ON THE OPTIMIZATION OF WINGLETS AT MODERATE REYNOLDS NUMBER USING IMMERSED BOUNDARY METHOD AND META-MODELING

João Marcelo Vedovoto
School of Mechanical Engineering - Federal University of Uberlândia
João Naves de Ávila Av. 2121, Uberlândia, MG, Brazil
jmvedovoto@mecanica.ufu.br

Felipe Antonio Chegury Viana
School of Mechanical Engineering - Federal University of Uberlândia
João Naves de Ávila Av. 2121, Uberlândia, MG, Brazil
fchegury@mecanica.ufu.br

José dos Reis Vieira de Moura Jr.
School of Mechanical Engineering - Federal University of Uberlândia
João Naves de Ávila Av. 2121, Uberlândia, MG, Brazil
mourajr@mourajr.com

Valder Steffen Jr.
School of Mechanical Engineering - Federal University of Uberlândia
João Naves de Ávila Av. 2121, Uberlândia, MG, Brazil
vsteffen@mecanica.ufu.br

Aristeu da Silveira Neto
School of Mechanical Engineering - Federal University of Uberlândia
João Naves de Ávila Av. 2121, Uberlândia, MG, Brazil
aristeus@mecanica.ufu.br

Abstract. Regarding the design of aircraft components, the state-of-art involves the use of extensive mathematical modeling and numerical simulation as well as optimization techniques. However, due the high computational costs in fluid dynamics, the search for accurate and viable solutions using numerical optimization constitutes a great challenge. This work has to main goals: (i) the use of the *Immersed Boundary Method* as a strategy of modeling and simulation of flows over complex three-dimensional geometries; and (ii) the optimization of a winglet using meta-modeling techniques. The *Immersed Boundary Method* has been developed by the Laboratory of Heat and Mass Transfer and Fluid Dynamics (LTCM) of the Federal University of Uberlandia. This technique makes use of two independent domains in the solution of the flows over complex geometries: a Eulerian domain, which is discretized using Finite Volume Method over a non-uniform mesh to integrate the Navier-Stokes equations and a second-order approximation for time and space derivatives. Another important remark is that the Lagrangian domain, which represents the immersed boundary, is represented by a superficial unstructured mesh, composed by triangles. In this scenario, meta-modeling is used as a way to overcome the high computational cost when dealing with fluid dynamics. The meta-models represent a low-cost approximation which can be easily dealt by numerical optimization codes. The in-house parallel code runs on a Beowulf-class cluster, a viable and reliable alternative to solve problems that demand very large computational resources. Finally, numerical results show the simulation of a three-dimensional flow over several airfoils NACA-0012 and design optimization of winglets aiming the maximum lift and minimum drag coefficients.

Keywords*: Immersed Boundary Method, Virtual Physical Model, Meta Modeling, Numerical Optimization.*

1. INTRODUCTION

In the Immersed Boundary (IB) methods, the presence of a solid or a gaseous interface inside a flow can be simulated by adding a source term into the Navier-Stokes equations, which acts as a fluid body force. Actually, the way this force is evaluated is the one of main point of this methodology. Furthermore, an important characteristic presented by IB methodologies is that the immersed obstacle can be represented by a Lagrangian mesh while the flow domain is discretized by an Eulerian grid such as the Cartesian or cylindrical ones. There is, also, an interpolation/distribution procedure that promotes the transferring of information from one domain to another. This domain independence allows promoting the displacement of the immersed body and/or a deformation relative to the flow grid.

The development of the IB method was credited to Charles Peskin and his collaborators, aiming to simulate the blood flow through cardiac valves. Accordingly to Peskin (1977), the source of the additional force term was due to the elastic boundary deformation rate, in which their constitutive points were tied by elastic membranes. More recently, Lima e Silva et al. (2003) proposed a model that evaluates the force field by the momentum equation based on a three points scheme, similar to what can be viewed in Mohd-Yusof (1997). However, that approach uses a more simplified interpolation scheme requiring less computational resources. The technique has been called Virtual Physical Model (VPM) since it employs a momentum equation and models the no-slip condition on the geometry wall in an indirect manner (Campregher, 2005).

Despite advances in computer capacity, the enormous computational cost of running complex engineering simulations makes it impractical to rely exclusively on simulation for the purpose of design optimization (Jin et al., 2001). This scenario is especially common in the computational fluid dynamics (CFD). To cut down the cost associated with this codes, surrogate models, also known as meta-models, are constructed from and then used in place of the actual simulation models. A variety of meta-modeling techniques exist: polynomial response surface (PRS) methodology (Box et al. 1978; Myers and Montgomery 1995) and artificial neural network (ANN) methods (Smith 1993; Cheng and Titterington 1994) are two well-known approaches for constructing simple and fast approximations of complex computer codes.

