



**HAL**  
open science

## Geometric modelling and deformation for shape optimization of ship hulls and appendages

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

► **To cite this version:**

Elisa Berrini, Bernard Mourrain, Yann Roux, Mathieu Durand, Guillaume Fontaine. Geometric modelling and deformation for shape optimization of ship hulls and appendages. 2016. hal-01373249v1

**HAL Id: hal-01373249**

**<https://hal.science/hal-01373249v1>**

Preprint submitted on 28 Sep 2016 (v1), last revised 11 May 2017 (v4)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Geometric modelling and deformation for shape optimization of ship hulls and appendages

Elisa Berrini<sup>a,b</sup>, Bernard Mourrain<sup>a</sup>, Yann Roux<sup>b,c</sup>, Mathieu Durand<sup>c</sup>, Guillaume Fontaine<sup>c</sup>

<sup>a</sup> Université Côte d'Azur, INRIA Sophia-Antipolis Méditerranée, 2004 route des Lucioles, 06902 Sophia-Antipolis, France

<sup>b</sup> MyCFD, 29 Avenue des frères Roustan, 06220 Golfe-Juan, France

<sup>c</sup> 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 optimization loop, which will repeatedly run numerical simulations and deform the geometry for an automatic search of an optimal shape.

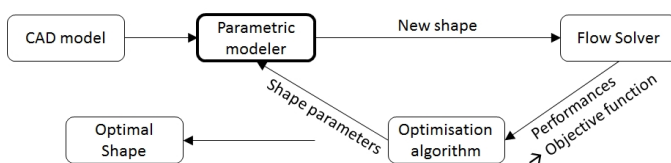


Figure 1: Automatic shape optimisation loop.

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.

A parametric modeller is intuitively strongly linked to the CAD (Computer Aided-Design) software used to build the geometry. However, deforming the control points of the standard NURBS representations used in CAD to generate automatically new shapes is not appropriate. The number of control points to represent adequately the shape may be too large to be used in shape optimisation. Another obstacle is the complexity of the geometric models that can be trimmed, or subdivided into too numerous patches that cannot be deformed in a structured way or that are just not clean enough for CFD computations.

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 gener-

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

ate 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 and on being independent from the CAD software used to design the model by representing objects with a skeleton. Finally, we propose a methodology to link shape representation, shape deformation and numerical simulation.

The motivation of working with architectural parameters is lead by the intuitiveness for an architect to control a shape by such expert variables instead of control points, which have no physical meaning in computations. We propose a way to control shapes efficiently in terms of these architectural parameters, by controlling and deforming the generating curve and the section 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, blades of wind turbines, 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 :

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

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.

Shape deformation of ships 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 animations purposes, and they have been applied to shape optimisation for ships. Close 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 method 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 possibly to non-realistic results.

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, thus the 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 optimize 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*<sup>TM</sup> or *Grasshopper* for *Rhinoceros 3D*<sup>TM</sup>. They have been combined with isogeometric flow solvers for ship hull optimization, for instance in [21]. Specific software have been developed during the last decades for ship applications. One of the most widespread is *CAESES* from *Friendship System*<sup>TM</sup>, allowing the user to create geometries using advanced parameters that can be modified easily by hand or automatically with a CFD optimisation loop [22]. The shape can be controlled either by parameters introduced in the design process or by curves or surfaces around the object called delta-shift. Points of the object are linked to the delta-shift, and follow the deformations of the delta-shift curves or surfaces.

Similarly, a ship dedicated tool *Bataos* [23] 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.

### 155 3. Shape parametrization

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

#### 163 3.1. Geometrical parametrization

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

- 166 1. Lines plan are used by naval architects to define the ex-  
167 ternal shape of the boat, as consistent shapes must be ob-  
168 tained once deformed.
- 169 2. Classical and efficient techniques in 3D animation are  
170 based on the deformation of medial axis curves associ-  
171 ated to a shape [14].

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

175 We consider the skeleton as a set of curves composed of a  
176 generating curve and section curves. Each section curve needs  
177 to be identified on the generating curve: a local coordinate sys-  
178 tem, with an origin and a rotation, allows us to know its posi-  
179 tion and orientation. We are going to describe more precisely  
180 this skeleton based representation in the next section and how  
181 the architectural parameters are associated to the geometry in  
182 the following sections.

##### 183 3.1.1. Generating curve and section curves

184 In our skeleton concept, the generating curve describes the  
185 general shape of the object, whereas sections describe more pre-  
186 cisely the outlines of the object around the generating curve,  
187 similarly to the architect's line plan.

188 The generating curve needs to describe the prominent fea-  
189 tures of the object. It is defined to be lying on the geometry and  
190 connects all the section curves. It is not necessarily planar, but  
191 symmetrical considerations of the object allow to describe it as  
192 a planar curve in most cases.

193 Section curves are computed as the intersection curves be-  
194 tween the studied object and a family of planes. To each section  
195 curve, we associate a point on the generating curve, a local co-  
196 ordinate system, an origin and a rotation which allows to know  
197 the position and the orientation of the section curve. The cut-  
198 ting planes are defined to be normal to the tangent vector of the  
199 generating curve at the corresponding point adjusted with the  
200 rotation associated to the section.

201 For practical purposes, we represent the generating curve  
202 and the section curves as B-splines curves with a given num-  
203 ber of control points. We further simplify the representation by  
204 choosing a finite subset of the section curves, associated with a  
205 finite sampling of the generating curve (See Fig.2).

207 This leads to a representation of the geometry in terms of a  
208 finite set of control points. We denote by  $\mathbf{c}_g$  the control points  
209 of the generating curve and by  $\mathbf{c}_i$  the control points of the  $i^{\text{th}}$   
210 sampled section curve for  $i = 1, \dots, N$ .

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

213 To construct a skeleton-based representation from an initial  
214 geometric model, we first choose a relevant generating curve  
215 according to the model. For airfoil based shapes, the trailing  
216 edges is an ideal choice, as is the keeline for a hull. To ob-  
217 tain the section curves, we compute the intersection between  
218 the object and the set of planes defined according to the tan-  
219 gent of the generating curve. If a non-null rotation is associated  
220 to the section, the cutting plane is first transformed according  
221 to this rotation. The planes are sampled along the generating  
222 curve, following a chord length or a curvature based distribu-  
223 tion. At this stage, we obtained a first skeleton from the model.  
224 The number of control points or the quality (continuity, smooth-  
225 ness, etc.) of the curves depends on the original design.

226 Then, we reconstruct new B-Spline curves with a fitting  
227 process [24] from a point cloud sampled on the current sections  
228 and the generating curve. We use a small number of control  
229 points (e.g.  $\leq 10$ ) to represent these curves, that are smoothed  
230 and cleaned. In the applications that we have considered this is  
231 usually enough to ensure a good level of approximation. The  
232 average normalized distance between the intersection curves  
233 and the B-spline section curves is kept under  $10^{-5}$  m.

234 In the Fig.2, we illustrate the skeleton we obtained with  
235 Rhinoceros 3D<sup>TM</sup>. The original model of the AC45 foil model  
236 is made of 22164 control points. Our skeleton representation is  
237 made of only 578 points,  $560 = 2 * 28 * 10$  points for sections  
238 and 18 points for the generating curve.

