for instance [29]).

5.2. Numerical solution

Let us consider $\mathbf{x}$ as the vector containing the unknown of the system, in other words the surfaces control points c_{ij} and the normals n_k at the frontier with two reconstructed surfaces. The surface is constructed so that the total energy is minimized:

$$E_{total} = E_{fitting} + E_{fixT} + E_{mobileT1} + E_{mobileT2} + E_{mobile\ normals} \quad (9)$$

A dedicated algorithm is used to compute a value of $\mathbf{x}$, for which E_{total} is less than a threshold. Let us describe it briefly.

The general form of quadratic constraints that we treat is:

$$\varphi_i(x) = \frac{1}{2} x^T H_i x + b_i^T x + c_i = 0, \ i = 1, ..., N \quad (10)$$

where H_i is a symmetric matrix, b_i is a vector and c_i a constant. Some of the constraints that we use are not quadratic e.g. the continuity between patches. In such cases we use a geometrically meaningful linearization, e.g. expressing the constraint in a quadratic form using the normal of the surface.

Given the definition of the quadratic constraints in Eq. (10) and a value $\mathbf{x} = \mathbf{x}_n$ at iteration n, we can linearize $\varphi_i(x)$ using:

$$\mathbf{x} = \mathbf{x}_n + \delta x$$

$$\varphi_i(\mathbf{x}) \approx \varphi_i(\mathbf{x}_n) + \nabla\varphi_i(\mathbf{x}_n)^T (\mathbf{x} - \mathbf{x}_n) = 0, \ i = 1, ..., N$$

where $\nabla\varphi_i(\mathbf{x}_n) = H_i\mathbf{x}_n + b_i^T$. We can rewrite this linearization in the following matrix form:

$$\begin{pmatrix} \nabla\varphi_1(\mathbf{x}_n)^T \\ \vdots \\ \nabla\varphi_N(\mathbf{x}_n)^T \end{pmatrix} \cdot \mathbf{x} = \begin{pmatrix} \nabla\varphi_1(\mathbf{x}_n)^T \cdot \mathbf{x}_n - \varphi_1(\mathbf{x}) \\ \vdots \\ \nabla\varphi_N(\mathbf{x}_n)^T \cdot \mathbf{x}_n - \varphi_N(\mathbf{x}) \end{pmatrix},$$

that is, a linear system of the form $H_n \cdot \mathbf{x} = r_n$, whose solution is the next point $\mathbf{x}_{n+1}$. We solve this system iteratively until a fix point is reached.

This technique is able to reconstruct efficiently and accurately surfaces, which contain the skeleton curves and satisfy tangency constraints on the boundary.

We illustrate it with the reconstruction of the surface of a sail boat hull, in Fig.8 , and on the bulbous bow of a fishing trawler, in Fig.9.

For the sail boat hull example, we choose to reconstruct the middle part of the hull with two surfaces. Each surface has to be smoothly connected to a fix part of the hull (the transom or the stem) and to the other middle surface. The algorithm converges in 3 iterations, and the resulting surfaces satisfy the tangency constraints at the three junction curves.
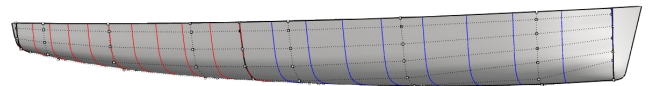

Figure 8: Patch of surfaces reconstructed of a sail boat hull

For the bulbous bow example, the bulb is reconstructed with two surfaces. The first one is connected to the fixed hull and the other is connected to the first surface. The algorithm converges in 4 iterations, and the resulting surfaces satisfy the tangency constraints at the four junction curves.

6. Applications

In this section we present two different applications of the parametric modeller, one on the foil of an AC45 and one on the bulbous bow of a fishing trawler. In both cases, we aim to increase a performance criterion with shape variations. The parametric modeller is automatized and linked to a flow solver. A specific flow solver is used for each application: potential
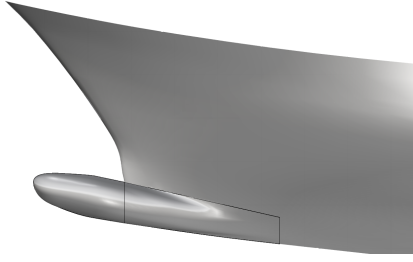
Figure 9: Patch of surfaces reconstructed of bulbous bow

flow solver for the foil and RANSE for the bulbous bow.

Timings in the experimentations refer to the same hardware configuration as described in section 4.2.

6.1. Application on a sail boat's foil: AC45

In the recent years, new high-speed boats were developed using foils. The purpose of a foil is to lift the hull of the boat above water surface. The hull resistance (friction and wave making drag) is decreasing, allowing to reach very high speeds.

For sailing yachts, the foils are built as an "L" shape with a vertical part countering the sails forces, and a horizontal part supporting the yacht weight.

While sailing, the foil allows the yacht to fly as shown in Fig.10. However, to maintain this flying state, the stability of the foil is a critical aspect for both security and performance.

Designers have to manage numerous parameters in order to produce a foil with a low drag, but high stability.

We consider here the AC45 foil. This type of foil is "one-design" meaning that its shape is the same for all AC45 boats.

For this application, we aim to optimise the shape of the AC45 foil in order to decrease its total drag while keeping stability and the ease of use as high as possible. The foil performances are computed with the potential flow solver ARAVANTI.