The current work has two proposals. The first one is to present an extension of the VPM to a three-dimensional domain and apply it to the flow around NACA-0012 airfoils. This step gives an analysis code for the problem. In sequence, this code is used to build a PRS model, which can be used in a design optimization context. Optimization is used to reduce drag while maximizes the lift related to the airfoil.

2. MATHEMATICAL AND NUMERICAL MODELING

The Immersed Boundary method uses two distinct domains to evaluate a flow over a complex geometry. An Eulerian domain is used to describe the behavior of the mean flow and covers the entire flow domain. For its turn, the Lagrangian domain is used to represent the interface fluid/fluid or fluid/solid.

This is one of the great advantages attributed to Immersed Boundary methods since it is possible to simulate flow around complex geometries using a more simplified Eulerian formulation for the fluid and a Lagrangian more versatile and simple grid for the interface fluid/solid. The coupling between Eulerian and the Lagrangian domains is done by Virtual Physical Model (Lima e Silva et al 2003).

In this work, Cartesian meshes were used to discretize the flow domain, configuring a simple and easy implementation, and at low computational cost. The following describes both domains in more details.

2.1. The Eulerian Domain

The domain was discretized by Finite Volume method over a structured non-uniform mesh. The flow is considered incompressible and isothermal. The integral form of the Navier-Stokes for such assumptions becomes:

$$\frac{\partial}{\partial t}\int_{\Omega}\rho\phi d\Omega+\int_{S}\rho\phi v\cdot n\,dS=\int_{S}\Gamma\nabla\phi\cdot n\,dS+\int_{\Omega}q_{\phi}d\Omega\,, \tag{1}$$

where ϕ is a property being transported, q_{ϕ} is the term of generation or destruction of ϕ, and Γ^{ϕ} is the diffusivity of ϕ.

The time derivative was approximated by a second-order three-time level (Ferziger & Peric, 2002), and the spatial derivatives by the Central-Difference Scheme.

The pressure-velocity coupling was done by the SIMPLEC method (Van Doormal e Raithby, 1984), with no relaxation in the velocity components equation. A co-located arrangement of variables was employed, and the Rhie-Chow (Rhie-Chow, 1983) interpolation method was used to avoid numerical oscillation due to pressure checkerboard fields.

The linear system originated from the velocity components discretization was solved by the SOR method. The SIP algorithm was used to solve the linear system generated by the discretization of the pressure correction equation.

The time and space integration of equation (1) over an elementary volume, after some mathematical arrangements leads to the following equation:

$$\begin{aligned}&\left(\frac{3\phi_p^{n}-4\phi_p^{n-1}+\phi_p^{n-2}}{2\Delta t}\right)\Delta x\Delta y\Delta z+\left(\rho_e u_e\phi_e-\rho_w u_w\phi_w\right)^n\Delta y\Delta z+\left(\rho_n u_n\phi_n-\rho_s u_s\phi_s\right)^n\Delta x\Delta z+\\&\left(\rho_t u_t\phi_t-\rho_b u_b\phi_b\right)^n\Delta x\Delta y=\left[\left(\Gamma^{\phi}\frac{\partial\phi}{\partial x}\right)_e-\left(\Gamma^{\phi}\frac{\partial\phi}{\partial x}\right)_w\right]^n\Delta y\Delta z+\\&\left[\left(\Gamma^{\phi}\frac{\partial\phi}{\partial y}\right)_n-\left(\Gamma^{\phi}\frac{\partial\phi}{\partial y}\right)_s\right]^n\Delta x\Delta z+\left[\left(\Gamma^{\phi}\frac{\partial\phi}{\partial z}\right)_t-\left(\Gamma^{\phi}\frac{\partial\phi}{\partial z}\right)_b\right]^n\Delta x\Delta y+q_{\phi}\Delta x\Delta y\Delta z\end{aligned}\,, \tag{2}$$

The first term of the left-hand side of the equation (2) represents the discretization of the transient term by the *three-time level scheme* (Muzaferija and Peric, 1997). This scheme is a second order accurate in time.

2.2. The Lagrangian domain

The Lagrangian approach for analyzing the movement of a particle constitutes of placing a system of coordinates at the particle and follow it individually. In other words, the system of coordinates moves through the flow following the particle. Thus, at each time step the particle keeps its own system of coordinates relatively to a global system of coordinates.

In the Virtual Physical model the geometry to be simulated is characterized by a Lagrangian set of points (see. Fig. (2)). This methodology permits to take advantage of the Lagrangian approximations like the ability to simulate moving bodies by just applying translation operations to the set of points.