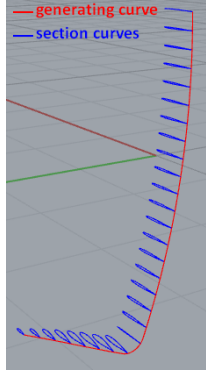
239 For the bulbous bow, the total number of control for the  
240 half-hull is 2574, and 57 are directly linked to the bulbous bow  
241 part. Our model is made of 185 points,  $160 = 16 * 10$  points  
242 for sections and 25 points for the generating curve. We choose  
243 to define more control points linked to the bulb than the origi-  
244 nal model, as we look for a precise control of the shape of the  
245 bulbous bow. Moreover, this representation is temporary as the  
246 total number of parameter that control the shape is then reduced  
247 by the architectural parameters and by the observer functions,  
248 see in sections 3.2 and 3.3.

##### 251 3.1.2. Local coordinate systems for the section curves

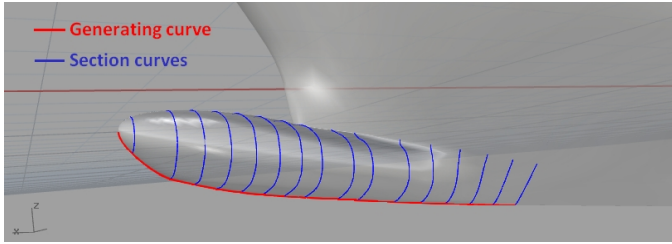
252 Section curves are identified on the generating curve thanks  
253 to a local coordinate system. Each local coordinate system has  
254 its origin defined from a point on the generating curve, allowing  
255 to locate the section in 3D space.

256 The first axis  $U$  is defined into the section plane, its direc-  
257 tion is imposed by a main feature of the section, as the leading  
258 edge for an airfoil section, or the maximum height for a bulbous  
259 bow section.

260 The second axis  $V$  is represented by the tangent of the generat-  
261 ing curve  $T_{\sigma(t)}$  at the origin point of the local coordinate system.



(a) Skeleton of a sail boat's foil



(b) Skeleton of a bulbous bow

Figure 2: Examples of skeletons

263 In most cases, the first and second axes are orthogonal by construction,  
 264 but some sections representing special features of the geometry, as the extremities of the foil, are not defined in the  
 265 plane  $P_T$  orthogonal to  $T_{\sigma(t)}$ . Thus  $U$  and  $V$  are not orthogonal to each other.  
 266  
 267

268 Let  $R_T$  be the rotation that transform  $U$  such as  $U \in P_T$ . We  
 269 apply the inverse of the rotation  $R_T$  to  $T_{\sigma(t)}$  in order to obtain the  
 270 second axis  $V$  orthogonal to  $U$ .  $R_T$  is associated to the section.  
 271

272 Then the third axis  $W$  is computed as the cross product of  
 273 the first two axes.  
 274

275 The implicit definition of the second axis  $V$  allows the local  
 276 coordinate system to move when the generating curve is modified, computing the new orientation of the section auto-  
 277 matically. The translation and rotation matrices that turns the  
 278 original tangent to the new one is applied to the other axis  $U$   
 279 and  $W$  and to the section control points. Therefore modifying  
 280 the generating curve induces affine transformations on the section  
 281 curves, given by the modification of the local coordinate  
 282 system.

### 283 3.2. Architectural parameters

284 Architectural parameters describe the main characteristics  
 285 of the object. They are chosen according to the design practice  
 286 and effects on the object performance. Our goal is to control the  
 287 shape of the studied object through the architectural parameters  
 288 value.

289 We associate different parameters to the generating curve  
 290 and the section curves in order to control the whole shape.

291 For example, the main characteristics of an L-shaped sail  
 292 boat foil are the length of the two parts and the angle between  
 293 them. Then each airfoil profile section has particular features,  
 294 as chord length, thickness, angle of attack, etc. [25]. For a  
 295 bulbous bow, the main features are the length, the angle, the  
 296 height and thickness [26]. We illustrate those parameters in  
 297 Fig.3 and Fig.4. New types of parameters can be implemented  
 298 easily to enrich the model, such as the sectional areas curve, the  
 299 volume of the bulb, etc.

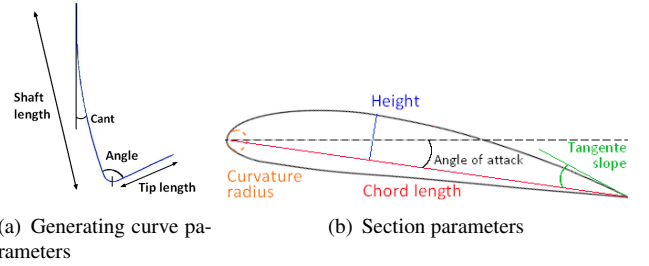


Figure 3: Foil parameters

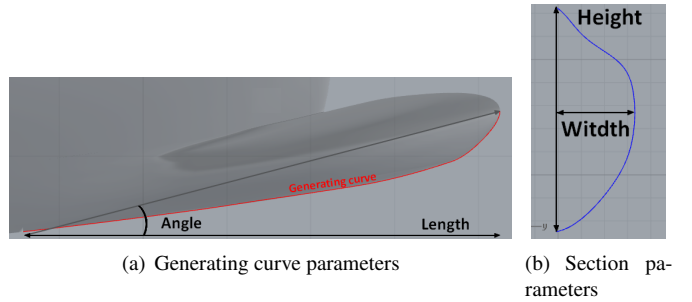


Figure 4: Bulbous bow parameters

### 300 3.3. Observer function

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

310 In practice, these functions will be represented with a B-  
 311 Spline curves passing through the section parameter values ac-  
 312 cording to their position on the generating curve. The B-Spline  
 313 curves belong to a finite dimensional space with a small num-  
 314 ber of control points. These are the parameters that we will use  
 315 to control the shape.

317 An illustration is shown in Fig.5, where the observer func-  
 318 tion made of 13 control points represents the chord length dis-  
 319 tribution of 28 section curves.

320 From this consideration, managing the B-Spline instead of  
 321 each section parameters represent 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.

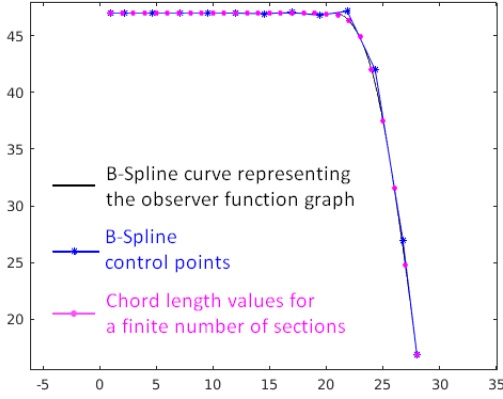


Figure 5: Distribution of the chord length parameter along the generating curve of a foil

#### 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  $\phi : G \rightarrow P$  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:  $\phi^{-1} : P \rightarrow G$ .