The AC45 foil is currently used by the Groupama Team France sailing team for the 35th America's Cup. An illustration of the sail boat flying thanks to the foil is shown in Fig.10, one foil in the water (right) and the other one visible in the retracted position (left).

6.1.1. Simulation with ARAVANTI

ARAVANTI, the flow code used in the present study is developed and commercialized by the company K-Epsilon. ARAVANTI is a coupled fluid-structure solver, with a finite element method for solving the structure and multiple different methods for the fluids (e.g. vortex line method, particle method, panel method, etc.) [31, 32].

The method used here is a vortex line method with solved wake. ARAVANTI is coupled to XFOIL in order to incorporate the flow behaviour such as laminar transition, and stall.

Figure 10: Illustration of the AC45 on the Groupama Team France sail boat, *Credit: ® Eloi Stichelbaut / Groupama Team France*

The foil is represented with a finite number of elements, i.e. airfoil sections given by the skeleton. For each element a local velocity, a local Reynolds number and a local angle of attack is computed. Each element has an associated XFOIL database containing the lift and drag of the section for a given range of angles of attack (usually between $-5°$ and $20°$).

ARAVANTI use this database to find the lift of each element of the foil according to its current local angle of attack. Then the lift is converted to a local vorticity. The wake is imposed with the computed gradient of vorticity then solved. These steps are repeated until convergence thanks to a direct iterative method, which is able to find a stationary solution.

For this type of application, ARAVANTI does not need to be linked to any CAD software. As inputs, it requires a 2D point cloud description of the sections, similar to the XFOIL input, and a text file indicating the 3D position of each section with points and quaternions. These files are generated automatically by the parametric modeller.

In our specific case for AC45 foil study, only the underwater part of the foil is simulated. The influence of the free surface is taken into account with an anti-symmetry plane model. This model is a satisfying approximation for high speed. As [33] suggests, with a Froude number greater than 1, an infinite Froude number free-surface condition can be used. In our case, the Froude number is around 5.45.

We illustrate in Fig.11 the wake computed with ARAVANTI and the vortex lines. The vortex line is located at 25% of the aft of the leading edge along the foil. From the vorticity repartition colormap, we see that the parts of the AC45 which generate most of the force allowing to lift up the boat are the knee and the tip.

The reference frame is defined as follows: X is in the opposite direction of the flow, Z is in the vertical direction (oriented upwards) and Y is horizontal, perpendicular to X.
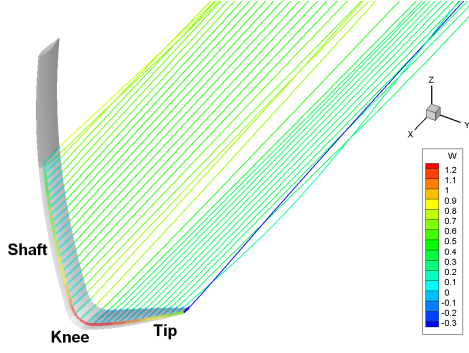
9

Figure 11: Illustration of the wake and vortex line on the AC45

6.1.2. Proposed performances criteria

We choose to define the foil performances with three criteria computed with ARAVANTI.

1. The total drag F_x of the foil in the reference frame. A low drag increases the total performance and speed of the boat.
2. A stability criterion, represented by $\frac{\partial F_z}{\partial z}$, where F_z is the total force in the z direction of the foil. The aim of this criterion is to ensure that the boat will stay at a fixed z height thanks to a self adjusting F_z balancing the vertical movements of the foil.
3. A stability and usage criterion, represented by $\frac{\partial rake}{\partial V}$, where the *rake* is the angle of incidence of the foil in the Y rotation, and V is the boat speed. The rake is a parameter that the crew have to adjust while sailing to modify the vertical forces F_z. Thus a foil shape where this parameter does not change a lot when the speed is varying is valuable.

Computations are performed with a fixed F_y given as the opposite force to balance the force applied by the sails on the hull. F_z is also fixed to counter the weight of the hull and be able to lift it up. The speed of the yacht is first set to 22 knots. ARAVANTI solves for the leeway and rake angles of the foil, until computed forces converge to the imposed forces.

F_x is computed during the simulation, and we aim to decrease it as much as possible. In the reference frame we used, F_x is oriented along the negative x direction. Thus, the sign of F_x will be negative, but we can consider the absolute value to compare the foil performance.

To compute the second criterion, we estimate $\frac{\partial F_z}{\partial z}$ with finite differences. We vary the foil displacement by a small Δz and compare the computed F_z. To be stable, the foil has to generate a F_z opposed to the direction of the displacement. Thus the ratio $\frac{\partial F_z}{\partial z}$ has to be negative and as large as possible.
For example, if the boat is riding too high above the water surface, the foil force F_z has to decrease in order to make the whole system lower.

We use the same process for the third criterion, $\frac{\partial rake}{\partial V}$, by solving the rake angle for a small speed variation ΔV. Here,

the rake has to increase as little as possible when the speed increases. Thus the ratio $\frac{\partial rake}{\partial V}$ has to be positive and as small as possible.

For both cases, we ensure that the finite difference is a satisfying approximation by choosing appropriate steps Δz and ΔV.

The aim of our study is to reduce the total drag of the AC45 as much as possible while keeping stability criteria as large as possible.

6.1.3. Proposed deformations

We identified the most relevant parameters that influence a foil performances as the tip length, the angle between the shaft and the tip and the cant angle, illustrated in Fig.12. Here, we consider the cant angle as a shape parameter and not as control parameter of sailing.