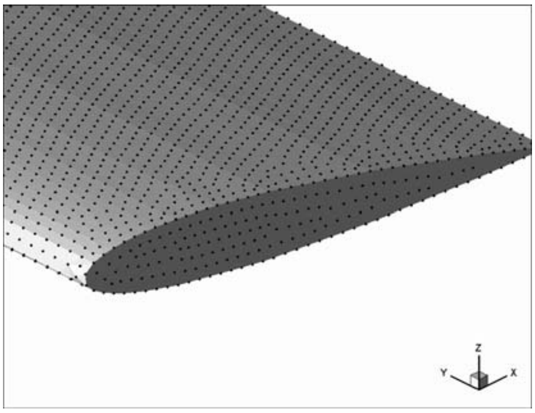


Figure 2: The surface of an airfoil NACA-0012 characterized by Lagrangian points.

The main characteristic of Immersed Boundary method is to simulate the presence of a fluid/solid or fluid/fluid interface inside a flow by adding a source term of force $\vec{f}$ to the Navier-Stokes equations. In Fig. (3) an arbitrary Lagrangian point k is shown with coordinates $\vec{x}_k$, as well as an elementary volume of fluid with coordinates $\vec{x}$. The evaluation of $\vec{f}$ differentiates the IB methods among them.

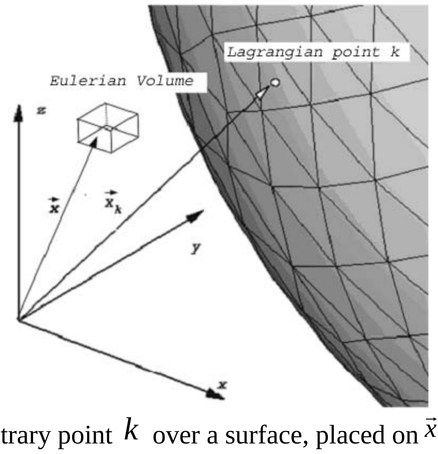


Figure 3: Schematic drawing of an arbitrary point k over a surface, placed on $\vec{x}_k$, and a element of fluid positioned in $\vec{x}$

In the Virtual Physical Model, the Lagrangian force is obtained from a balance of momentum over a particle k, placed at $\vec{x}_k$. This particle also has properties pressure p_k and velocity $\vec{V}_k$. Thus, the force can be evaluated as:

$$\vec{F}_k = \frac{\partial\left(\rho\vec{V}_k\right)}{\partial t} + \vec{\nabla}\cdot\left(\rho\vec{V}_k\,\vec{V}_k\right) - \mu\nabla^2\vec{V}_k + \vec{\nabla}p_k \tag{3}$$

The Eq. (3) can be interpreted as the necessary force so that a particle of fluid immediately adjacent to the Lagrangian point k reaches the velocity of this point, imposing a non-slip condition between the fluid and the immersed body.

Each term of the Eq. (3) has a particular meaning. The first term (the transient one) is responsible by the acceleration force ($\vec{F}_{acc}$). The other terms, of spatial derivatives, are known as the advective term, the diffusive term and the pressure gradient term, respectively. These terms are responsible for inertial forces ($\vec{F}_{inert}$), viscous forces ($\vec{F}_{visc}$), and pressure forces ($\vec{F}_{press}$). More details about this model and about each term evaluation can be found in Lima e Silva et al (Lima e Silva *et al*, 2003), and Campregher (2005).

The properties of the flow in the Eulerian mesh have to be interpolated to the Lagrangian mesh to calculate the Lagrangian forces. Once evaluated, the Lagrangian forces must be transferred back to the Eulerian domain. The connection between Lagrangian and Eulerian domains is promoted by the force distribution procedure.

2.2.1. The Virtual Physical Model

The discretization of Eq (3) is done by constructing a three-dimensional reference axis, with origin placed at the point k, as can be seen in the Fig (4). A Lagrangian polynomial is then used to obtain the space derivatives along each coordinate direction. Let *m* be a number of points employed to construct a polynomial interpolation of order *m-1*. Thus, the value of a property ϕ along *i* direction, at any point p, is given by:

$$\phi_i(p)=\sum_m \psi_m(p)\phi_m\ , \tag{4}$$

where,

$$\psi_m(p)_i=\prod_{n,n\neq m}\left[\frac{x_i(p)-x_i(n)}{x_i(p)-x_i(n)}\right]. \tag{5}$$

Substituting the *m* points, according to the stencil on Fig (4), the ϕ property value along the *x* axis (where k, k_1 and k_2 points lay) one can obtain as:

$$\phi_p=\left[\frac{(x_p-x_{k1})(x_p-x_{k2})}{(x_k-x_{k1})(x_k-x_{k2})}\right]\phi_k+\left[\frac{(x_p-x_k)(x_p-x_{k2})}{(x_{k1}-x_k)(x_{k1}-x_{k2})}\right]\phi_{k1}+\left[\frac{(x_p-x_k)(x_p-x_{k1})}{(x_{k2}-x_k)(x_{k2}-x_{k1})}\right]\phi_{k2}\,. \tag{6}$$