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 coordinate 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  $\xi$ . The generating curve parameterized by  $t \in [0, 1]$  is denoted  $\xi_g$ . Its controlled coefficients are  $\mathbf{c}_g$ . The  $i^{\text{th}}$  sampled section curve corresponding to the parameters  $t_i$  on  $\xi_g$  is denoted  $\xi_i$  for  $i = 1, \dots, N$ . It is parameterized by  $s \in [0, 1]$  and its control points are  $\mathbf{c}_i = (\mathbf{c}_{i,0}, \dots, \mathbf{c}_{i,M})$ . In the following paragraphs,  $\xi^0$  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  $V$ :

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

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  $i^{\text{th}}$  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  $\xi$ . 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  $\xi_i$  (or  $\xi_g$ ). 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:

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

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  $\xi_i, i = 1, \dots, N$  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:

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

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

404 introducing stiffness between successive control points  $\mathbf{c}_{i,j}$ . We  
 405 add correction terms to control respectively  $C^1$  and  $C^2$  proper-  
 406 ties of control points.

$$E_0(\mathbf{c}) = \sum_j \|\Delta \mathbf{c}_j\|^2 \quad \Delta \mathbf{c}_j = \mathbf{c}_j - \mathbf{c}_{j-1}$$

$$E_1(\mathbf{c}) = \sum_j \|\Delta^2 \mathbf{c}_j\|^2 \quad \Delta^2 \mathbf{c}_j = \mathbf{c}_{j+1} - 2\mathbf{c}_j + \mathbf{c}_{j-1}$$

407 Finally, the proposed minimisation system is described as  
 408 follows.

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

409 As we decoupled it into a minimization system for each section  
 410 curve  $\xi_i$  and for the generating curve  $\xi_g$ , an optimisation prob-  
 411 lem similar to Eq.4 is solved for the generating curve  $\xi_g$ .

412  
 413 In these formulations,  $\varepsilon$  is a weight allowing to balance the  
 414 influence of the shape control term. In fact, if this term is too  
 415 high, the system will converge to a solution too close to the  
 416 initial curv, and will have difficulty to respect the target param-  
 417 eters.  $\varepsilon$  can be seen as a penalty coefficient, but we chose to  
 418 decrease it at each iteration because in our particular case the  
 419 initial guess, the original curve, needs to be degraded to match  
 420 new architectural parameters. The coefficients  $\lambda_i$  weighing the  
 421 shape constraints and  $\mu_i$  weighing the correction matrices are  
 422 both very small, usually around  $10^{-4}$ .

#### 423 4.2. Numerical solution

424 The problem has a relatively large number of degrees of  
 425 freedom: in most cases we have 2 DOF (Degree Of Freedom)  
 426 per control points (e.g. modification on X and Y direction as  
 427 sections and the generating curve are planar) and usually curves  
 428 have around 10 control points. If the generating curve is not  
 429 planar, then we have 3 DOF per control points.

430 The definition of the problem is well adapted to Sequential  
 431 Quadratic Programming (SQP) [27]. SQP algorithm uses New-  
 432 ton's method to find roots of the gradient. We use finite differ-  
 433 ence to compute the gradients of the system. We start with an  
 434 initial value of  $\varepsilon$  and the original curve as the starting point of  
 435 the algorithm, then we decrease  $\varepsilon$  at each iteration and start the  
 436 SQP again with the last computed curve. The algorithm stops  
 437 when the value of the objective function reaches a fixed thresh-  
 438 old.

439  
 440 An illustration of airfoil deformation is shown in Fig.6. Air-  
 441 foil parameters are described in Fig3(b). In this example only  
 442 height is modified: chord length, curvature radius at leading  
 443 edge and tangent slope at trailing edge are kept identical to the  
 444 original airfoil. The final shape is obtain with 16 iterations for  
 445 the suction face and 10 iterations for the pressure face.

446 With a four-cores HP Probook-450 with a Intel® Core™  
 447 i7-4702MQ CPU 2.20GHZ, RAM 8.00 Go, the total time to  
 448 perform this airfoil deformation (pressure and suction faces) is  
 449 12 seconds.

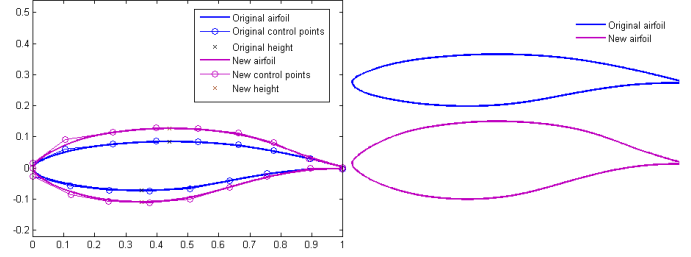


Figure 6: Deformation of the height of an airfoil section, with fixed curvature radius at leading edge and tangent slope at trailing edge

## 450 5. Surface reconstruction

451 The optimisation method outputs deformed sections and gener-  
 452 ating curves, corresponding to the skeleton of a new shape.  
 453 To evaluate the shape's performances with a CFD solver, we  
 454 first need to reconstruct the 3D surface wrapping the deformed  
 455 skeleton. Moreover, building a new surface allows to obtain a  
 456 cleaned-up model for the meshing tool. Classical techniques  
 457 such as *lofting* [7] are a relevant choice for objects that can be  
 458 represented with only one surface such as foils.

459  
 460 For more complex objects, multi-patch surfaces are required.  
 461 In such cases, a particular attention has to be given to the con-  
 462 tinuity between them: for our application, patches have to be  
 463 at least  $C^1$ . We developed a technique based on *form finding*  
 464 [28] to reconstruct suitable surfaces. We expose this technique  
 465 in the following paragraphs.

### 466 5.1. Problem setting

467 Given the section and generating curves of the new deformed  
 468 skeleton, we construct a surface which contains these curves  
 469 and satisfies tangency conditions on the boundaries of the sur-  
 470 face.

471 Like for the skeleton deformation, we compute the surface  
 472 by solving an optimization problem, where the control points  
 473 of the surface  $c_{ij}$  are the unknowns. To define this optimization  
 474 problem, we discretize the problem by sampling the curves. We  
 475 obtain a point set on which we will wrap the surface. The sur-  
 476 face is computed by fitting techniques [29], taking into account  
 477 smoothing and tangency constraints on the boundary of the sur-  
 478 face to ensure the continuity between adjacent surfaces in the  
 479 geometric model. As explained in [28], the constraints we use  
 480 are quadratic in the control point coordinates  $c_i$ . We describe  
 481 them in the following paragraphs.

482  
 483 First, we define for each point of the point set a coordinate  
 484 mapping:

$$\mathbb{R}^3 \longrightarrow [0, 1] \times [0, 1]$$

$$P_l \longrightarrow (u_l, v_l), \quad l = 0, \dots, N_P$$

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

## 487 Surface fitting

488 This constraint ensure that the surface  $\sigma$  passes thought the  
489 points  $P_l$ :

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

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

## 490 Tangency constraint with fix parts of the object

491 Let  $n_l$  be the normal at  $P_l$  of the fix surface adjacent to a sur-  
492 face we want to reconstruct. In the  $u$  direction, the continuity  
493 constraint is expressed by:

$$E_{fixT} : \langle \sigma_u(u_l, v_l) \cdot n_l \rangle^2 = 0, \quad l = 0, \dots, N_P \quad (6)$$

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

494 We have similar constraints in the  $v$  direction.

## 495 Tangency constraint with mobile parts

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

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

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

503

Moreover, the normal vector  $n_k$  must satisfy:

$$E_{mobile normals} : \langle n_k \cdot n_k \rangle^2 = 1, \quad k = 0, \dots, N_n \quad (8)$$

504 Notice that these constraints require an initial value of  $\sigma_1, \sigma_2$   
505 and  $n_k$ .

506

## 507 Regularization

508 A regularization energy term can also be introduced for the sur-  
509 faces, to improve the "fairness" of the surface. It is a quadratic  
510 function of the unknowns control coefficients  $c_{i,j}$ , similar to the  
511 regularization term for curves used in Section 4.1. We do not  
512 detail it here (see for instance [28]).