To generate a new CAD from the original CAD model, our tools takes on average 12 seconds to build the skeleton, 5.1 second for the generating curve deformation and 5 seconds for the section curve deformation. In our case, we perform only deformation of the generating curve. Moreover there is no need to build a new surface around the skeleton, as ARAVANTI does not require a continuous surface as an input. A set of points distributed on the section curves of the skeleton is sufficient. The skeleton we used on the AC45 is illustrate in Fig.2(a).
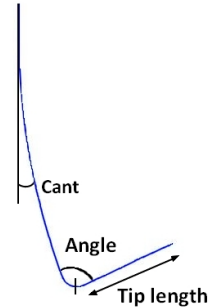

Figure 12: Foil shape parameters

The variations of the parameters are distributed in a parameter space defined in Tab.1, and illustrated in Fig.13.

	Tip length	**Angle**	**Cant**
Initial value	1.37m	77.24°	2.42°
Min variation	−30% (= 0.96m)	−30% (= 54.1°)	−313.7% (= −5.2°)
Max variation	+40% (= 1.92m)	+20% (= 92.65°)	+727.2% (= 20°)

Table 1: Limits of parameters domain

To sample the parameter space, we use a Latin Hypercube distribution [34]. Our choice is based on the future use of optimisation algorithms such as EGO, that are often initialized
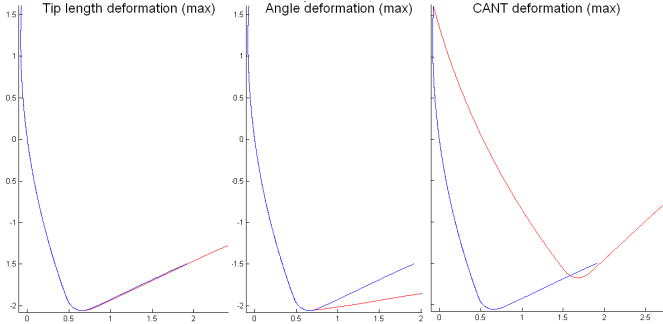
Figure 13: Shape variation of the foil in the paramter space

with such parameter space values distributions as they are well adapted for response surface methods [35].

6.1.4. Results

We used a Latin Hypercube distribution with 20 points to sample the parameter space described in Tab.1. For each set of parameters, we build a new corresponding foil with our parametric modeller and we evaluate automatically the value of the 3 criteria, F_x; $\frac{\partial F_z}{\partial z}$; $\frac{\partial rake}{\partial V}$, with ARAVANTI.

As our aim is to reduce the total drag as much as possible while keeping stability criterion as large as possible, the optimal solution is located on a Pareto front. We represented the Pareto fronts of the drag with each stability criterion in Fig.14. The Pareto front is defined with few points at this stage of the study. More points will be added while performing the automatic shape optimisation process. In a future work, the performance of the foils located on the current Pareto front will be improved by including an adapted optimisation algorithm.

We named the foils closest to the Pareto fronts (A,B,C,D), Foil A being the one with the least drag and worst stability, Foil D being the one with the most drag, but the best stability and Foils B & C being in between. Even if Foil A has the worst stability of the Pareto front, it is still better than the original AC45. The other criteria vary around the original values.

Note that the foils A, B, C and D refer to the same shapes on both Pareto fronts $\frac{\partial F_z}{\partial z}$ vs F_x and $\frac{\partial rake}{\partial V}$ vs F_x.

We detail the points on the Pareto fronts in Tab.2, with the initial AC45 results for comparison. We illustrate the results in Fig.15.

The two shape variations Foil A and Foil B are rather different for the tip length and angle values. We can deduce a link between these two parameters that leads to more efficient foils, either a short tip with a great angle or a long tip with a small angle. Both cases suggest to increase the cant angle.

The two extreme shapes in the Pareto front $\frac{\partial F_z}{\partial z}$ vs F_x, Foil A and D, show a very different behaviour of the foil according to the parameters, illustrated in Fig.16 where we see the vorticity distribution along the foil. In the case of Foil A (Fig.16(a)) the vorticity is uniformly distributed on the shaft, knee and tip.
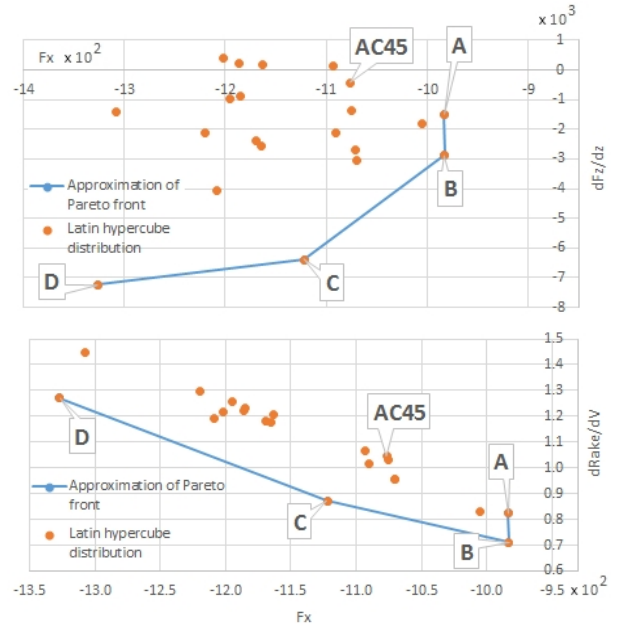

