Low-speed airfoil optimization using constrained differential evolution

Mihai-Vladut HOTHAZIE*,[1], Matei MIRICA[2]

*Corresponding author
[1]INCAS – National Institute for Aerospace Research “Elie Carafoli”,
B-dul Iuliu Maniu 220, Bucharest 061126, Romania,
hothazie.mihai@incas.ro
[2]International Computer High School of Bucharest,
Strada Balta Albina nr. 9,
miricamatei97@gmail.com

DOI: 10.13111/2066-8201.2020.12.4.8

Received: 11 September 2020/ Accepted: 12 October 2020/ Published: December 2020

International Conference of Aerospace Sciences “AEROSPATIAL 2020”, Virtual Conference
15-16 October 2020, Bucharest, Romania,
Section 1 - Aerodynamics

Abstract: *Nowadays, algorithms designed to optimize the shape of an airfoil are being developed by many researchers. In this paper, to achieve an optimum shape configuration, a methodology based on an evolutionary algorithm is proposed. The main objective is to find the optimum shape of a known airfoil that gives the best aerodynamic performance for a fixed lift coefficient. For the airfoil parametrization, the class-shape method is used to develop a well-behaved geometry. The paper underlines the implementation of a constrained differential evolutionary algorithm using the free penalty scheme by varying the coefficients of the shape parametrization function. The aim is to obtain a better aerodynamic performance for a predetermined lift coefficient by imposing a fixed maximum airfoil thickness interval. The method is a general optimization procedure and can be implemented in a wide range of engineering design problems.*

Key Words: *shape, airfoil, optimization, evolutionary algorithm, differential evolution, constraints*

1. INTRODUCTION

The substantial progress of the computational capabilities of modern computers allows us to evaluate complex engineering problems without the need of simplifications.

As a result, complex aerodynamic equations can now be solved with the help of modern digital computers.

In the quest for the best aerodynamic geometry of an airfoil, numerous optimization techniques have been developed.

However, many of these methods optimize the airfoil for certain situations and decrease the aerodynamic performances in other conditions.

To overcome these limitations, the criteria for the optimization process must be wisely chosen. For this, our paper proposes an evolutionary algorithm based on a free penalty scheme coupled with an efficient procedure to obtain the aerodynamic performances of an airfoil. It is noteworthy that the proposed objective function takes into account the lift-drag ratio at multiple angles of attack with corresponding weights.

The search for the optimal and robust airfoil geometry is initiated from a known airfoil configuration, although, a random geometry can be used.

In the end, a final population of numerous airfoils is obtained, and selected cases are chosen based on the robustness criterion. To illustrate the usefulness of this technique, a methodology is presented, and a series of results are analyzed followed by the corresponding conclusions.

2. METHODOLOGY

A computer program written in Fortran 90 was developed based on a differential evolution method. For this, a systematic aerodynamic analysis is applied, starting with a class-shape parametrization of the airfoil geometry followed by an approximation of the airfoil outline with a given number of panels. Thus, the inviscid solution is calculated using a linear vortex panel method.

The viscous part of the solution is based on a boundary layer analysis using Cebeci-Smith's method.

Lastly, an objective function is formulated based on the airfoil performance coefficients and the optimum solution is obtained using the evolutionary part of the algorithm.

2.1 Class-shape parametrization

The Class-Shape method was recently developed by Kulfan in 2006 [1]. The general mathematical formulation necessary to describe a two-dimensional airfoil geometry is defined as the product of the class function $C(x)$ and a shape function $S(x)$:

$$z(x) = C_{N_1}^{N_2}(x) \cdot S(x) \tag{1}$$

where the class function has the following form:

$$C_{N2}^{N1}(x) \equiv (x)^{N_1}(1-x)^{N_2}, \quad 0 \le x \le 1 \tag{2}$$

Both exponents, N_1 and N_2, vary from 0 to 1 and generate a random airfoil shape. To define a NACA type airfoil, a rounded leading edge is obtained when N_1 equals to 0.5, while a sharp trailing requires N_2 to be equal to 1.