513

## 514 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:

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

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

517

The general form of quadratic constraints that we treat is:

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

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

522

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:

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

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

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:

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

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

526 This technique is able to reconstruct efficiently and accu-  
527 rately surfaces, which contain the skeleton curves and satisfy  
528 tangency constraints on the boundary.

529

530 We illustrate it with the reconstruction of the surface of a  
531 sail boat hull, in Fig.7. In this example, we choose to recon-  
532 struct the middle part of the hull with two surfaces. Each sur-  
533 face has to be smoothly connected to a fix part of the hull (the  
534 transom or the stem) and to the other middle surface. The algo-  
535 rithm converges in 5 iterations, and the resulting surfaces satisfy  
536 the tangency constraints at the three junction curves.

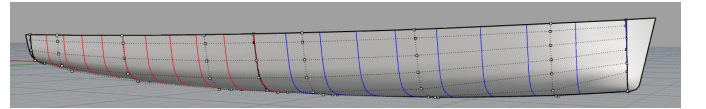


Figure 7: Patch of surfaces reconstructed of a sail boat hull

## 537 6. Applications

538 In this section we present two different applications of the  
539 parametric modeller, one on the foil of an AC45 and one on  
540 the bulbous bow of a fishing trawler. In both cases, we aim  
541 to increase a performance criterion with shape variations. The  
542 parametric modeller is automatized and linked to a flow solver.  
543 A specific flow solver is used for each application: for the foil,  
544 we use a potential flow solver and for the bulbous bow we use  
545 RANS.

546



547 Timing of the code refer to a four-cores HP Probook-450  
548 with a Intel® Core™ i7-4702MQ CPU 2.20GHZ, RAM 8.00  
549 Go.

550

### 551 6.1. Application on a sail boat's foil: AC45

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

556

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

560 While sailing, the foil allows the boat to fly as shown in  
561 Fig.8. However, to maintain this flying state, the stability of the  
562 foil is a critical aspect for both security and performance.

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

565

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

568 For this application, we aim to optimize the shape of the  
569 AC45 foil in order to decrease its total drag while keeping sta-  
570 bility and the ease of use as high as possible. The foil per-  
571 formances are computed with the potential flow solver ARA-  
572 VANTI.

573

574 The AC45 foil is currently used by the Groupama Team  
575 France sailing team for the 35<sup>th</sup> America's Cup. An illustration  
576 of the sail boat flying thanks to the foil is shown in Fig.8, one  
577 foil in the water (right) and the other one visible in the retracted  
578 position (left).



Figure 8: Illustration of the AC45 on the Groupama Team France sail boat,  
Credit: © Eloi Stichelbaut / Groupama Team France

#### 579 6.1.1. Simulation with ARAVANTI

580 ARAVANTI, the flow code used in the present study is de-  
581 veloped and commercialized by the company K-Epsilon. ARA-  
582 VANTI is a coupled fluid-structure solver, with a finite element

583 method for solving the structure and multiple different methods  
584 for the fluids (e.g. vortex line method, particle method, panel  
585 method, etc.).

586 The method used here is a vortex line method with solved  
587 wake. ARAVANTI is coupled to XFOIL in order to incorporate  
588 the flow behaviour, laminar or separated.

589

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

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

602

603 In our specific case for AC45 foil study, only the underwa-  
604 ter part of the foil is simulated. The influence of the free sur-  
605 face is taken into account with an anti-symmetry plane model.  
606 This model is a satisfying approximation for high speed. As  
607 [30] suggests, with a Froude number greater than 1, an infinite  
608 Froude number free-surface condition can be used. In our case,  
609 the Froude number is around 5.45.

610

611 We illustrate in Fig.9 the wake computed with ARAVANTI  
612 and the vortex lines. The vortex line is located at 25% of the aft  
613 of the leading edge along the foil. From the vorticity repartition  
614 colormap, we see that the parts of the AC45 which generate  
615 most of the force allowing to lift up the boat are the knee and  
616 the tip.

617 The reference frame is defined as follows: X is in the oppo-  
618 site direction of the flow, Z is in the vertical direction (oriented  
619 upwards) and Y is horizontal, perpendicular to X.

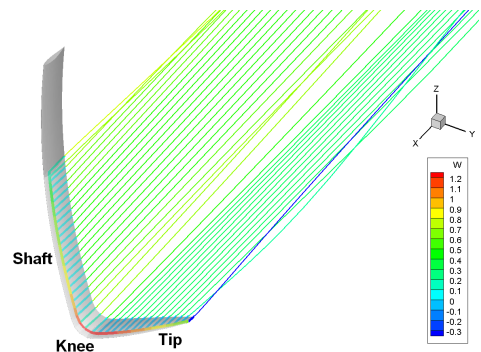


Figure 9: Illustration of the wake and vortex line on the AC45

620

#### 579 6.1.1. Simulation with ARAVANTI

580 ARAVANTI, the flow code used in the present study is de-  
581 veloped and commercialized by the company K-Epsilon. ARA-  
582 VANTI is a coupled fluid-structure solver, with a finite element

#### 621 6.1.2. Proposed performances criteria

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

- 624 1. The total drag  $F_x$  of the foil in the reference frame. A  
625 low drag increases the total performance and speed of the  
626 boat.
- 627 2. A stability criterion, represented by  $\frac{\partial F_z}{\partial z}$ , where  $F_z$  is the  
628 total force in the  $z$  direction of the foil. The aim of this  
629 criterion is to ensure that the boat will stay at a fixed  $z$   
630 height thanks to a self adjusting  $F_z$  balancing the vertical  
631 movements of the foil.
- 632 3. A stability and usage criterion, represented by  $\frac{\partial rake}{\partial V}$ , where  
633 the *rake* is the angle of incidence of the foil in the  $Y$  ro-  
634 tation, and  $V$  is the boat speed. The rake is a parameter  
635 that the crew have to adjust while sailing to modify the  
636 vertical forces  $F_z$ . Thus a foil shape where this param-  
637 eter does not change a lot when the speed is varying is  
638 valuable.

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

645  $F_x$  is computed during the simulation, and we aim to de-  
646 crease it as much as possible. In the reference frame we used,  
647  $F_x$  is oriented along the negative  $x$  direction. Thus, the sign of  
648  $F_x$  will be negative, but we can consider the absolute value to  
649 compare the foil performance.

651 To compute the second criterion, we estimate  $\frac{\partial F_z}{\partial z}$  with fi-  
652 nite differences. We vary the foil displacement by a small  $\Delta z$   
653 and compare the computed  $F_z$ . To be stable, the foil has to gen-  
654 erate a  $F_z$  opposed to the direction of the displacement. Thus  
655 the ratio  $\frac{\partial F_z}{\partial z}$  has to be negative and as large as possible.  
656 For example, if the boat is going too high above the water sur-  
657 face, the foil force  $F_z$  has to decrease in order to make the whole  
658 system lower.