Figure 14: Latin hypercube distribution of foils shape

#	% Tip length variation	% Angle variation	% Cant variation	Total drag ($|F_x|$) in N	$\frac{\partial F_z}{\partial z}$	$\frac{\partial rake}{\partial V}$
AC45	-	-	-	1077	−423	1.045
A	−8.62%	+18.58%	+650.53%	**983**	−1495	0.825
B	+34.02%	+9.74%	+416.79%	983	−2863	0.710
C	+31.56%	−5.96%	+591.87%	1122	−6392	0.872
D	+11.88%	−29.63%	+721.38%	1327	−7240	1.271

Table 2: Parameters and criteria values of points on both Pareto fronts

Whereas for Foil D (Fig.16(b)), the vorticity in essentially located on the shaft, thus the lifting force is principally generated from this part.

To conclude, the behaviours we observed of the different foils match expected results, and some tendencies are well known by designers.

A further study will include the sinkage as well as shape parameters for the sections. We will also take into account the moment of the boat about the x direction M_x. The moment has an influence on the predicted performance of the foil, and especially the value of cant angle can be affected in order to find a configuration that counters M_x.

Also, an optimisation algorithm will be integrated in the loop, helping to determine with certitude the best tendency of parameter values.

6.2. Application for a bulbous bow

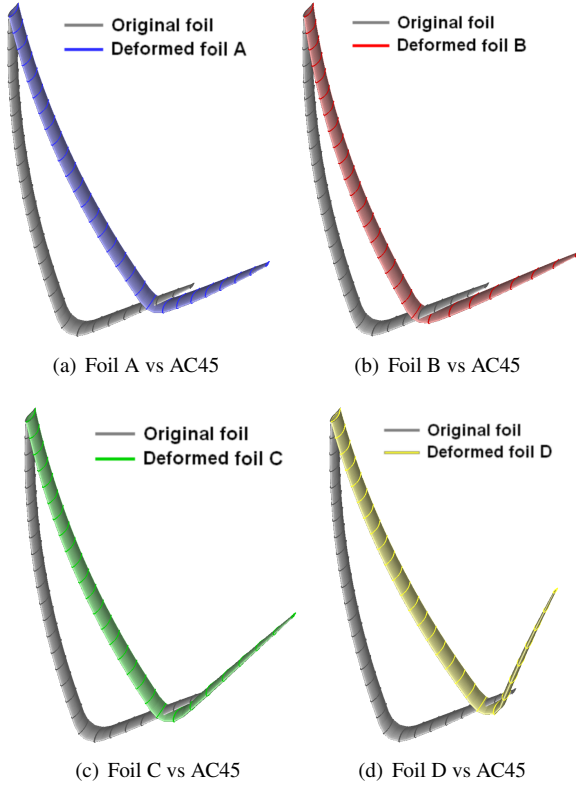We present an application of our parametric modeller for deforming a fishing trawler bulbous bow.

(a) Foil A vs AC45   (b) Foil B vs AC45

(c) Foil C vs AC45   (d) Foil D vs AC45

Figure 15: Shape variations on Pareto fronts

(a) Foil A

(b) Foil D

Figure 16: Wake of Foils A and D, front view

The original trawler was designed without a bulbous bow. We aim to reduce the total drag of the hull by adding a bulbous bow.

An initial bulb was designed by a naval architect, then we propose to vary three parameters to control the shape: the angle the length and the width at mid-bow of the bulb.

To generate a new CAD from the original CAD model, our tools takes on average 27.6 seconds to build the skeleton, an average of 14.1 seconds to perform deformations, and 20 seconds to reconstruct the surface.

RANS (Reynolds-Averaged NavierStokes equations) simulation being more complex to set-up, the link with the parametric modeller was not fully automatised.

6.2.1. Simulation with FINE^TM/Marine

To generate non-conformal, fully hexahedral, unstructured meshes for complex arbitrary geometries, we use HEXPRESS^TM from Numeca International. The advanced smoothing capability provides high-quality boundary layers insertion [36]. The software HEXPRESS^TM creates a closed water-tight triangularized volume, embedding the ship hull, then a body-fitted computational grid is built. One of the meshes used in our simulations is shown in Fig.17.

The grid generation process requires a clean and closed geometries to provide robust meshes. Thanks to the shape consistency control and the smooth reconstruction of surfaces, the
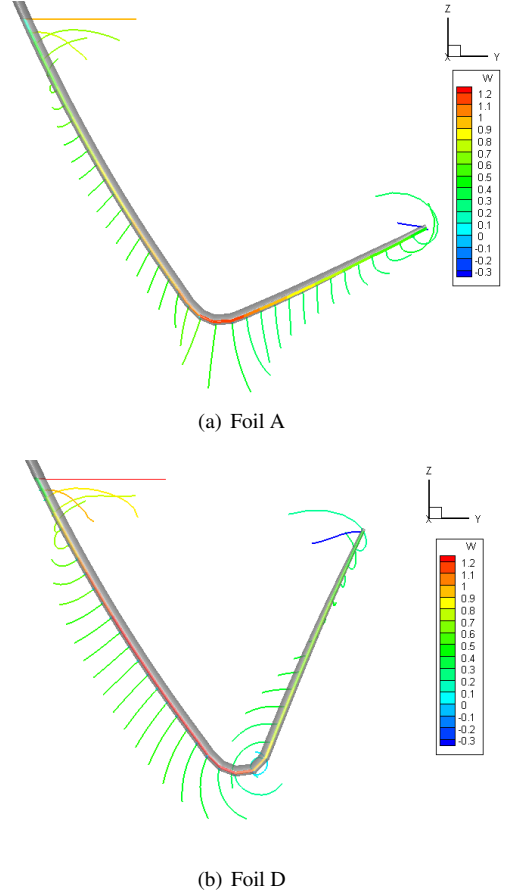
modeller generates shapes which are well-adapted to these requirements and which allow to produce high-quality meshes for computations.