Variation of the exponents yields to other airfoils geometries and a particular selection makes the basic shape in the airfoil class. To define the trailing edge thickness, an additional term is added to *(1)* so that:

$$z(x) = C_{1.0}^{0.5}(x) \cdot S(x) + (x) \cdot \Delta z_{te}. \tag{3}$$

For the shape function, a Bernstein polynomial expression is chosen:

$$S(x) = B_i^n(x) = C_n^i \cdot x^i (1-x)^{n-i} \tag{4}$$

The class function defines general classes of geometries, whereas the shape function determines individual shapes within the geometry class providing easy control of the airfoil critical design parameters.

To completely define the upper and the lower surface of the class-shape function transformation, equation (3) becomes:

$$z = \sqrt{x} \cdot (1-x) \cdot \sum_{i=0}^{n} \left[W_i \cdot C_n^i \cdot x^i \cdot (1-x)^{n-i} \right] + x \cdot \Delta z_{te} \tag{5}$$

In our problem, the coefficients of the shape function (W_i) are the optimization variables, and this method, in particular, proves to be highly efficient in the differential evolution algorithm.

They range between -1 to 1, so the search region of the algorithm is narrowed, reducing the computational time.

For the x component, a cosine spacing is used with uniform increments of β to generate a higher density of points near the leading and the trailing edge of the airfoil:

$$x = \frac{1 - cos\beta}{2}, \quad 0 \le \beta \le \pi \tag{6}$$

2.2 Vortex panel-method

A linear strength vortex panel method was used to predict the aerodynamic properties of the airfoil [2].

The solutions will initially be inviscid flow predictions from which we will obtain the lift and pitching moment properties.

After that, we will consider viscous forces in the boundary layer and we will predict the viscous component of the solution and the skin drag properties. This particular panel method has the property that the variation of the density circulation on each panel is linear and is continuous across two adjacent panels.

The midpoint of the panel referred to as the control point, is where the condition of no normal velocity is applied.

The velocity potential at a given control point x_i, y_i is composed of the effect of all vortex panels along the airfoil and the free stream velocity:

$$\emptyset(x_i, y_i) = V_\infty (x_i cos\alpha + y_i sin\alpha) - \sum_{j=1}^{m} \int \frac{\gamma(s_j)}{2\pi} tan^{-1}\left(\frac{y_i - y_j}{x_i - x_j}\right) ds_j \tag{7}$$

where α represents the angle of attack.

The vortex strength is given by the linear variation along with each panel:

$$\gamma(s_j) = \gamma_j + (\gamma_{j+1} - \gamma_j)\frac{s_j}{S_j} \tag{8}$$

The no penetration boundary condition at each control point is given by:

$$\frac{\partial}{\partial n_i} \emptyset(x_i, y_i) = 0, \quad i = 1,2,\ldots,m \tag{9}$$

Doing all the necessary differentiation and integration, Kuethe and Chow (1986) give the following results:

$$\sum_{j=1}^{m} \left(C_{n1_{ij}} \gamma'_j + C_{n2_{ij}} \gamma'_{j+1} \right) = sin(\theta_i - \alpha) \quad i = 1,2,\ldots,m \tag{10}$$

in which $\gamma' = \gamma/2\pi V_\infty$ represents the dimensionless circulation and θ_i the angle between the i -th panel and x axis.

The coefficients above are given by:

$$\begin{gathered} C_{n1_{ij}} = 0.5DF + CG - C_{n2_{ij}} \\ C_{2n_{ij}} = D + 0.5QF/S_j - (AC + DE)G/S_j \end{gathered} \tag{11}$$

in which

$$\begin{gathered} A = -(x_i - X_j)cos\theta_i - (y_i - Y_j)sin\theta_j \\ B = (x_i - X_j)^2 + (y_i - Y_j)^2 \\ C = sin(\theta_i - \theta_j) \\ D = cos(\theta_i - \theta_j) \\ E = (x_i - X_j)sin\theta_j - (y_i - Y_j)cos\theta_j \\ F = ln\left(1 + \frac{S_j^2 + 2AS_j}{B}\right) \\ G = tan^{-1}\left(\frac{ES_j}{B + AS_j}\right) \\ P = (x_i - X_j)sin(\theta_i - 2\theta_j) + (y_i - Y_j)cos(\theta_i - 2\theta_j) \\ Q = (x_i - X_j)cos(\theta_i - 2\theta_j) + (y_i - Y_j)sin(\theta_i - 2\theta_j) \end{gathered} \tag{12}$$

