AIRFOIL SHAPE OPTIMIZATION FOR DRAG COEFFICIENT THROUGH OPENFOAM AND DAKOTA SOFTWARE

TỐI ƯU HÓA HỆ SỐ LỰC CẢN BIÊN DẠNG HÌNH HỌC CÁNH THÔNG QUA PHẦN MỀM OPENFOAM VÀ DAKOTA

Nguyen Ngoc Hoang Quan[1], Luu Van Thuan[2], Ngo Khanh Hieu[3]
[1] Vietnam Aviation Academy, Ho Chi Minh City, Vietnam,
[2] DFM Engineering Vietnam, Ho Chi Minh City, Vietnam
[3] Ho Chi Minh City University of Technology
quannnh@vaa.edu.vn, thuanlv@dfm-engineering.com, ngokhanhhieu@hcmut.edu.vn

Abstract*: Today, shape optimization is one of the areas that is focused on research and development in industries. Thanks to the strength of computer technology, the shape simulation and optimization model could be analyzed quickly, robustly and exactly. Such processes have generally two major ingredients: a suitable parameterization of the geometry to be optimized and an optimization algorithm. There are many ways to accomplish this process, one of which is the modern optimization method by coupling computational fluid dynamics (CFD) and optimization algorithm. At the same time, it is necessary to define the objective function, the design variable and the algorithm for the optimal phase. In this paper, a method of optimizing geometry by combining CFD and evolution algorithms (EA) is presented with the goal of reducing the drag coefficient. The initial geometry was built by a list of control points, and they are connected by BSpline curve. The control points are moved automatically through the EA method by Dakota (Design and Analysis toolKit for Optimization and Terascale Applications) software. The control points are adjusted their positions through the iterative loop in order to achieve a better result meet the objective function. Using this methodology, we finally find a new geometry has a smaller drag coefficient than the initial geometry.*

Keywords*: CFD, control point, drag coefficient, evolutionary algorithm, geometry optimization.*

Classification number: *2.1*

Tóm tắt*: Ngày nay, trong các ngành công nghiệp tối ưu hóa hình học là một trong những lĩnh vực đang được tập trung nghiên cứu và phát triển. Với sự phát triển hết sức mạnh mẽ của ngành khoa học máy tính, việc nghiên cứu tối ưu hoá dựa trên nền tảng mô phỏng số đã đạt một tầm cao mới với mức chính xác, hiệu quả, nhanh chóng và tiết kiệm nhiều thời gian, chi phí. Quá trình này gồm hai giai đoạn chính: Tham số hóa hình học và tối ưu hóa dựa trên thuật toán tối ưu. Có nhiều phương pháp để thực hiện quy trình này, một trong những phương pháp là kết hợp giữa quá trình mô phỏng số và thuật toán tối ưu. Để thực hiện quy trình này, hàm mục tiêu, biến thiết kế và thuật toán tối ưu phải được lựa chọn. Trong bài báo này, một phương pháp tối ưu hóa biên dạng 2D cánh bằng việc kết hợp mô phỏng số thông qua phần mềm OpenFOAM và thuật toán tiến hóa thông qua phần mềm DAKOTA với hàm mục tiêu giảm thiểu hệ số lực cản. Hình học ban đầu được xây dựng bằng các đường cong B - Spline thông qua các biến điều khiển. Các biến này sẽ được điều chỉnh vị trí qua mỗi vòng lặp để đạt được giá trị tối ưu, từ đó xây dựng nên hình học mới có hệ số lực cản nhỏ hơn hình học ban đầu.*

Từ khóa*: Mô phỏng số, điểm điều khiển, hệ số lực cản, thuật toán tiến hóa, tối ưu hóa hình học.*

Chỉ số phân loại: *2.1*

1. Introduction

In the process of a new fluid dynamic or mechanical product design, the geometric selection for optimizing the physical properties is important. This stage requires a lot of testing. However, testing a real prototype is time and resource consuming. Therefore, reducing the design search time and space before manufacturing a prototype is an advantage to any engineer. One of the solutions to this requirement is to apply the development of computer science. Across all industries, geometry optimization processes for fluid dynamic or mechanic devices are getting increasingly important. Faster, more effective and less expensive product design requirements push such disciplines towards optimization at early design stages.