During the computation, automatic mesh refinement has been used. Automatic, adaptive mesh refinement is a technique for optimising the grid in the simulation, by adapting the grid to the flow as it develops during the simulation to increase the precision locally. This is done by locally dividing cells into smaller cells, or if necessary, by merging small cells back into larger cells in order to undo earlier refinement. During the computation, the number of cells increases from 1.9 to approximatively to 2.2 million cells, for a half hull mesh. Fig.17 shows the mesh refinement around the hull and the free surface at the end of the computation.

We use the flow solver ISIS-CFD, available as a part of the FINE^TM/Marine computing suite. It is an incompressible, unsteady Reynolds-averaged Navier-Stokes (RANS) solver [37, 38]. For the turbulent flow, additional transport equations for the modeled variables are discretized and solved. The two-equation k-ω SST linear eddy-viscosity model of Menter is used for turbulence modeling. The solver is based on the finite volume method to build the spatial discretisation of the transport equations.
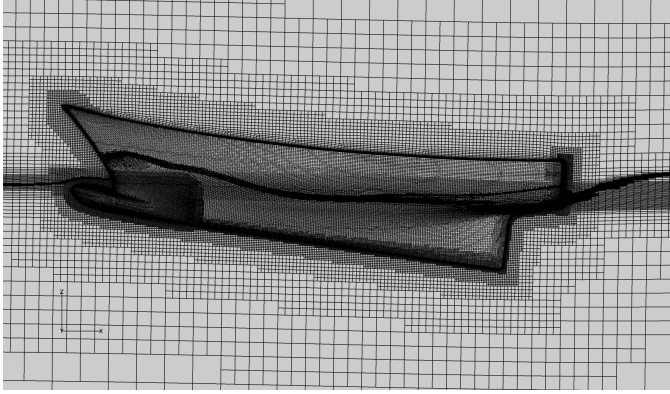
Figure 17: View of the mesh around the hull with free free surface deformation

The unstructured discretisation is face-based, which means that cells with an arbitrary number of faces are accepted. This makes the solver ideal for adaptive grid refinement, as it can perform computations on locally refined grids without any modification. Free-surface flow is simulated with a volume of fluid approach: the water surface is captured with a conservation equation for the volume fraction of water, discretised with specific compressive discretisation schemes, [38]. The vessel's dynamic trim and sinkage are resolved during the simulation.

The studied trawler has a waterline length of 22.35 metres and a displacement of 150 metric tons. Simulations are done at a speed of 13 knots (6.688m/s), corresponding to a a Froude number of 0.4517. Trim and sinkage are solved. The hull forward motion is imposed by accelerating it to its final speed with a $\frac{1}{4}$ sinusoidal ramp. Fluid characteristics are shown in Tab.3.

	$\rho\ (kg/m^3)$	$\mu\ (Pa.s)$
Water	1026.02	0.00122
Air	1.2	$1.85 * 10^{-5}$

Table 3: Fluid characteristics

6.2.2. Proposed deformations

The skeleton used for the bulbous bow is illustrate in Fig.2(b).

We propose to vary three parameters to control the shape: the angle, the length, and the width at mid-bow of the bulb. Fig.18 illustrates variations of bulbous bow shape according to the parameters.

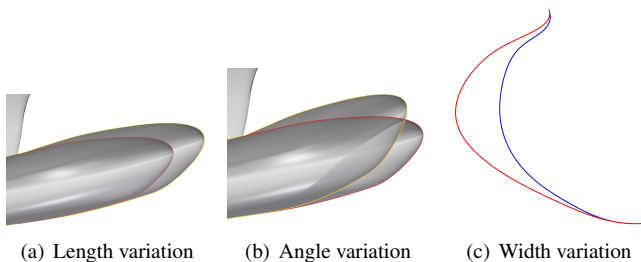(a) Length variation   (b) Angle variation   (c) Width variation

Figure 18: Bulbous bow shape variations

Parameters such as sectional area, vertical position of the centroid, type of sections (delta, oval, nabla) are also important parameters for the shape of the bulbous bow. These parameters are being integrated into the parametric modeller. The current study focuses on the three parameters (angle, length and width), and will be extended in a future work with new parameters.

The variations are distributed in a parameter space defined in Tab.4, according to limits given by architectural criteria.

The initial bulb being quite short, we assumed that shapes with a lower length than 1.86m will not positively influence the drag, likewise we restricted the bulb to not be longer than the extremity of the upper bow. For the angle, we noticed that when the length of the bow is increased, keeping the original value will cause the bulb to pierce the free surface, again this configuration is unwanted.

	Length	Angle	Width
Initial value	1.61m	31.52°	0.83m
Min variation	+15% (= 1.86m)	−25% (= 23.64°)	−20% (= 0.66m)
Max variation	+90% (= 3.07m)	0% (= 31.52°)	+20% (0.99m)

Table 4: Limits of parameters domain

As for the application to the foil, we use a Latin Hypercube distribution to sample the parameter space in order to prepare a relevant dataset for the future use of optimisation algorithms such as EGO.