Deriving Eq (6) to *x* direction one has:

$$\frac{\partial\phi_p}{\partial x}=\left[\frac{(x_p-x_{k1})+(x_p-x_{k2})}{(x_k-x_{k1})(x_k-x_{k2})}\right]\phi_k+\left[\frac{(x_p-x_k)+(x_p-x_{k2})}{(x_{k1}-x_k)(x_{k1}-x_{k2})}\right]\phi_{k1}+\left[\frac{(x_p-x_k)+(x_p-x_{k1})}{(x_{k2}-x_k)(x_{k2}-x_{k1})}\right]\phi_{k2}\,, \tag{7}$$

and the second derivative results:

$$\frac{\partial^2\phi_p}{\partial x^2}=\left[\frac{2\phi_k}{(x_k-x_{k1})(x_k-x_{k2})}\right]+\left[\frac{2\phi_{k1}}{(x_{k1}-x_k)(x_{k1}-x_{k2})}\right]+\left[\frac{2\phi_{k2}}{(x_{k2}-x_k)(x_{k2}-x_{k1})}\right]. \tag{8}$$

From the equations above, it is possible to obtain every spatial derivative needed in Eq (3), just substituting the point *p* and the aimed variable ϕ.

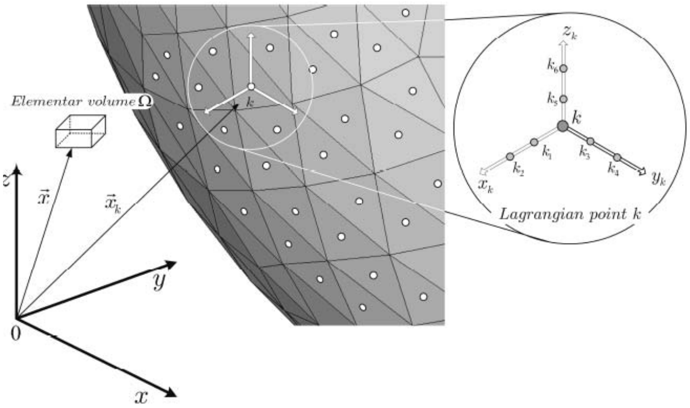


Figure 4: Position of the Lagrangian point $\vec{x}_k$

A detailed view of a triangular element can be obtained in the Fig (5). The element sides are formed by line segments S_1, S_2 and S_3, between the vertex points P_1, P_2 and P_3. Thus, one has $S_1 = \overline{P_2P_1}$, $S_2 = \overline{P_3P_2}$ and $S_3 = \overline{P_3P_1}$.

The ΔA_k is the triangular element surface area, which can be evaluated as:

$$\Delta A_k = \sqrt{S(S - S_1)(S - S_2)(S - S_3)} \tag{9}$$

where $S = (1/2)(S_1 + S_2 + S_3)$. The ΔS_k is the average length of the triangle sides. It worth noting that each of those geometric properties are associated to a Lagrangian point *k*.

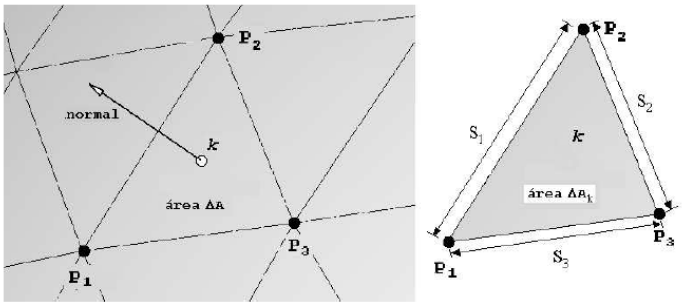


Figure 5: Detailed view of a triangular element

2.2.2. The distribution procedure

The Lagrangian force term $\vec{F}$, calculated at a Lagrangian point (denoted by k) is then distributed to Eulerian domain by means of the Dirac Delta Function. In an N-dimensional this function is defined as:

$$\vec{f}(\vec{x}) = \int_{R^n} \delta(\vec{x} - \vec{x}_k)\vec{F}(\vec{x}_k)d^n\vec{x}_k \quad . \tag{10}$$

Applying the Eq. (10) for a volume V of the Lagrangian domain,

$$\vec{f}(\vec{x}) = \int_{\Omega_n} \vec{F}(\vec{x}_k)\delta(\vec{x} - \vec{x}_k)d\vec{x}_k \quad . \tag{11}$$

The δ function has the following property:

$$\int_{R^n} \delta(\vec{x}-\vec{x}_k)d\vec{x} = \begin{cases} 1 & if\ \vec{x}_k \in V \\ 0 & if\ \vec{x}_k \notin V \end{cases}, \tag{12}$$

where $V \in \Omega$ .This function acts as the core of a transformed integral (centered in $\vec{x}_k$), which promotes the transposition between the Lagrangian and Eulerian domains (Griffith and Peskin, 2005).