At the trailing edge, Kutta condition is applied:

$$\gamma_1' + \gamma_{m+1}' = 0 \tag{13}$$

Now, the resulting system consists of $m + 1$ equation and is rewritten as follows:

$$\sum_{j=1}^{m+1} A_{n_{ij}} \gamma_j' = (RHS)_i, \quad i = 1,2,\ldots,m+1 \tag{14}$$

where, for $i < m + 1$:

$$\begin{gathered} A_{n_{i1}} = C_{n1_{i1}} \\ A_{n_{ij}} = C_{n1_{ij}} + C_{n2_{ij-1}}, \quad j = 2,3,\ldots,m \\ A_{ni_{m+1}} = C_{n2_{im}} \\ (RHS)_i = sin(\theta_i - \alpha) \end{gathered} \tag{15}$$

and for $i = m + 1$

$$\begin{gathered} A_{n_{i1}} = A_{n_{im+1}} = 1 \\ A_{n_{ij}} = 0 \quad j = 2,3,\ldots,m \\ (RHS)_i = 0 \end{gathered} \tag{16}$$

The resulting system is solved based on a LU decomposition. Having the solutions, the velocity at every control point is computed by the following formula:

$$
\begin{gathered}
V_i = cos(\theta_i - \alpha) + \sum_{j=1}^{m} \left(C_{t1_{ij}} \gamma'_j + C_{t2_{ij}} \gamma'_{j+1} \right) \\
A_{t_{i1}} = C_{t1_{i1}} \\
A_{t_{ij}} = C_{t1_{ij}} + C_{t2_{ij-1}} \quad j = 2,3,\ldots,m \\
A_{ti_{m+1}} = C_{t2_{im}}
\end{gathered}
\tag{17}
$$

Knowing the velocities of each control points, the pressure coefficient can be determined as follows:

$$
C_{pi} = 1 - V_i^2 \tag{18}
$$

2.3 Boundary layer analysis

The boundary layer represents the region close to the wall, where the viscous forces cannot be neglected.

In this case, the viscous part solution of the flow is obtained using the boundary layer method described by Cebeci-Smith [3].

To determine the parameters of the steady two-dimensional incompressible boundary layer, we start with the governing equations:

$$
\begin{gathered}
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \\
u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = U_e \frac{dU_e}{dx} + v \frac{\partial^2 u}{\partial y^2} - \frac{\partial}{\partial y} \left(\overline{u'v'} \right) \\
0 = \frac{\partial p}{\partial y}
\end{gathered}
\tag{19}
$$

For the laminar part of the flow, the Reynolds shear stress term $-\rho\overline{u'v'}$ equals to zero. Using the above system of equations, we introduce a Falkner-Skan transformation in which the similarity variable η is given by:

$$
\eta = y \sqrt{\frac{U_e}{\nu x}} \tag{20}
$$

Also, a dimensional stream function is introduced:

$$
f(x, \eta) = \frac{\psi(x, y)}{\sqrt{u_e \nu x}} \tag{21}
$$

With this transformation, the system of governing equations can be expressed as a third-order differential equation known as the Falkner-Skan equation.

$$
(bf'')' + \frac{m+1}{2} f f'' + m[1 - (f')^2] = x \left(f' \frac{\partial f'}{\partial x} - f'' \frac{\partial f}{\partial x} \right) \tag{22}
$$

The boundary conditions for the continuity and momentum equations are rewritten as follows:

$$\begin{aligned} \eta = 0 \;\Rightarrow\quad & \begin{cases} f = f_w = -\dfrac{1}{\sqrt{u_e \nu x}} \displaystyle\int_0^x v_w dx, \\ f' = 0 \end{cases} \\ \eta = \eta_e \Rightarrow\quad & \qquad\qquad f' = 1 \end{aligned} \tag{23}$$

The numerical solution is obtained by discretizing the system using Keller's Box method [4].