6.2.3. Results

We used a Latin Hypercube distribution with 20 values. We present in Tab.5 the results of the original hull without bulb, the hull with the initial bulb and the best variation obtained from the parameters variation. The best variation is not a fully optimised bulbous bow, as no optimisation algorithm has yet been used for this application. It is planned to use EGO [35] on the distribution obtained through this study as a future work.

The best drag reduction is obtained with the following parameter values : Length: +58.70% (= 2.56m) ; Angle: −19.81% (= 25.28°) ; Width: +9.99% (= 0.66m). The total drag F_x is composed of the friction resistance $F_{Fric,x}$ component and the pressure resistance $F_{Pres,x}$ component. The friction resistance is obtained by integrating the tangential component of stresses over the wetted surface of the hull, in the direction of motion, and the pressure resistance is obtained by integrating the normal component, in the direction of motion. Tab.5 presents the different components of resistance computed by ISIS-CFD and a comparison with ITTC-57 friction resistance computed with dynamic wetted surface. There is no correlation allowance used as the computation is performed at full scale.

In other terms, the best variation represents an improvement of 3.64% from the first bulb design. The pressure resistance represents an important part of the total drag of the original hull.

13

	Original hull (without bulb)	Initial bulb	Best variation
Static wetted surface in m^2	183.6	190.1	193.7
Dynamic wetted surface in m^2	195.4	200.7	204.2
Total drag F_x in N	79910	73740	71054
Pressure resistance $F_{Pres,x}$ in N	70599	63852	60970
Friction resistance $F_{Fric,x}$ in N	9311	9887	10083
ITTC-57 friction resistance computed with dynamic wetted surface in N	9042	9140	9445
% reduction of drag F_x	-	7.72%	11.08%

Table 5: Drag results and variations on the bulbous bow

We can observe the positive impact of the bulbous bow on this component. It reduces the pressure resistance by modifying the shape of the bow wave (see Fig.19). The friction resistance increases with the wetted surface. Despite that the static/dynamic wetted surface increases due to the additional surface of the bulbous bow, the total resistance decreases thanks to its effect on the flow.

Trim affects the drag by modifying the flow around the ship hull. The volume distribution of the bulbous bow is not preserved during the shape deformation, but the total displacement and location of the center of gravity are kept identical for each design.

Tab.6 shows the evolution of the total trim and sinkage of the three hulls. The total trim (respectively sinkage) is the sum of the hydrostatic the dynamic trim (respectively sinkage). The variation of total trim is relativity small between the different designs. As trim of the original hull and best variation are similar, the effect on the drag is mostly produced by the modification of the shape of the bulbous bow.

	Total trim in deg	Total sinkage in m
Original hull (without bulb)	1.656	-0.257
Initial bulb	1.495	-0.264
Best variation	1.728	-0.225

Table 6: Trim and sinkage results

Fig.19 illustrates the free surface elevation of the two designs, Initial bulb and Best variation, and Fig.20 show the wave patterns.

The sampling we performed with the Latin Hypercube is represented graphically with a response surface method, illustrated in Fig.21. Figure 21(a) represents cutting planes of the design space, showing two main local minima. In Figure 21(b), we show iso-values of the total drag F_x. We can identify a region where the objective function is predicted to be smaller than in the other parts of parameter domain.

Further investigations may lead to finding better drag reduc-
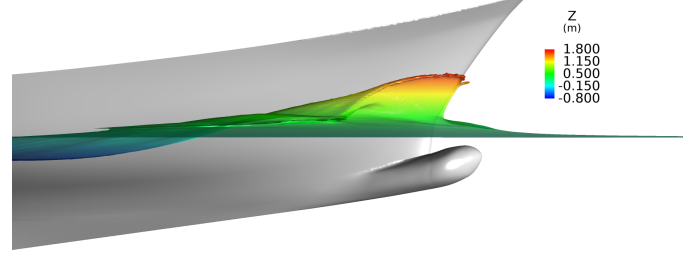
(a) Free surface elevation for the Initial bulb
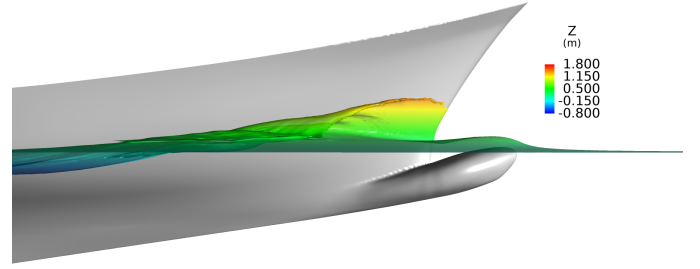
(b) Free surface elevation for the Best variation

Figure 19: Free surface elevation

tion results by using an adapted optimisation algorithm based on Kriging such as EGO to find minima using the model built from the response surface.

7. Conclusion and future work

This paper presents a method for parametrizing and deforming different type of shapes with a skeleton-based approach. The methodology we develop reduces the number of degrees of freedom thanks to observer functions described with B-Splines and provides a fine control of the geometry in terms of architectural parameters. Our tool can handle any shape that can be described with the skeleton-based parametrization.

Our parametric modeller allows to explore the domain of possible shapes in an efficient way, and allows to determine improvements of the design that are architecturally relevant.