In the Virtual Physical Model for three-dimensional domains, the Lagrangian force field ($F_{i,k}$) is distributed over the Eulerian mesh using Eq (13).

$$f_i = \sum F_{i,k} D_i \Delta A_k \Delta S_k \; . \tag{13}$$

The distribution function D_i is evaluated as:

$$D_i(x_k) = \prod_i \left\{ \frac{\varphi[(x_k - x_i)/\Delta x_i]}{\Delta x_i} \right\} , \tag{14}$$

where the φ function is defined as:

$$\varphi(r) = \begin{cases} \tilde{\varphi}(r) & if\ \|r\| < 1 \\ \frac{1}{2} - \tilde{\varphi}(2-r) & if\ 1 < \|r\| < 2 \\ 0 & if\ \|r\| > 2 \end{cases} , \tag{15}$$

$$\tilde{\varphi}(r) = \frac{3 - 2\|r\| + \sqrt{1 - 4\|r\| + 4\|r\|^2}}{8} . \tag{16}$$

The Distribution function is divided by a volume unit, that cancel out by multiplying for a characteristic area (ΔA_k) and for a characteristic length (ΔS_k). Thus, it remains the force density that is integrated over the volume Ω .

The interface solid/fluid is managed by an indicator function I_i , built from:

$$\nabla^2 I_i = \nabla G_i \, , \tag{17}$$

where the *G* function is defined as:

$$G_i = \sum D_i \vec{n}_k \Delta A_k \; , \tag{18}$$

and the $\vec{n}_k$ is the normal vector on the Lagrangian point k.

After the discretization of the Eq. (17), the algebraic equation system is evaluated by the MSI algorithm (Schneider and Zedan, 1981), a variation of the SIP procedure. By analyzing the Eq. (18), one can see that if the geometry is inserted into a non-uniform grid region, the interfacial region may become deformed, i.e., the geometry shell shape would be misrepresented.

Briefly describing, the force field evaluation procedure in the Virtual Physical Model can be stated as:

(1) With the flow field solved, the velocity components and the pressure are transferred, using the interpolation function given by Eq. (18), to the nearest Lagrangian points *(k, k1...k6*) depicted in Fig. (4);

(2) Once having $u_{i,k}$ and $p_{i,k}$, evaluates $F_{i,k}$ by Eq. (3);

(3) Calculates the force field components, due to each Lagrangian point k, via Eq. (13);

(4) Advances in time;

(5) The force field is inserted into the source-term of Eq. (1);

(6) A new flow field is obtained and the procedure re-starts.

3. META-MODELING TECHNIQUE AS A DESIGN TOOL

As a way to reduce the computational cost, the use of meta-models to represent the functions involved in an optimization problem has become an established approach. The statistical procedure used to generate them can be summarized as follows (Biles and Swain, 1980; Box et al. 1978; and Myers and Montgomery, 1995):

1) Choice of a model and experimental design: the nature of the surrogate itself is determined, and then a design space, including a range of design possibilities, is sampled in order to reveal its contents and tendencies.
2) Model fitting: the model whose shape is defined is fitted to the collected data.
3) Verification of model accuracy: the precedent steps are sufficient to build a first tentative model, whose overall quality and usefulness has to be evaluated by adequate sets of metrics. Each combination of design space sampling, model choice and fitting procedure leads to the use of specific verification procedures.

In the context of PRS, a polynomial function, $\hat{y}(\mathbf{x},\beta)$, is used to approximate the actual function, $y(\mathbf{x})$, as can be expressed by:

$$\hat{y}(\mathbf{x},\beta) = \beta_0 + \sum\nolimits_{i=1}^{k} \beta_i x_i + \sum_i \sum_j \beta_{ij} x_i x_j + \sum\nolimits_{i=1}^{k} \beta_{ii} x_i^2 + \ldots + \varepsilon \tag{19}$$

where k is the number of design variables considered in the problem.

The order of the polynomial used in the approximation can vary according to the problem, In general, first-order and second-order polynomials are the most common.

To provide a more complete picture of meta-model accuracy, three different metrics are used: the root mean square error, E_{RMS}, the R^2, and the generalized mean square cross-validation error, or $PRESS$ in the PRS terminology (Box et al. 1978; Myers and Montgomery, 1995). The equations for these three metrics are given by Eq. (20) to (21), respectively:

$$E_{RMS} = \sqrt{\frac{1}{n} \sum\nolimits_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{20}$$

where n is the number of points provided by the design of experiments, $\hat{y}_i$ is the corresponding predicted value for the observed value y_i. The smaller the E_{RMS}, the more accurate the meta-model.

$$R^2 = 1 - \frac{\sum\nolimits_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum\nolimits_{i=1}^{n} (y_i - \overline{y})^2} \tag{21}$$

where $\overline{y}$ is the mean of the observed values. The larger the value of R^2 (i.e., the closer to 1), the more accurate the meta-model.

$$PRESS = \frac{1}{n} \sum\nolimits_{i=1}^{n} \left(y_i - \hat{y}_i^{(-i)} \right)^2 \tag{22}$$

where $\hat{y}_i^{(-i)}$ represents the prediction at $\mathbf{x}^{(i)}$ using the PRS constructed using all sample points except $\left(\mathbf{x}^{(i)}, y_i\right)$. The smaller the $PRESS$ value, the more accurate the meta-model.