Firstly, we express the equations above as a first-order system by introducing new variables that represent the derivatives of f. Then, an approximation is done using a rectangular box with centered derivates and averages at the midpoints. To solve the nonlinear system, we use Newton's method. This yields to a linear system with a tridiagonal structure of the following form:

$$A\vec{\vec{\delta}} = \vec{\vec{r}} \tag{24}$$

The solution is obtained using the block elimination method described by Cebeci and Bradshaw. Based on the empirical criteria reported by Michael, the transition is assumed to occur when the momentum thickness exceeds a certain value determined by the following equation:

$$Re_{\theta_{tr}} = 2.9Re_{x_{tr}}^{0.4} \tag{25}$$

If the Reynolds number is sufficiently large, turbulence can occur. For the solution of the turbulent flow, a differential method developed by Cebeci and Smith is used [3]. It's based on the method described for the laminar flow, but with the addition of the Reynolds shear stress modeled using the algebraic eddy viscosity formulation.

Using the Falkner-Skan transformation, the eddy viscosity equations for the inner and outer regions are:

$$\begin{gathered} (\nu_t^+)_i = 0.16R_x^{1/2}\left[1 - exp\left(-\frac{y}{A}\right)\right]^2 \eta^2 v\gamma_{tr}\gamma \\ (\nu_t^+)_0 = 0.168R_x^{1/2}[\eta_e - f(\eta_e)]\gamma_{tr}\gamma \\ \frac{y}{A} = \frac{N}{26}R_x^{1/4}v_w^{1/2}\eta, \quad p^+ = mR_x^{1/4}(v_w)^{-3/2} \\ R_x = \frac{u_e x}{\nu}, \quad \nu_t^+ = \frac{\nu_t}{\nu} \end{gathered} \tag{26}$$

Obtaining the solutions, for both the inviscid and viscous part of the flow, drag, lift, and momentum can be calculated as follows:

$$\begin{aligned} L &= -\int p \cdot sin\,(\theta - \alpha)\, dA + \int \tau_w \cdot \cos(\theta - \alpha)\; dA \\ D &= \int p \cdot cos\,(\theta - \alpha)\, dA + \int \tau_w \cdot \sin(\theta - \alpha)\; dA \\ M_{c/4} &= -\int p \cdot sin\,(\theta - \alpha)\,\big((x - 0.25)dx + dy\big) \end{aligned} \tag{27}$$

where τ_w is the wall shear stress, α represents the angle of attack and θ the angle between the panel and x-axis.

2.4 Differential evolution

Differential evolution (DE) as introduced by Storn & Price [5] operates similarly as a standard evolutionary algorithm (EA).

Even though in EA a probability parameter is used, in DE the offspring population is generated by varying the current generation by a scaled difference between two random members. In the following, a general constrained optimization problem is presented:

Minimize the objective function $f(x)$ subjected to:

$$\begin{gathered} g_k(x) \leq 0, \quad k = 1, \dots K \\ h_l(x) = 0, \quad l = 1, \dots L, \quad L < n \\ l^k \leq x^j \leq u^j, \quad j = 1, \dots n \end{gathered} \tag{28}$$

where $g_k(x)$ represents the inequality constraints and $h_l(x)$, equality constraints.

In our code, a free parameter scheme was used in which a modified fitness function is introduced to ensure the superiority of any feasible point compared to all unfeasible ones [6], [7]. The modified fitness function is written as follows:

$$\hat{f}(x_{i,g}) = f(x_{i\prime g}) + G(x_{i\prime g}) + \theta_g(x_{i,g}), x_{i\prime g} \in s_g \tag{29}$$

where $f(x_{i\prime g})$ is the original fitness function, $G(x_{i\prime g})$ represents the constraints violation function and $\theta_g(x_{i,g})$ is an additional penalty term.

The penalty term has three expressions depending on the number of feasible points in the population:

$$\Theta_g(x_{i,g}) = \begin{Bmatrix} 0 & if & x_{i,g} \in \Omega \\ -f(x_{i,g}) & if & s_g \in \Omega = \emptyset \\ -f(x_{i,g}) + max_{y \in s_g \cap \Omega f(y)} & if & s_g \in \Omega = \emptyset, \quad x_{i,g} \notin \Omega \end{Bmatrix} \tag{30}$$

To direct the search process to a feasible region, the penalty term is used. Although the feasible points are favored, the randomized distribution is still maintained, ensuring the maximal probability of finding the global minimum.