661 We use the same process for the third criterion,  $\frac{\partial rake}{\partial V}$ , by  
662 solving the rake angle for a small speed variation  $\Delta V$ . Here,  
663 the rake has to increase as little as possible when the speed in-  
664 creases. Thus the ratio  $\frac{\partial rake}{\partial V}$  has to be positive and as small as  
665 possible.

667 For both case, we ensure that the finite difference is a sat-  
668 isfying approximation by choosing appropriate steps  $\Delta z$  and  $\Delta V$ .

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

### 674 6.1.3. Proposed deformations

675 We identified the most relevant parameters that influence a  
676 foil performances as the tip length, the angle between the shaft  
677 and the tip and the cant angle, illustrated in Fig.10. Here, we  
678 consider the cant angle as a shape parameter and not as control

679 parameter of sailing.

681 To generate a new CAD from the original CAD model, our  
682 tools takes on average 12 seconds to build the skeleton, 5.1 sec-  
683 ond for the generating curve deformation and 5 seconds for the  
684 section curve deformation. In our case, we perform only de-  
685 formation of the generating curve. Moreover there is no need  
686 to build a new surface around the skeleton, as ARAVANTI does  
687 not require a continuous surface as an input. Only a set of points  
688 distributed on the section curves of the skeleton is sufficient.  
689 The skeleton we used on the AC45 is illustrate in Fig.2(a).

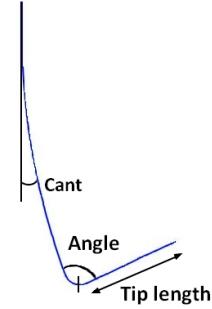


Figure 10: Foil shape parameters

690 The variations of the parameters are distributed in a param-  
691 eter space defined in Tab.1, and illustrated in Fig.11.

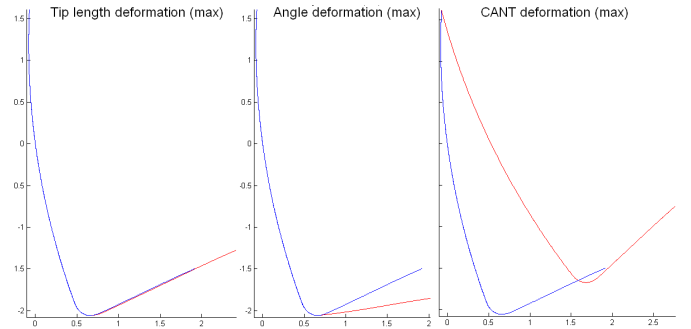


Figure 11: Shape variation of the foil in the paramter space

	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

692 To sample the parameter space, we use a Latin Hypercube  
693 distribution [31]. Our choice is based on the future use of op-  
694 timisation algorithms such as EGO, that are often initialized  
695 with such parameter space values distributions as they are well  
696 adapted for response surface methods [32].

### 698 6.1.4. Results

699 We used a Latin Hypercube distribution with 20 points to  
700 sample the parameter space described in Tab.1. For each set of

701 parameters, we build a new corresponding foil with our para-  
 702 metric modeller and we evaluate automatically the value of the  
 703 3 criteria,  $F_x$ ;  $\frac{\partial F_z}{\partial z}$ ;  $\frac{\partial rake}{\partial V}$ , with ARAVANTI.

704

705 As our aim is to reduce the total drag as much as possible  
 706 while keeping stability criterion as high as possible, the optimal  
 707 solution is located on a Pareto front. We represented the Pareto  
 708 fronts of the drag with each stability criterion in Fig.12.

709

710 We named the foils on Pareto fronts (A,B,C,D), Foil A be-  
 711 ing the one with the least drag and worst stability, Foil D being  
 712 the one with the must drag, but the best stability and Foils B &  
 713 C being in between. Even if Foil A has the worst stability of the  
 714 Pareto front, it is still better than the original AC45. The other  
 715 criteria vary around the original values.

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

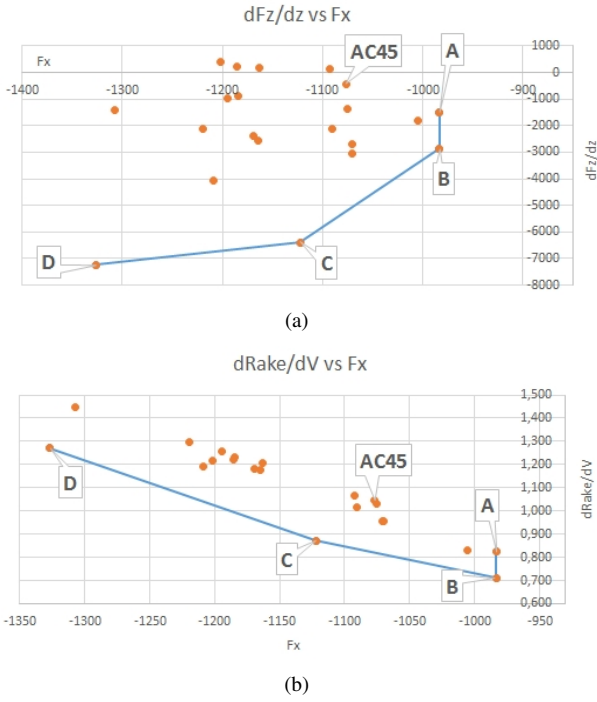


Figure 12: Pareto fronts

718 We detail the points on the Pareto fronts in Tab.2, with the  
 719 initial AC45 results for comparison. We illustrate the results in  
 720 Fig.13.

721

#	% 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	0% (= 1.37m)	0% (= 77.24°)	0% (= 2.42°)	1077	-423	1.045
A	-8.62% (= 1.25m)	+18.58% (= 91.6°)	+650.53% (= 18.16°)	<b>983</b> (+8.68%)	-1495 (+253.28%)	0.825 (+21.07%)
B	+34.02% (= 1.84m)	+9.74% (= 84.76°)	+416.79% (= 12.51°)	983 (+8.68%)	-2863 (+576.44%)	0.710 (+32.04%)
C	+31.56% (= 1.80m)	-5.96% (= 72.64°)	+591.87% (= 16.74°)	1122 (-4.22%)	-6392 (+1410.41%)	0.872 (+16.58%)
D	+11.88% (= 1.53m)	-29.63% (= 54.35°)	+721.38% (= 19.88°)	1327 (-23.22%)	<b>-7240</b> (+1610.95%)	1.271 (-21.64%)