While E_{RMS} represents the departure of the meta-model from the real simulation model, the variance captures how irregular the problem is. On the other hand, $PRESS$ gives an idea about how independent is the model from the chosen data. Indeed, it reflects the influence in case of remove one of the points. It is important to point out that only the R^2 measurement is non-dimensional. Thus, when using E_{RMS} and $PRESS$, it is necessary to consider the magnitude of the quantity of interest.

4. RESULTS AND DISCUSSION

4.1. Results of the simulations – Numerical experiments

The airfoils used in the simulations are shown in Fig (7). The numerical domain has dimensions *length (X)* = 0.88*m*, *width (Y)* = 0.342 *m* and *height (Z)* = 0.384 *m.* Such domain was discretized by, 118x111x92 grids in *X*, *Y* and *Z* axis, respectively.

The NACA 0012 airfoils, with winglets are represented by a triangular element mesh. Again, the Immersed Boundary methodology requires only the discretization of the surface that represents the fluid/solid interface, i.e. the body shell. The airfoil is centered at (x_c, y_c, z_c) = (*0.28, 0.34, 0.165) m*, having a chord *c=0.04m* and a width of 4*c.* For every simulation the attack angle α = *0º* and the Reynolds number, based on chord, is Re = 10000.

The boundary condition for velocity components on the Eulerian domain side walls were set as non-slip conditions, the inflow at *x=0 m* had a flat profile with values of $u=U_I$ (inlet velocity), $v=w=0 m/s$. The outflow was set of Neumann conditions.

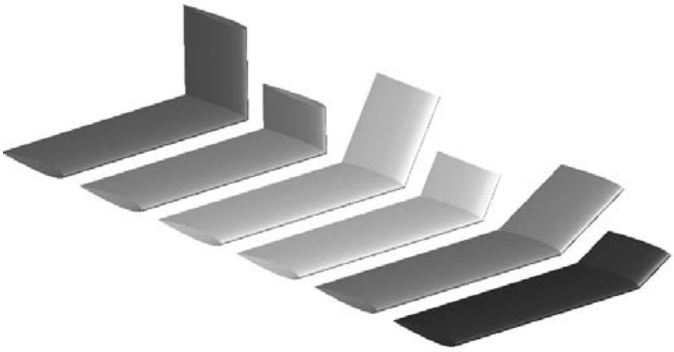
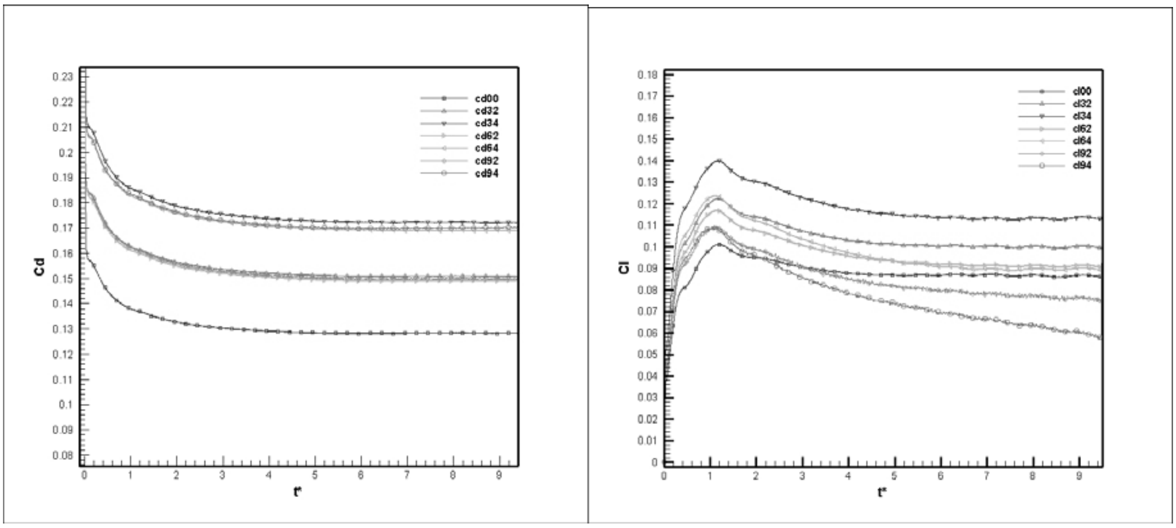


Figure 6: Airfoils used in the numerical experiments discretized using a triangular elements mesh.