For a totally feasible population, there is no need to penalize any member, so the penalty term is reduced to zero. Because of the nature of the population, the constrained violation function is also zero.

Secondly, if the population consists of only unfeasible members, the evaluation criterion is based only on the constrained violation function.

In the third case, for a mixt population, the modified fitness function for an unfeasible member will be expressed by the sum between the maximum value for a feasible point of the original fitness function and the constrained violation function.

This procedure ensures that the feasible point will always have a better fitness function than an unfeasible one.

The newly generated population requires four main steps. An initial population must be generated with a minimum number of feasible members, to ensure a high convergence rate. This is done by imposing an **mp** parameter.

The second phase generates a mutated population by varying the initial population using a sum between a scaled difference of two random members. In our code, the following method was used:

$$\hat{X}_i^{g+1} = X_{r1}^g + F \cdot \left(X_{r2}^g - X_{r2}^g\right) \tag{31}$$

where *F* represents the scaling factor. However, other variants can be used. For example:

$$\begin{aligned} \hat{X}_i^{g+1} &= X_{r1}^g + F^2 \cdot \left(X_{r2}^g - X_{r2}^g\right) \\ \hat{X}_i^{g+1} &= X_{r1}^g + F \cdot \left(X_{r2}^g - X_{r2}^g\right) + F^2 \cdot \left(X_{r2}^g - X_{r2}^g\right) \\ \hat{X}_i^{g+1} &= X_{r1}^g + F_1 \cdot \left(X_{r2}^g - X_{r2}^g\right) + F_2 \cdot \left(X_{r2}^g - X_{r2}^g\right) \end{aligned} \tag{32}$$

After that, a cross-over population is generated. Using the initial and the mutated population, a trial generation is created using the following rule: considering a cross-over parameter $cr \in (0,1)$ and two randomly generated numbers $R_i \in rand(0,1)$, $I_i \in \overline{1,np}$**,** a new member is obtained using the following expression:

$$Y_{i,g} = \begin{cases} \hat{X}_{i,j} & if\ \ R_i < cr\ \ or\ \ \ I_i = j \\ \quad X_{i,j}\ otherwise \end{cases} \tag{33}$$

The number I is generated to ensure that at least one member from the parent population is chosen.

The final step is the acceptance phase where a selection criterion is imposed to generate the new population.

A one-to-one competition between the initial and the cross-over population takes place. The following expression was used:

$$X_i^{g+1} = \begin{cases} \hat{X}_{i,j} & if\ \ \hat{f}(X_i^g) < \hat{f}(Y_i^g) \\ \quad Y_i^g\ otherwise \end{cases} \tag{34}$$

3. RESULTS

The results of the algorithm are presented below. Using the described parametrization technique, the coefficients of the shape function, W_i, become the design parameters. Assuming an airfoil with a rounded leading edge in incompressible flow, an optimization process is done on the following cases:

1. For a fixed C_L, an objective function is imposed to reduce C_D as follows:

$$f_{obj} = \sum_{i=1}^{n} a_i \left(\frac{C_D}{C_L}\right)_{\alpha_i} \tag{35}$$

where a_i represents the weight parameter of the objective function at α_i.

Furthermore, an interval for the maximum airfoil thickness is introduced.

2. Using the same objective function as before, no additional constraints are imposed for the optimization process.

To demonstrate the capabilities of the constrained differential evolution optimizer, three cases are presented.

Starting with a NACA 2415, an optimized geometry is obtained imposing the same lift coefficient as the baseline airfoil and a maximum airfoil thickness interval.

The objective criterion is to minimize the ratio between drag and lift coefficient at different angles of attack.

A comparison between the airfoil shape is shown in *Figure 1* where we can see a change in the maximum thickness point closer to the trailing edge.