At present, many studies for geometry optimization is performed by numerical simulation in the world. For example, research by Manuel J. Garcia, Pierre Boulanger and Santiago Giraldo [1]. This article investigates the use of coupled CFD and EA to optimize the shape of aerodynamic profiles. The objective is to reduce the drag coefficient on a given airfoil while preserving the lift coefficient within acceptable ranges. Besides in the field of geometry optimization, the article of Jong-Taek Oh and Nguyen Ba Chien [2] is also very noticeable. They demonstrate an optimization model basics by coupling CFD and genetic algorithms GA) in which an automated procedure to optimize the flow distribution in a manifold is established. After evaluating the results, the advantages and disadvantages of the method in the paper were analyzed, from that the authors have an overview of the optimization method based on the coupling between CFD and the optimal algorithm. Other interesting studies on CFD optimization are presented in [3]. In this paper, the author uses a stand-alone GA and a surrogate-based optimization (SBO) combined with a GA are the optimal algorithms. The two optimization methods have been used in conjunction with CFD analysis to optimize the shape of a bumpy airfoil. Then, the result of two methods was compared for accuracy and performance. From these articles and research, we see the problem of geometric optimization is rapidly developed, with many different methods, used for much different geometry in the world. However, there aren't many studies on this problem in Vietnam, especially the use of Dakota software and code coupling between OpenFOAM and Dakota. So, we wish to carry out a basic research on this field to understand basic knowledge or just to understand how the code works. The most important purpose of the paper is to build a new method that can potentially be applied to a lot of geometry optimization problems. After that, the model will be connected to other software to optimize for a certain target. The objective function is minimum drag coefficient. Airfoil is constructed from B-Spline curves based on control points. The input file of geometry and mesh is built from the blockMesh file. The aerodynamic information is obtained by solving the Navier-Stokes equations using the OpenFOAM toolkit. The optimal algorithm is used by the evolutionary algorithm. With the right model, this method can improve the aerodynamic behavior of a given shape. Finally, the results of the aerodynamic optimization of an airfoil are presented and discussion about the method and the possibility of improvement follow.

This report will do mainly two things. Firstly, it will describe and discuss the optimal model, their capabilities in terms of parameterized shape optimization and their limitations. Secondly, it will compare the result with a related article. This model will be developed as an analytical tool for the design of the Unmanned aerial vehicle, the hovercraft, the centrifugal fan...

2. Basic Definitions

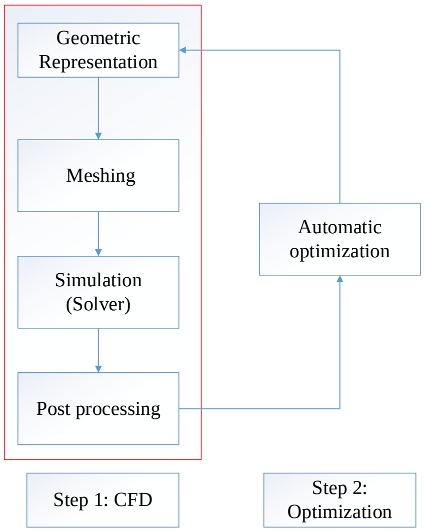

Figure 1. *Typical optimization loop*

The geometric optimization is a process of changing the shape of an object under certain conditions to achieve a defined objective function. Any change from new geometry will induce the system to change physical characteristics of its. This process has two main stages: CFD and Automatic optimization. In this paper, a geometric optimization method is demonstrated by coupling CFD using OpenFOAM and evolutionary algorithms (EAs) using Dakota (Figure 1). The two software are linked via a control file. As mentioned above, the working flow of coupled procedure between 2 programs includes the following steps:

- Step 1: Declare variable: All simulation and optimization variables, as well as control files, must be declared and constructed.
- Step 2: Geometry and mesh generation;
- Step 3: Simulation.
- Step 4: Post-processing: The results are calculated by the average value of some loops through a control command.
- Step 5: Evolutionary operators: the EA block adjusts the input variables declared in step 1 to improve results, satisfying the objective function based on the evolutionary operators. The coupled procedure will stop if it reaches the desired value or finishes a predefined evaluation step. All steps are automatically driven by an interface script.