The value of the L2 norm was about *10*$^{-3,}$ for all simulations, which is acceptable, once that the code has a second order in time-space accuracy.

Figures (7) and (8) show the evolution of the curves of drag coefficient and lift coefficient for all, respectively, simulations.



(a) (b)

Figure 7: Plot of drag coefficient X adimensional time (a). Plot of lift coefficient X adimensional time (b)

The adimensional time is given by eq (28):

$$t^* = {(t * U_\infty)} \Big/ c \qquad (23)$$

where c means the chord of airfoil, t is the dimensional time, and U_∞ is the free-stream velocity.

In Figures (8) and (9), one can note the notation CD32, or CD94, or even CL62 etc... In the first case, and similarly for the other examples, means the Drag curve for a 30° angle winglet with a length of 0.02m.

One can also note, in Fig (8), the significance of length in the drag coefficient, this fact was predicted in the analysis of variance.

4.2. Numerical Optimization

In this work, the PRS approach is used to build meta-models for the drag and lift coefficients, obtained from the expensive CFD codes previously described. These models will take as design variables the angle and the length of the winglet. At the end of the optimization, the drag coefficient must be reduced and the lift coefficient must be increased.

Following the steps proposed in Section 4, the choice of model and the design of experiments start by taking into account the extremely high cost in a single evaluation of the CFD codes (almost seven days, even using parallel implementation). Thus, a first order model is used for both lift and drag coefficients. Table 1 and Table 2 show the bounds and the design points used for both the angle and the length of the winglet.

Table 1. Design space specifications.

Design Space	**Angle (x_1)**	**Length (x_2)**
Lower bound	30	0.02
Upper bound	90	0.04

Table 2 Experimental design.

Angle (x_1)	**Length (x_2)**	**Drag (y_1)**	**Lift (y_2)**
30	0.02	0.16	0.0116
30	0.04	0.172	0.0154
60	0.02	0.149	0.0105
60	0.04	0.169	0.0121
90	0.02	0.15	0.0087
90	0.04	0.17	0.0078

Using the previously shown setup, the following meta-models were obtained:

$$y_1 = 0.1560 - 0.0060x_1 + 0.0173x_2 \tag{24}$$

$$y_2 = 0.0129 - 0.0053x_1 + 0.0015x_2 \tag{25}$$

Table 3 shows the error measurements for both quantities. As can be noticed, the drag coefficient has better R^2 while both present good values of E_{RMS} and $PRESS$ errors when considering the magnitude of the quantities.

Table 3. Error measurements.

	Drag (y_1)	**Lift (y_2)**
R^2	0.919395	0.842792
E_{RMS}	0.003771	0.001387
$PRESS$	0.005528	0.002333

After building the meta-models for the responses, the numerical optimization continues by defining the design problem according to the following formulation:

$$optimize \;\; f(\mathbf{x}) \tag{26}$$

subject to:

$$\begin{aligned} &\mathbf{x}_i^L \leq \mathbf{x}_i \leq \mathbf{x}_i^U, && \text{i} = 1, 2, 3, ..., \text{N} \\ &g_j(\mathbf{x}) \leq 0, && \text{j} = 1, 2, 3, ..., \text{M} \\ &h_j(\mathbf{x}) = 0, && \text{k} = \text{M} + 1, \text{M} + 2, \text{M} + 3, ..., \text{L} \end{aligned} \tag{27}$$

where:

- $optimize$ can be either finding the minimum or the maximum of $f(\mathbf{x})$,
- $f(\mathbf{x})$ is the objective function,
- $g_j(\mathbf{x})$ and $h_j(\mathbf{x})$ are inequality and equality constraints, respectively, and
- $\mathbf{x}_i^L \leq \mathbf{x}_i \leq \mathbf{x}_i^U$ are the side constraints of the design space.

In the present work, three different formulations are solved using MATLAB functions fmincon and linprog. In the first formulation, both drag and lift are considered as objective functions. In this case the optimization consists in minimizing the drag and minimizing the lift coefficients. These responses are combined into a functional whose minimization, via fmincon, implies that all responses tend to a target value and depart from a non-desirable one. The target and avoidable values are not necessarily design goals, but play an important role at the optimization problem by defining the tendency of the sought optima with respect to the baseline design. This scheme is known as Compromise Programming, better described by Vanderplaats (2005).

The second and the third formulations make uses of the constrained optimization. In one turn, the drag is taken as an objective function to be maximized and the lift as a constraint, on the other turn, the lift is considered as an objective function to be minimized and the drag is a constraint. In both cases, the numerical implementation uses linprog, a projection method as described in Gill et al. (1981). Basically, linprog implements a variation of the well-known simplex method for linear programming.