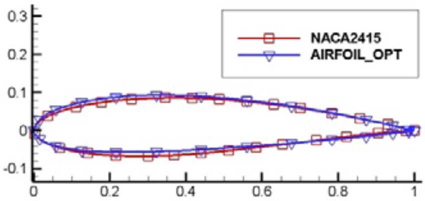
This slows the boundary layer transition which causes a decrease in the skin friction drag coefficient.

Figure 2 shows the comparison between the drag coefficient polars. It can be seen that for the desired angles of attack the drag coefficient is lower, validating the extension of the predominantly laminar flow regime area.

Figure 3 compares the lift-drag curves for the NACA 2415 and the obtained airfoil which shows improved aerodynamic performance.

As presented in the table below, the objective function is stated for three angles of attack at a given Reynolds number.

However, the algorithm can be easily extended to include corresponding Reynolds numbers for different angles of attack.

<table>
<tr><td rowspan="11">
Figure: 1 Airfoil geometry comparison</td><td colspan="2">Objective function</td><td colspan="4">$0.1\left(\frac{C_D}{C_L}\right)_{0^o}+0.8\left(\frac{C_D}{C_L}\right)_{2^o}+0.1\left(\frac{C_D}{C_L}\right)_{4^o}$</td><td colspan="2">$Re=10^6$</td></tr>
<tr><td colspan="2">N_{pop}</td><td colspan="2">N_{iter}</td><td colspan="2">F</td><td colspan="2">C_r</td></tr>
<tr><td colspan="2">40</td><td colspan="2">50</td><td colspan="2">0.4</td><td colspan="2">0.7</td></tr>
<tr><td colspan="8">NACA 2415</td></tr>
<tr><td colspan="2">W_i</td><td>0.2421</td><td>0.2421</td><td>0.2463</td><td>-0.187</td><td>-0.180</td><td>-0.127</td></tr>
<tr><td colspan="8">AIRFOIL_OPT</td></tr>
<tr><td colspan="2">W_i</td><td>0.2030</td><td>0.2382</td><td>0.2415</td><td>-0.188</td><td>-0.188</td><td>-0.052</td></tr>
<tr><td>α</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr>
<tr><td>C_l</td><td>0.25911</td><td>0.38140</td><td>0.50352</td><td>0.62561</td><td>0.74737</td><td>0.86907</td><td>0.98980</td></tr>
<tr><td>$C_d(opt)$</td><td>0.00498</td><td>0.00499</td><td>0.00508</td><td>0.00549</td><td>0.00590</td><td>0.00660</td><td>0.00704</td></tr>
</table>

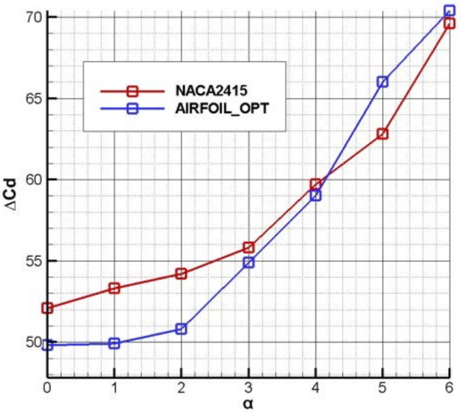


Figure: 2 CD vs α

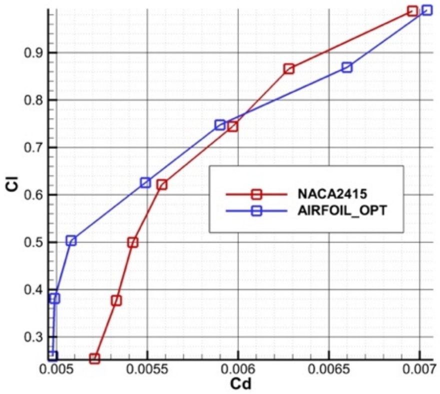


Figure: 3 CL vs CD

A similar approach is done for a second airfoil, NACA 64215. *Figure 4* shows that the maximum airfoil thickness is increased, which speeds up the boundary layer transition, which in turn, moves the boundary layer separation point closer to the trailing edge.

Also, it is noticeable that the geometry of the airfoil tends to a symmetrical design. When compared to the baseline airfoil, as shown in *Figure 5* and *Figure 6*, the lift-drag ratio is higher for the optimized airfoil indicating improved performance.