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

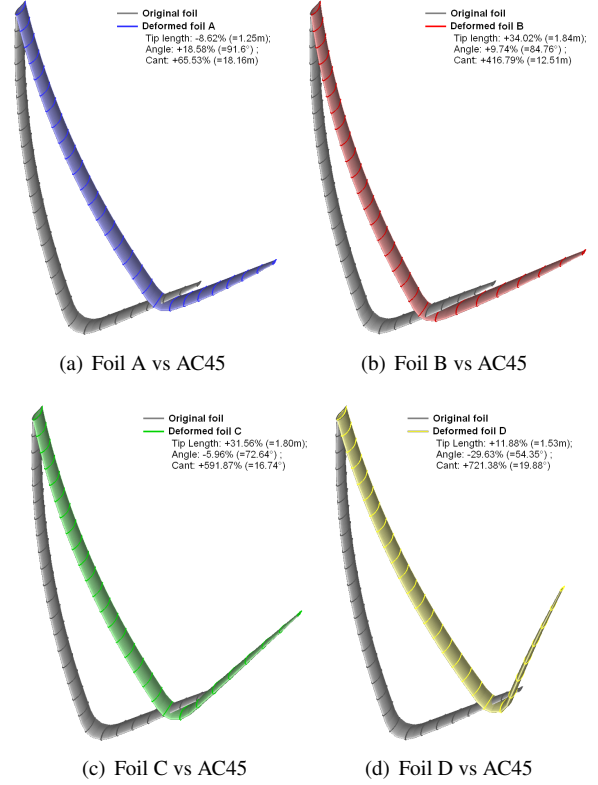


Figure 13: Shape variations on Pareto fronts

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

727 The two extreme shapes in the Pareto front  $\frac{\partial F_z}{\partial z}$  vs  $F_x$ , Foil  
 728 A and D, show a very different behaviour of the foil according  
 729 to the parameters, illustrated in Fig.14 where we see the vortic-  
 730 ity distribution along the foil. In the case of Foil A (Fig.14(b))  
 731 the vorticity is uniformly distributed on the shaft, knee and tip.  
 732 Whereas for Foil D (Fig.14(d)), the vorticity is essentially lo-  
 733 cated on the shaft, thus the lifting force is principally generated  
 734 from this part.

735

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

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

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

748

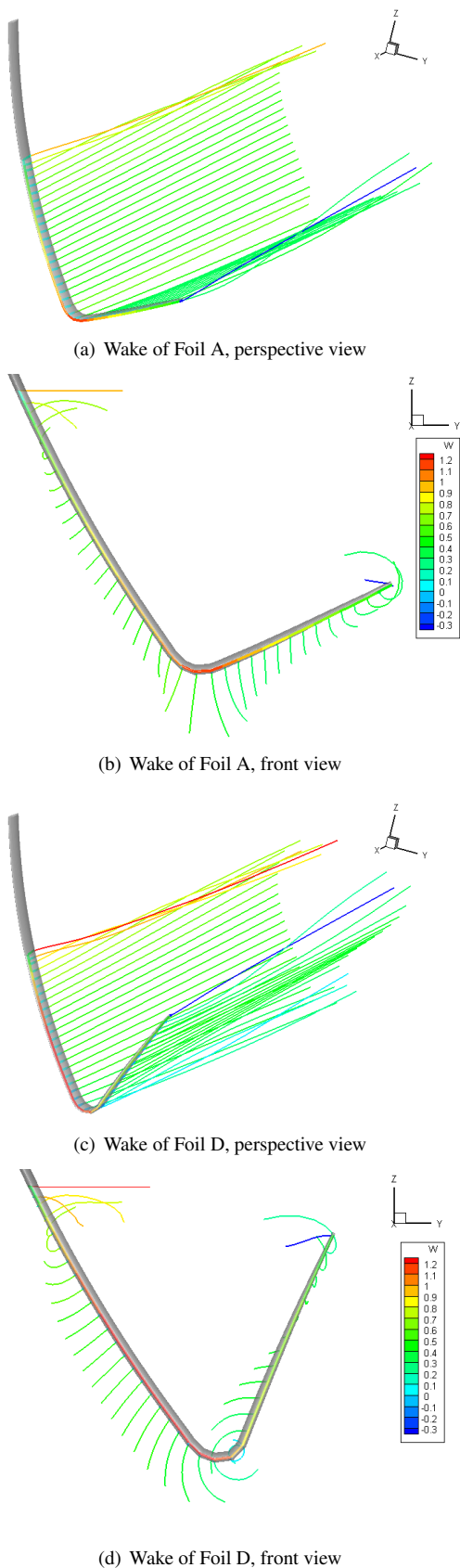


Figure 14: Wake of Foils A and D

## 6.2. Application for a bulbous bow

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

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.

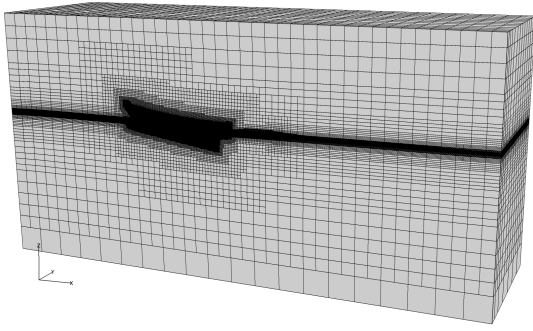
### 6.2.1. Simulation with FINE<sup>TM</sup>/Marine

To generate non-conformal, fully hexahedral, unstructured meshes for complex arbitrary geometries, we use HEXPRESS<sup>TM</sup> from Numeca International. The advanced smoothing capability provides high-quality boundary layers insertion [33]. The software HEXPRESS<sup>TM</sup> 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.15.

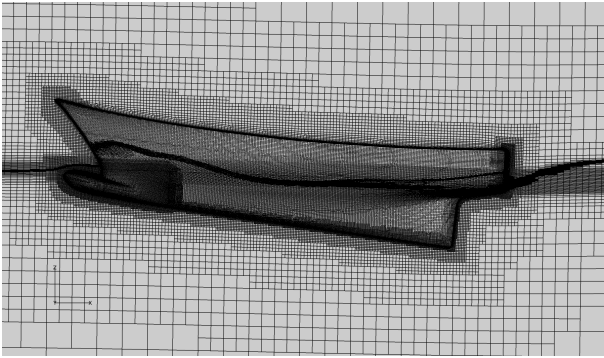
The grid generation process requires a clean and closed geometries to provide robust meshes. Thanks to the shape consistency control and the smooth reconstruction of surfaces, the modeler generates shapes which are well-adapted to these requirements and which allow to produce high-quality meshes for 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 approximately to 2.2 million cells, for a half hull mesh. Fig.15(a) shows a view of the whole grid and Fig.15(b) 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<sup>TM</sup>/Marine computing suite. It is an incompressible, unsteady Reynolds-averaged Navier-Stokes (RANS) solver [34, 35]. For the turbulent flow, additional transport equations for the modeled variables are discretized and solved. The two-equation  $k-\omega$  SST linear eddy-viscosity model of Menter is used for turbulence modeling. The solver is based on the finite



(a) General view of the mesh and the computational domain



(b) View of the mesh around the hull with free free surface deformation

Figure 15: View of the mesh

804 volume method to build the spatial discretisation of the trans-  
805 port equations.

806 The unstructured discretisation is face-based, which means that  
807 cells with an arbitrary number of faces are accepted. This makes  
808 the solver ideal for adaptive grid refinement, as it can perform  
809 computations on locally refined grids without any modification.  
810 Free-surface flow is simulated with a volume of fluid approach:  
811 the water surface is captured with a conservation equation for  
812 the volume fraction of water, discretised with specific compressive  
813 discretisation schemes, [35]. The vessel's dynamic trim  
814 and sinkage are resolved during the simulation.