2.1. A brief introduction about CFD

CFD is a science-based computer CFD is a science that, based on computer technology solving the equations of fluid motion that predict and analyze the physical properties of fluid flows.

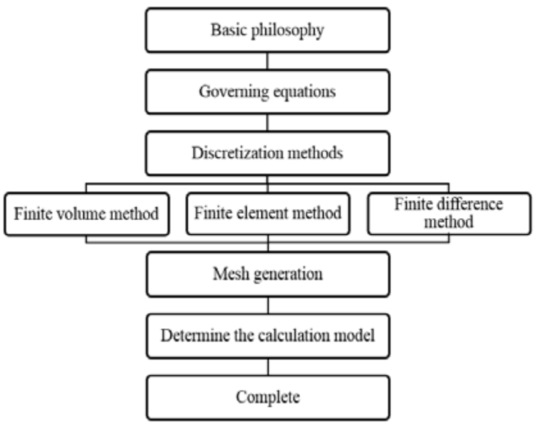

Figure 2. *Flow Chart of CFD Methodology.*

2.1.1. The Governing Equation of CFD

Almost the simulations of the fluid are based the basic equations of fluid dynamics as the continuity equation, the momentum equation, the energy equation and the basic laws of physics as the law of conservation of mass, the law of conservation of momentum and the law of conservation of energy [4]. However, this model is the incompressible flow, so it needn't the energy equation.

2.1.2. Discretization methods

In order to solve the governing equations of the fluid motion, first, their numerical analog must be generated. This is done by a process referred to as discretization. In the discretization process, each term within the partial differential equation describing the flow is written in such a manner that the computer can be programmed to calculate.

2.1.3. Meshing

Meshing is defined as the process of dividing the entire component into a smaller number of elements, but still accurately representing the geometry involved in the problem. The dimensions of these smaller elements should be selected appropriately according to the requirement to ensure the accuracy of the simulation results.

2.1.4. Turbulence models

Turbulence models are used to predict the effects of turbulence in fluid flow without resolving all scales of the smallest turbulent fluctuations. Some models have been developed that can be used to approximate turbulence based on the Reynolds Averaged Navier-Stokes (RANS) equations.

2.2. Optimization methods

The main components of EAs are discussed, explaining their role and related issues of terminology.

2.2.1. Evolutionary algorithm theory

The process of evolution through natural selection was proposed by Darwin to account for the variety of life and its suitability (adaptive fit) for its environment. The common underlying idea behind all these techniques is the same: given a population of individuals the environmental pressure causes natural selection (survival of the fittest) and this causes a rise in the fitness of the population. Currently, EA is used in many different fields. Most commercial

Solver products are based on EA. According to Eiben. A.E and Smith. J. E, published in Introduction to Evolutionary Computing [5], the evolutionary process makes the population adapt to the environment better and better. The evaluation (fitness) function represents a heuristic estimation of solution quality and the search process is driven by the variation and the selection operators. It consists of three main steps:

- Step 1: Generate the initial population of individuals randomly. (First generation);
- Step 2: Evaluate the fitness of each individual in that population (time limit, sufficient fitness achieved...);
- Step 3: Repeat the following generational steps until termination:

– Select the best individuals for reproduction. (Parents);

– Breed new individuals through crossover and mutation operations to give birth to offspring. Evaluate the individual fitness of new individuals;

– Replace the least-fit population with new individuals.