In the case of the objective function, it can be seen that the weight parameters are slightly modified, and the obtained design parameters are presented below.

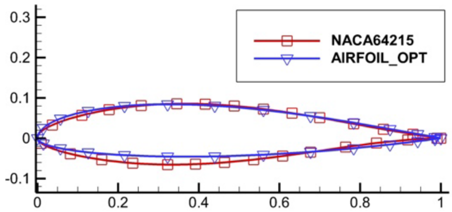


Figure: 4 Airfoil geometry comparison

Objective function	$0.15\left(\frac{C_D}{C_L}\right)_{0^o}+0.7\left(\frac{C_D}{C_L}\right)_{2^o}+0.15\left(\frac{C_D}{C_L}\right)_{4^o}$					$Re = 10^6$	
N_{pop}		N_{iter}		F		C_r	
30		60		0.6		0.6	
NACA 64215							
W_i	0.2421	0.2421	0.2463	-0.187	-0.180	-0.127	
AIRFOIL_OPT							
W_i	0.2129	0.2129	0.1937	-0.128	-0.124	-0.128	
α	0	1	2	3	4	5	6
C_l	0.17016	0.29082	0.41146	0.53189	0.65222	0.77229	0.89212
$C_d(opt)$	0.00491	0.00469	0.00471	0.00472	0.00501	0.00549	0.00606

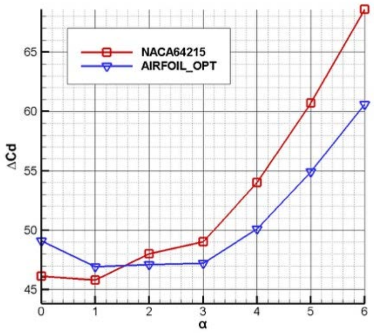


Figure: 5 CD vs α

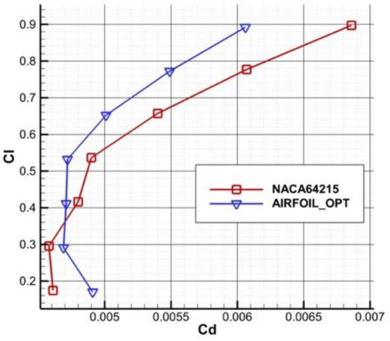


Figure: 6 CL vs CD

In the third case, an airfoil configuration is obtained starting from a random geometry with no other constraints imposed.

The objective function is to obtain the best ratio between lift and drag coefficient for specified angles of attack.

It is noteworthy that the obtained airfoil geometry, see *Figure 7*, is symmetrical which translates to the benefits of lower cost and ease of construction.

Also, *Figure: 9* shows that the aerodynamic performance is superior compared to similar airfoils.

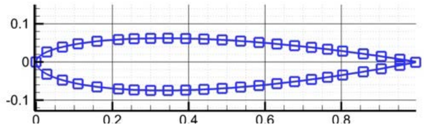


Figure: 7 Optimized airfoil geometry

Objective function	$0.15\left(\frac{C_D}{C_L}\right)_{0^o}+0.7\left(\frac{C_D}{C_L}\right)_{2^o}+0.15\left(\frac{C_D}{C_L}\right)_{4^o}$					$Re = 10^6$	
N_{pop}		N_{iter}		F		C_r	
40		80		0.5		0.7	
AIRFOIL_OPT							
W	0.1604	0.1623	0.1604	-0.193	-0.193	-0.193	
α	0	1	2	3	4	5	6
C_l	0.02468	0.04678	0.16837	0.28960	0.41079	0.53131	0.65219
$C_d(op)$	0.00480	0.00483	0.00510	0.00533	0.00558	0.00580	0.00662