After run each o the previously presented formulations, a set of optimal results is obtained. Due the linear nature of the meta-models, the results do not vary if one runs a certain formulation starting from a different initial design. However, to finish the design optimization, it is necessary to go back to the CFD model and evaluate each of the obtained design. Just after that, it is possible to judge which formulation gives the best answer. Table 4 shows the results obtained from the optimization using the meta-models and the validation with the CFD codes.

Table 4. Optimization Results

Approach	Angle	Length	PRS values		CFD values	
			Drag	Lift	Drag	Lift
Multi Objective	51.12787	0.02	0.153887	0.011043	0.158	0.008
Minimize Drag	51.50943	0.02	0.153849	0.011010	0.1665	0.017
Maximize Lift	52.00000	0.02	0.153800	0.010967	0.168	0.016

It is now clear that is necessary to feedback the actual CFD codes with the results obtained from the optimization runs. This is important not only for comparison, but also for decision making. As can be viewed in Table 4, the PRS values for all approaches are very close to each other. Then, when the CFD values are taken into account, it is possible to see that for drag they can be closed, but definitely for lift they are distinct. Finally, the design obtained by the minimization of drag seems to be the best among them.

5. CONCLUSION

The main objective of this work was to show that it is possible to make an optimization of geometry, aiming aerodynamic efficiency, without the use of embarked methods of optimization in the CFD Solver. This objective was achieved successfully. The Immersed Boundary Method have shown great capability in dealing with complexes geometries and/or moving bodies, once the Eulerian mesh is a Cartesian mesh, very simple to be created, and there are no further difficulties in generating the Lagrangian mesh, which is a advantage if compared with other methodologies of studying Fluid-Structure Interaction.

The use of statistical tool for creating a meta-model was fundamental, and this proceeding shown great results It was possible also, to notice that, the meta-model was adjusted well to the problem, with correlations of up to 99%, guaranteeing that the reliability of the optimization. Also the meta-modeling have shown an excellent tool for leading with multidisciplinary problems, like Fluid Structure Interaction.

As final remarks, the whole methodology presented in this paper, show the beginning of a methodology for designing, operating and optimizing complexes systems.

6. REFERENCES

Biles, W. E. and Swain, J. J. Optimization and Industrial Experimentation. John Wiley & Sons, New York, USA, 1980.

Box, G.E.P.; Hunter, W.G.; Hunter, J.S. 1978: Statistics for experimenters. New York: John Wiley & Sons.

Campregher, Rubens, Modelagem Matemática tridimensional para problemas de interação fluido-estrutura, doctoral thesis, Uberlândia 2005

Cheng, B.; Titterington, D.M. 1994: Neural networks: a review from a statistical perspective. Statistical Science, Vol. 9, pp. 2–54.

Ferziger J and Peric M., Computational methods for fluid dynamics 3th ed., 2002, Springer-Verlag, New-York, USA.

Gill, P. E. and W. Murray, and M.H. Wright, Practical Optimization, Academic Press, London, UK, 1981.

Griffith, B. e Peskin, C., 2005, "On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems.", Journal of Computational Physics 208, 75-105.

Lima e Silva A.L.F., Silveira-Neto, A., and Damasceno, J.J.R; Numerical simulation of two dimensional flows over a circular cylinder using the immersed boundary method, J. Comp. Phys. 189, 351 (2003).

Jin, R., Chen, W., and Simpson, T. W., 2001, "Comparative studies of metamodelling techniques under multiple modeling criteria," Structural and Multidisciplinary Optimization, 23, pp. 1-13.

Myers, R.H.; Montgomery, D.C. 1995: Response surface methodology: process and product optimization using designed experiments. New York: Wiley & Sons.

Mohd-Yusof, J. Combined immersed boudaries/B-splines methods for simulations in complex geometries, CTR Annual Research Briefs, NASA Ames/Stanford University, 1997.

Muzaferija, S. e Peric, M., 1997, "Computational of free-surface flows using the Finite-Volume method and moving grids", Numerical Heat Transfer, Part B 32, 369-384.

Peskin C.S., Numerical analysis of the blood flow in the heart, J. Comp. Phys. 25, 220 (1977)

Rhie C. and Chow W., Numerical study of turbulent flow past an airfoil with trailing edge separation, AIAA Journal 21, 1525 (1983)

Schneider G.E. and Zedan M., A modified strongly implicit procedure for the numerical solution of field problems, Numer. Heat Transfer 4, 1 (1981).

Smith, M. 1993: Neural networks for statistical modeling. New York: Von Nostrand Reinhold.

Van Doormal, J. e Raithby, G., "Enhancements of the simple method for predicting incompressible fluid flows", Numerical Heat Transfer 7, 147-163, 1984

Vanderplaats, G. N. Numerical Optimization Techniques for Engineering Design, Vanderplaats Research and Development, Inc., 4th edition, 2005.

7. RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.