815  
816 The studied trawler has a waterline length of 22.35 metres  
817 and a displacement of 150 metric tons. Simulations are done at  
818 a speed of 13 knots (6.688m/s). Trim and sinkage are solved,  
819 while the hull speed is imposed according to a  $\frac{1}{4}$  sinusoidal  
820 ramp law. 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

### 821 6.2.2. Proposed deformations

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

823 We propose to vary three parameters to control the shape:  
824 the angle, the length, and the width at mid-bow of the bulb.

825

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

828

829 The initial bulb being quite short, we assumed that shapes  
830 with a lower length than 1.86m will not positively influence  
831 the drag, likewise we restricted the bulb to not be longer than  
832 the extremity of the upper bow. For the angle, we noticed that  
833 when the length of the bow is increased, keeping the original  
834 value will cause the bulb to pierce the free surface, again this  
835 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

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

### 840 6.2.3. Results

841 We used a Latin Hypercube distribution with 20 values. We  
842 present in Tab.5 the results of the original hull without bulb, the  
843 hull with the initial bulb and the best variation obtained from  
844 the parameters variation.

845

846 The best drag reduction is reached at the following param-  
847 eter values : Length: +58.70% (= 2.56m) ; Angle: -19.81%  
848 (= 25.28°) ; Width: +9.99% (= 0.66m).

	Drag ( $F_x$ ) in N	% difference
Original hull (without bulb)	79910	-
Initial bulb	73740	7.72%
Best variation	71054	11.08%

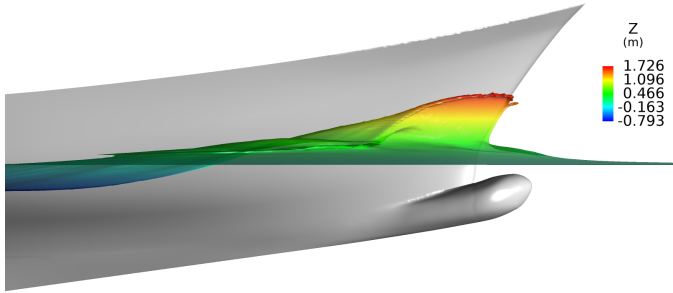
Table 5: Drag results and variations on the bulbous bow

849 In other terms, the best variation represents a save of 3.64%  
850 from the first bulb design. We illustrate the free surface eleva-  
851 tion of these two cases in Fig.16.

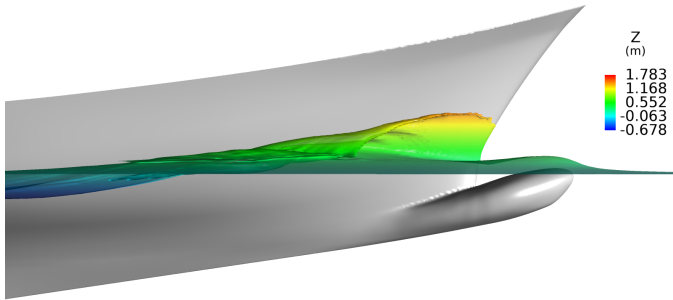
852

853 The sampling we performed with the Latin Hypercube is  
854 represented graphically with a response surface method, illus-  
855 trated in Fig.17. Figure 17(a) represents cutting planes of the  
856 design space, showing two main local minima. In Figure 17(b),  
857 we show iso-values of the total drag  $F_x$ . We can identify a re-  
858 gion where the objective function is predicted to be smaller than  
859 in the other parts of parameter domain.

860 Further investigations may lead to finding better drag reduc-  
861 tion results by using an adapted optimisation algorithm based  
862 on Kriging such as EGO to find minima using the model built  
863 from the response surface.

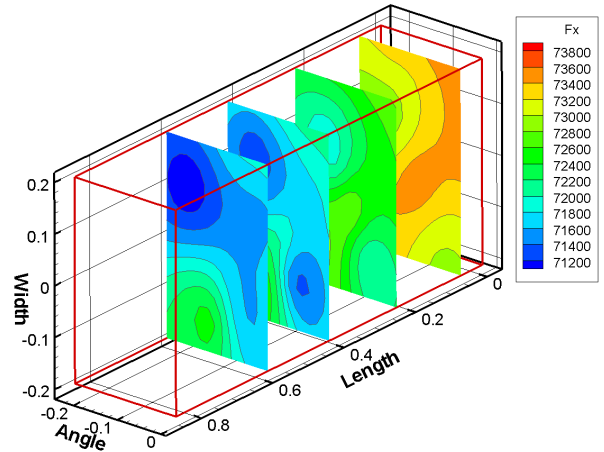


(a) Free surface elevation for the Initial bulb

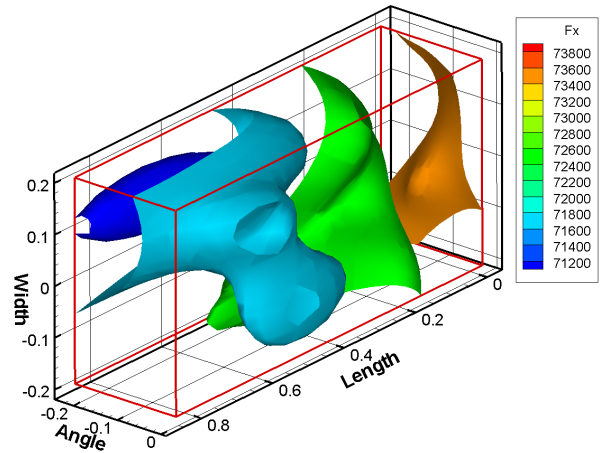


(b) Free surface elevation for the Best variation

Figure 16: Free surface elevation



(a) Cutting planes of the response surface



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

Figure 17: View of the response surface

## 864 7. Conclusion and future work

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

872  
 873 Our parametric modeller allows to explore the domain of  
 874 possible shapes in an efficient way, and allows to determine im-  
 875 provements of the design that are architecturally relevant.

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

879  
 880 Further work will focus on handling more complex geome-  
 881 tries with the skeleton representation. Section curves with mul-  
 882 tiple components, branching curves will be possible.

883  
 884 We will also develop the link with optimisation algorithm  
 885 solvers. A fully automatised optimisation loop will be devel-  
 886 oped. Sensitivity of the simulation results to parameters will be  
 887 taken into account in order to reduce the degrees of freedom as  
 888 much as possible.

889  
 890 **Acknowledgements:** The project was achieved with the finan-  
 891 cial support of ANRT (Association Nationale de la Recherche  
 892 et de la Technologie).