As shown by the experiments, we are able to improve the hydrodynamic performances of a AC45 foil and a bulbous bow, with a few number of parameters.

Further work will focus on handling more complex geometries with the skeleton representation. Section curves with multiple components and branching curves will be considered.
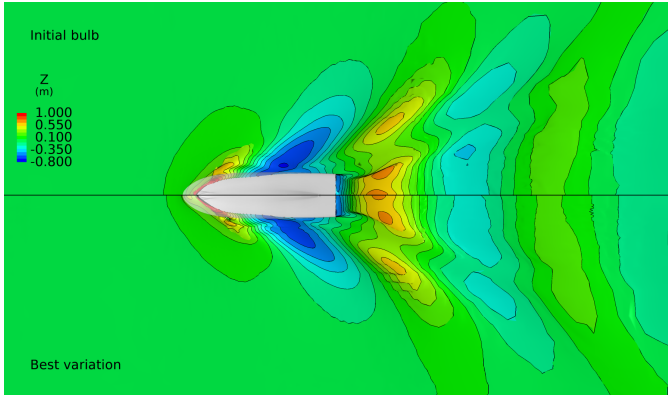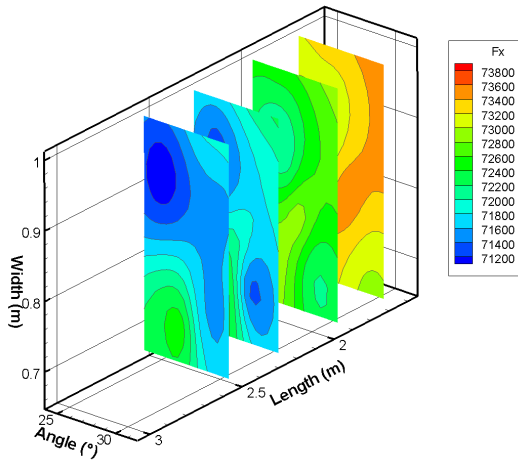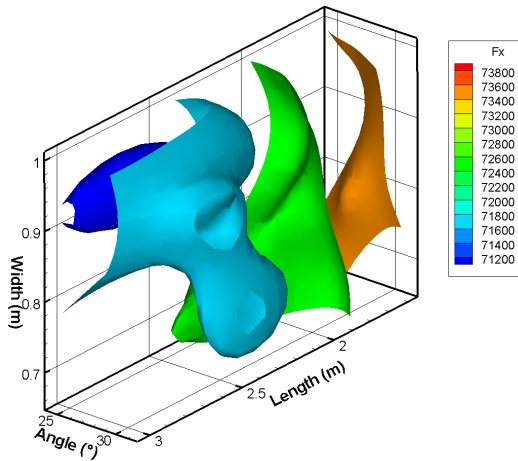
Figure 20: Wave generation of the initial bulb (upper part) and the best variation (lower part) designs

(a) Cutting planes of the response surface

(b) Iso values of the total drag F_x in the response surface

Figure 21: View of the response surface

We will also develop the link with optimisation algorithm solvers. A fully automatised optimisation loop will be developed. Sensitivity of the simulation results to parameters will be taken into account in order to reduce the degrees of freedom as much as possible.

References

[1] Brizzolara, S., Vernengo, G., Pasquinucci, C.A., Harries, S.. Significance of parametric hull form definition on hydrodynamic performance optimization. In: 6th International Conference on Computation Mehods in Marine Engineering (Marine 2015). 2015, p. 254–265.

[2] Peri, D., Rossetti, M., Campana, E.. Design optimization of ship hulls via cfd techniques. Journal of Ship Research 2001;10:140–149.

[3] Blanchard, L., Berrini, E., Duvigneau, R., Roux, Y., Mourrain, B., Jean, E.. Bulbous bow shape optimization. In: 5th International Conference on Computation Mehods in Marine Engineering (Marine 2013). 2013, p. 412–423.

[4] Jones, D.R., Schonlau, M., Welch, W.J.. Efficient global optimization of expensive black-box functions. J of Global Optimization 1998;13(4):455–492.

[5] Hansen, N.. The CMA Evolution Strategy: A Comparing Review; vol. 192; chap. Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms. Springer Berlin Heidelberg; 2006, p. 75–102.

[6] Duvigneau, R., Chandrashekar, P.. Kriging-based optimization applied to flow control. International Journal for Numerical Methods in Fluids 2012;69(11):1701–1714.

[7] Piegl, L., Tiller, W.. The NURBS Book (2Nd Ed.). New York, NY, USA: Springer-Verlag New York, Inc.; 1997.

[8] Ju, T., Schaefer, S., Warren, J.D.. Mean value coordinates for closed triangular meshes. ACM Trans Graph 2005;24(3):561–566.

[9] Kang, J., Lee, B.. Geometric interpolation and extrapolation for rapid generation of hull forms. In: COMPIT. 2012, p. 202–212.

[10] Hock, J., Goh, C., Li, Y.. Hybrid evolutionary shape manipulation for efficient hull form design optimisation. In: COMPIT. 2016, p. 264–279.

[11] Peri, D., Diez, M.. Robust design optimization of a monohull for wave wash minimization. In: 5th International Conference on Computation Mehods in Marine Engineering (Marine 2013). 2013, p. 89–100.