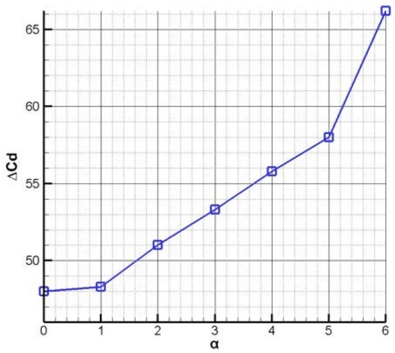


Figure: 8 CD vs α

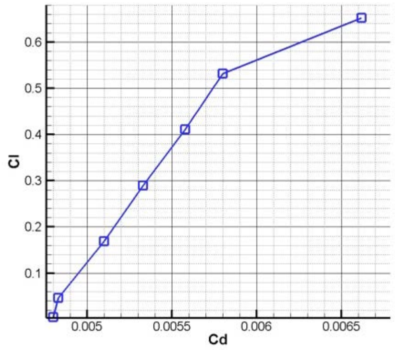


Figure: 9 CL vs CD

4. CONCLUSIONS

This paper presents an optimization technique for solving aerodynamic problems with a high efficiency in-house code that can be modified at any time for other engineering problems. The method described proves to be capable of preventing a local optimization, directing the search process to a global solution.

Reference work performed in [8] showed that airfoil parametrization based on thickness (Class Shape) and polynomial camber line brings severe limitations from the robustness point of view. The resulting optimized airfoils parametrized with Class Shape satisfy the imposed constraints and provide results comparable to those obtained in [9], where the authors used the genetic algorithm from Matlab. Moreover, the same results can be achieved starting from an arbitrary shape and imposing the same constraints proving a high convergency rate of the algorithm.

The technique can be modified using other optimization schemes, but a specific set of rules should be followed:

- An initial feasible solution shouldn’t be inserted by the user in order to converge.
- The sensitivity of the algorithm shouldn’t be very high if one or more initial parameters are changed.

Taking everything into account, the methodology can be easily modified to accommodate more refined objective functions at different Reynolds numbers. Also, the parametrization technique used proves to be flexible enough to obtain satisfactory results. To speed up the searching process, constraints must be implemented. The aerodynamic model can be changed to a simplified one in order to accelerate the optimization process, but it should be noted that the results are of lower quality.

REFERENCES

[1] M. Kulfan, Universal Parametric Geometry Representation Method, *Journal of Aircraft*, Vol. **45**, 2008.

[2] H. Liu, Linear Strength Vortex Panel Method for NACA 4412 Airfoil, *IOP Conf. Ser.: Mater. Sci. Eng.* **326** 012016, 2018.

[3] T. Cebeci, J. Cousteix, *Modeling and Computation of Boundary-Layer Flows*, Horizons Publishing Inc, 1999.

[4] S. Dănăilă, C. Berbente, *Metode numerice în dinamica fluidelor*, Editura Academiei Române, 2003.

[5] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, *Journal of Global Optimization*, **11**, 341–359 1997, https://doi.org/10.1023/A:1008202821328.

[6] Z. Kajee-Bagdadi, *Differential Evolution Algorithms for Constrained Global Optimization*, Master Thesis, Faculty of Science, University of the Witwaterstrand, Johannesburg, 2007.

[7] M.-V. Hothazie, G. Ichim, M.-V. Pricop, Development and validation of constraints handling in a Differential Evolution optimizer, The 38th "Caius Iacob" Conference on Fluid Mechanics and its Technical Applications, 7 - 8 November, 2019, Bucharest, Romania, *INCAS BULLETIN*, vol. **12**, issue 1, pp. 59 – 66, https://doi.org/10.13111/2066-8201.2020.12.1.6, 2020.

[8] A. Dina, S. Danaila, M.-V. Pricop, I. Bunescu, Using genetic algorithms to optimize airfoils in incompressible regime, International Conference of Aerospace Sciences "AEROSPATIAL 2018", 25 - 26 October 2018, Bucharest, Romania, *INCAS BULLETIN*, vol. **11**, issue 1, pp. 53 – 60, https://doi.org/10.13111/2066-8201.2019.11.1.4, 2019.

[9] M. V. Pricop, M. Boşcoianu, M. G. Cojocaru, A. Dumitrache, *Airfoil optimization based on low fidelity flow models*, ICNPAA 2018, Erevan, Armenia, AIP Conference Proceedings 2046, 020076 (2018); doi: 10.1063/1.5081596, View online: https://doi.org/10.1063/1.5081596.