893

## 894 References

- 895 [1] Brizzolara, S., Vernengo, G., Pasquinucci, C.A., Harries, S.. Signif-  
896 icance of parametric hull form definition on hydrodynamic performance  
897 optimization. In: 6th International Conference on Computation Mehods  
898 in Marine Engineering (Marine 2015). 2015, p. 254–265.
- 899 [2] Peri, D., Rossetti, M., Campana, E.. Design optimization of ship hulls  
900 via cfd techniques. *Journal of Ship Research* 2001;10:140–149.
- 901 [3] Blanchard, L., Berrini, E., Duvigneau, R., Roux, Y., Mourrain, B.,  
902 Jean, E.. Bulbous bow shape optimization. In: 5th International Confer-  
903 ence on Computation Mehods in Marine Engineering (Marine 2013).  
904 2013, p. 412–423.
- 905 [4] Jones, D.R., Schonlau, M., Welch, W.J.. Efficient global opti-  
906 mization of expensive black-box functions. *J of Global Optimization*  
907 1998;13(4):455–492.
- 908 [5] Hansen, N.. *The CMA Evolution Strategy: A Comparing Review*; vol.  
909 192; chap. Towards a New Evolutionary Computation: Advances in the  
910 Estimation of Distribution Algorithms. Springer Berlin Heidelberg; 2006,  
911 p. 75–102.
- 912 [6] Duvigneau, R., Chandrashekar, P.. Kriging-based optimization applied  
913 to flow control. *International Journal for Numerical Methods in Fluids*  
914 2012;69(11):1701–1714.
- 915 [7] Pieggl, L., Tiller, W.. *The NURBS Book (2Nd Ed.)*. New York, NY,  
916 USA: Springer-Verlag New York, Inc.; 1997.
- 917 [8] Ju, T., Schaefer, S., Warren, J.D.. Mean value coordinates for closed  
918 triangular meshes. *ACM Trans Graph* 2005;24(3):561–566.
- 919 [9] Kang, J., Lee, B.. Geometric interpolation and extrapolation for rapid  
920 generation of hull forms. In: *COMPIT*. 2012, p. 202–212.
- 921 [10] Hock, J., Goh, C., Li, Y.. Hybrid evolutionary shape manipulation for  
922 efficient hull form design optimisation. In: *COMPIT*. 2016, p. 264–279.
- 923 [11] Peri, D., Diez, M.. Robust design optimization of a monohull for wave  
924 wash minimization. In: 5th International Conference on Computation  
925 Mehods in Marine Engineering (Marine 2013). 2013, p. 89–100.
- 926 [12] Duvigneau, R., Visonneau, M.. Shape optimization strategies for com-  
927 plex applications in computational fluid dynamics. In: 2nd International  
928 Conference on Computer Applications and Information Technology in the  
929 Maritime Industries. 2003, p. 1–8.
- 930 [13] Raven, H.C., Hoekstra, M.. A practical system for hydrodynamic opti-  
931 mization of ship hull forms. *VNSI Innovatiedag, Wageningen* 2003;:1–7.
- 932 [14] Yoshizawa, S., Belyaev, A.G., Seidel, H.P.. Skeleton-based variational  
933 mesh deformations. *Comput Graph Forum* 2007;26(3):255–264.
- 934 [15] Zhou, K., Huang, X., Xu, W., Guo, B., Shum, H.Y.. Direct manipula-  
935 tion of subdivision surfaces on GPUs. *ACM Trans Graph* 2007;26(3):Ar-  
936 ticle 91.
- 937 [16] Skouras, M., Thomaszewski, B., Bickel, B., Groos, M.. Computational  
938 design of rubber balloons. *Comput Graph Forum* 2012;31(2pt4):835–844.
- 939 [17] Schafer, H., Keinert, B., Niessner, M., Buchenau, C., Guthe, M.,  
940 Stamminger, M.. Real-time deformation of subdivision surfaces from  
941 object collisions. In: *Proceedings of the 6th High-Performance Graphics*  
942 *Conference*. EG; 2014, p. 1–8.
- 943 [18] Allaire, G., Jouve, F., Toader, A.M.. A level-set method for shape  
944 optimization. *C R Acad Sci Paris* 2002;334(12):1125–1130.
- 945 [19] Aguilar, J.C.. *Optimisation de formes hydrodynamiques. couche limite*  
946 *intrinseque tridimensionnelle*. Ph.D. thesis; Ecole des Mines de Paris;  
947 1996.
- 948 [20] Guido, Y.. *Controle et optimisation de forme dans les equations de*  
949 *navier-stokes*. Ph.D. thesis; Ecole des Mines de Paris; 1997.
- 950 [21] Ginnis, A., Duvigneau, R., Politis, C., Kostas, K., Bellibassakis, K.,  
951 Gerostathis, T., et al. A Multi-Objective Optimization Environment for  
952 Ship-Hull Design Based on a BEM-Isogeometric Solver. In: 5th Inter-  
953 national Conference on Computational Methods in Marine Engineering,  
954 Hamburg, Germany. 2013,.
- 955 [22] Papanikolaou, A., Harries, S., Wilken, M., Zaraphonitis, G.. Inte-  
956 grated ship design and multiobjective optimization approach to ship de-  
957 sign. In: *International Conference on Computer Applications in Ship-*  
958 *building*. 2011, p. 1–12.
- 959 [23] Jacquin, E., Derbanne, Q., Cordier, S., Alessandrini, B.. Hull form  
960 optimization using a free surface ranse solver. In: 25th Symposium on  
961 Naval Hydrodynamics. 2004, p. 1–14.
- 962 [24] Wang, W., Pottmann, H., Liu, Y.. Fitting b-spline curves to point clouds  
963 by curvature-based squared distance minimization. *ACM Trans Graph*  
964 2006;25(2):214–238.
- 965 [25] Mukesh, R., Lingadurai, K., Selvakumar, U.. Airfoil shape optimization  
966 using non traditional optimization technique and its validation. *Journal of*  
967 *King Saud University, Engineering Sciences* 2014;26(2):191–197.
- 968 [26] Kracht, A.. Design of bulbous bows. *SNAME Transactions*  
969 1978;86:197–217.
- 970 [27] Nocedal, J., Wright, S.J.. *Numerical optimization*. Springer Series in  
971 *Operations Research and Financial Engineering*; Berlin: Springer; 2006.
- 972 [28] Tang, C., Sun, X., Gomes, A., Wallner, J., Pottmann, H..  
973 Form-finding with polyhedral meshes made simple. *ACM Trans Graph*  
974 2014;33(4):70:1–70:9.
- 975 [29] Bo, P., Ling, R., Wang, W.. A revisit to fitting parametric surfaces to  
976 point clouds. *Computers & Graphics* 2012;36(5):534–540.
- 977 [30] Faltinsen, O.M.. *Hydrodynamics of High-Speed Marine Vehicles*. Cam-  
978 bridge University Press; 2006. ISBN 9780511546068.
- 979 [31] Iman, R.L., Helton, J.C., Campbell, J.E.Y.. An approach to sensitivity  
980 analysis of computer models, part 1. introduction, input variable selec-  
981 tion an preliminary variable assessment. *Journal on Quality Technology*  
982 1981;13(3):174–183.
- 983 [32] Jones, D.R.. A taxonomy of global optimization methods based on re-  
984 sponse surfaces. *Journal of Global Optimization* 2001;21:345–383.
- 985 [33] Wackers, J., Deng, G., Leroyer, A., Queutey, P., Visonneau, M..  
986 Adaptive grid refinement algorithm for hydrodynamic flows. *Computers*  
987 *& Fluids* 2012;55:85–100.
- 988 [34] Duvigneau, R., Visonneau, M., Deng, G.. On the role played by turbu-  
989 lence closures in hull shape imization at model and full scale. *J Marine*  
990 *Science and Technology* 2003;8(1):11–25.
- 991 [35] Queutey, P., Visonneau, M.. An interface capturing method for free-  
992 surface hydrodynamic flows. *Computers & Fluids* 2007;36(9):1481–  
993 1510.