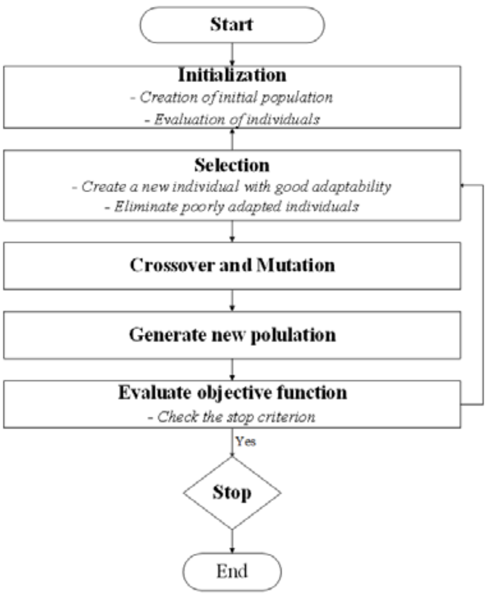

Figure 3. *General scheme of an EA.*

2.2.2. The evolutionary operators

In the EA process, the population is generated toward the best solution by improving the quality of individuals. At the start point, EA will randomly produce several individuals, called the initial population. Each individual represents a point in a search space and a possible solution. Evolutionary operators are the components that perform the actual evolution of a population including selection, crossover, and mutation.

The above operators will be continuous and repeated throughout the evaluation process until the stop condition is reached. Population quality is improved after each iteration, but there is no way to ensure that the current result is the best solution. Therefore, it is important to determine the appropriate stop conditions, which are the constraints of computational time or when the results are located around the best-known space.

2.2.3. Objective function

An objective function can be defined as a mathematical equation to be optimized given certain constraints and the relationship between one or more design variables that use to select better solutions over poorer solutions. It uses the correlation of variables to determine the value of the final outcome. The objective function shows how much each variable contributes to the optimized value in the problem. It can be represented in the following way:

$$\text{maximize or minimize } F = \sum_{j=1}^{n} c_j X_j \qquad (1)$$

Where:

X_j: The jth decision variable;

c_j: The weighted coefficient corresponding to the j^{th} variable.

In a shape optimization process, multiple objective functions can be built. However, the more the objective function increases the complexity of the problem. This requires an increase in the number of control variables, the dependent variables as well as various approaches that must be used.

2.2.4. Change and update the geometry

As described in previous sections, this geometry takes a set of control points and their movement will alter onto the boundary of the geometry. To be able to do this, it is necessary to determine the moving area of the control points. Each control point will have a region of influence which is determined via two bounding points (figure 4). This paper will analyze two cases: the initial control points are moved a small distance vertically - the *Y* direction (i.e., there is no horizontal movement - the *X* direction) and the initial control points are moved both vertically and horizontally.

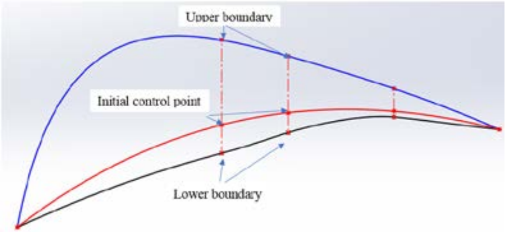

Figure 4. *Schematic figure of movement of a control point.*

3. Airfoil shape optimization for drag coefficient

Drag is a restrictive force that opposes the motion of an aircraft.

$$C_d = \frac{D}{\frac{1}{2} \times \rho \times V^2 \times S} \tag{2}$$

Where: ρ: Density, *V*: Velocity, *S*: Reference area.

The problem of reducing drag is extremely important. Minimizing drag, in other words, the drag coefficient is minimum under the same set of velocity, density, and area conditions. Minimizing aerodynamic drag will help to reduce energy loss, increase the speed and performance of the object.