[12] Duvigneau, R., Visonneau, M.. Shape optimization strategies for complex applications in computational fluid dynamics. In: 2nd International Conference on Computer Applications and Information Technology in the Maritime Industries. 2003, p. 1–8.

[13] Raven, H.C., Hoekstra, M.. A practical system for hydrodynamic optimizationof ship hull forms. VNSI Innovatiedag, Wageningen 2003;:1–7.

[14] Yoshizawa, S., Belyaev, A.G., Seidel, H.P.. Skeleton-based variational mesh deformations. Comput Graph Forum 2007;26(3):255–264.

[15] Zhou, K., Huang, X., Xu, W., Guo, B., Shum, H.Y.. Direct manipulation of subdivision surfaces on GPUs. ACM Trans Graph 2007;26(3):Article 91.

[16] Skouras, M., Thomaszewski, B., Bickel, B., Groos, M.. Computational design of rubber balloons. Comput Graph Forum 2012;31(2pt4):835–844.

[17] Schafer, H., Keinert, B., Niessner, M., Buchenau, C., Guthe, M., Stamminger, M.. Real-time deformation of subdivision surfaces from object collisions. In: Proceedings of the 6th High-Performance Graphics Conference. EG; 2014, p. 1–8.

[18] Allaire, G., Jouve, F., Toader, A.M.. A level-set method for shape optimization. C R Acad Sci Paris 2002;334(12):1125–1130.

[19] Aguilar, J.C.. Optimisation de formes hydrodynamiques. couche limite intrinseque tridimensionnelle. Ph.D. thesis; Ecole des Mines de Paris; 1996.

[20] Guido, Y.. Controle et optimisation de forme dans les equations de navier-stokes. Ph.D. thesis; Ecole des Mines de Paris; 1997.

[21] Ginnis, A., Duvigneau, R., Politis, C., Kostas, K., Bellibassakis, K.,

15

Gerostathis, T., et al. A Multi-Objective Optimization Environment for Ship-Hull Design Based on a BEM-Isogeometric Solver. In: 5th International Conference on Computational Methods in Marine Engineering. Hamburg, Germany; 2013, p. 1–12.

[22] Kostas, K., Ginnis, A., Politis, C., Kaklis, P.. Ship-hull shape optimization with a t-spline based bemisogeometric solver. Computer Methods in Applied Mechanics and Engineering 2015;284:611–622. Isogeometric Analysis Special Issue.

[23] Papanikolaou, A., Harries, S., Wilken, M., Zaraphonitis, G.. Integrated ship design and multiobjective optimization approach to ship design. In: International Conference on Computer Applications in Shipbuilding. 2011, p. 1–12.

[24] Jacquin, E., Derbanne, Q., Cordier, S., Alessandrini, B.. Hull form optimization using a free surface ranse solver. In: 25th Symposium on Naval Hydrodynamics. 2004, p. 1–14.

[25] Wang, W., Pottmann, H., Liu, Y.. Fitting b-spline curves to point clouds by curvature-based squared distance minimization. ACM Trans Graph 2006;25(2):214–238.

[26] Mukesh, R., Lingadurai, K., Selvakumar, U.. Airfoil shape optimization using non traditional optimization technique and its validation. Journal of King Saud University, Engineering Sciences 2014;26(2):191–197.

[27] Kracht, A.. Design of bulbous bows. SNAME Transactions 1978;86:197–217.

[28] Nocedal, J., Wright, S.J.. Numerical optimization. Springer Series in Operations Research and Financial Engineering; Berlin: Springer; 2006.

[29] Tang, C., Sun, X., Gomes, A., Wallner, J., Pottmann, H.. Form-finding with polyhedral meshes made simple. ACM Trans Graph 2014;33(4):70:1–70:9.

[30] Bo, P., Ling, R., Wang, W.. A revisit to fitting parametric surfaces to point clouds. Computers & Graphics 2012;36(5):534–540.

[31] Durand, M., Leroyer, A., Lothodé, C., Hauville, F., Visonneau, M., Floch, R., et al. Fsi investigation on stability of downwind sails with an automatic dynamic trimming. Ocean Engineering 2014;90:129–139. Innovation in High Performance Sailing Yachts - INNOVSAIL.

[32] Lothodé, C., Durand, M., Roux, Y., Leroyer, A., Visonneau, M., Dorez, L.. Dynamic fluid structure interaction of a foil. In: Innov'Sail 2013, 26th - 28th June. 2013, p. 1–6.

[33] Faltinsen, O.M.. Hydrodynamics of High-Speed Marine Vehicles. Cambridge University Press; 2006.

[34] Iman, R.L., Helton, J.C., Campbell, J.E.Y.. An approach to sensitivity analysis of computer models, part 1. introduction, input variable selection an preliminary variable assessment. Journal on Quality Technology 1981;13(3):174–183.

[35] Jones, D.R.. A taxonomy of global optimization methods based on response surfaces. Journal of Global Optimization 2001;21:345–383.

[36] Wackers, J., Deng, G., Leroyer, A., Queutey, P., Visonneau, M.. Adaptive grid refinement algorithm for hydrodynamic flows. Computers & Fluids 2012;55:85–100.

[37] Duvigneau, R., Visonneau, M., Deng, G.. On the role played by turbulence closures in hull shape imization at model and full scale. J Marine Science and Technology 2003;8(1):11–25.

[38] Queutey, P., Visonneau, M.. An interface capturing method for free-surface hydrodynamic flows. Computers & Fluids 2007;36(9):1481–1510.