3.1. Numerical model

3.1.1. Geometric and mesh representation

The geometry of the airfoil is referred from an article by Manuel J. Garcia and Pierre Boulanger [1]. Airfoil geometry is constructed from 11 control points (figure 5). And the control points are used as the design variables during the geometry optimization process. Control points 1 and 7 will be fixed to keep the chord length of the airfoil. The result of optimizing the shape depends on the accuracy and relevance of the selected control point. The size of the search space, number of loops, number of evaluation step and computation time significantly increases with a large number of control points. Vice versa, reducing the number of control points will also reduce the size of the search space, thus, providing faster computation. However, if there are not enough control points, the accuracy of the airfoil geometry will not be guaranteed. Furthermore, the control points are given by an input file and the geometry in 2D is built by employing a 2D meshing tool.

```
//Control point of airfoil
x01 0;                                  //fixed point
y01 0;                                  //fixed point
x02 8.98819494;
y02 17.00469313;
x03 23.44218411;
y03 26.47873645;
x04 40.36814214;
y04 29.58276464;
x05 56.35841153;
y05 27.4504332;
x06 66.43976531;
y06 16.03299638;
x07 81.62252704;                         //fixed point
y07 0.364386280;                        //fixed point
x08 55.14379059;
y08 -26.47873645;
x09 40.56814214;
y09 -28.89231824;
x10 23.32072201;
y10 -25.26411551;
x11 9.7169675100;
y11 -15.91153429;
```

Figure 5. *The control point of the airfoil.*

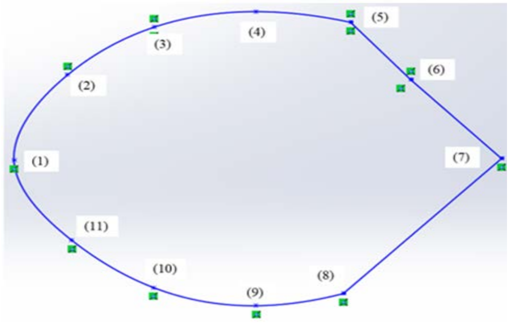

Figure 6. *The geometry of the airfoil.*

In this model, the search space of control points with lower bound (LB) and Upper bound (UP) are given as follows:

Table 1*: The search space of the control point.*

	x02	y02	x03	y03	x04	y04	x05	y05	x06	y06	x08	y08	x09	y09	x10	y10	x11	y11
LB	8.5	16	23	25	30	27	53	19	66	10	55	-27	35	-29	22	-26	9	-16
UB	9.5	18	24	27	42	30	57	28	67	17	56	-13	41	-17	24	-15	10	-11

The computational domain is a prerequisite for all simulation problems. Therefore, the size of the computational domain should be reasonably calculated. The expansion of the computational domain will limit the influence of the boundary conditions in the simulations. However, the large size of a computational domain will increase the mesh point number and run time of the simulation as well as require a computer with higher configurations. The size of the computational domain in the model is selected so that it is still possible to simulate the properties of the fluid flow after breaking out of the wing profile (a rough indication was that at least 7 – 10 times the model length would be required in each direction in order to obtain a somewhat accurate result). In fact, many dimensions were selected during the simulation process and the final dimensions as shown in figure 7 were selected.

To reduce the computational cost as well as enhance the accuracy and stability of simulation, the mesh was automatically generated with hexahedral meshes.

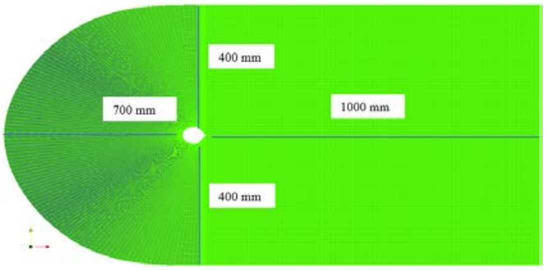

Figure 7. *The size of the computational domain and the computational mesh.*

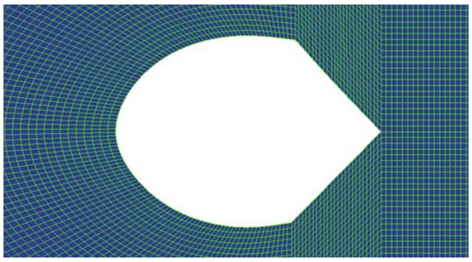

Figure 8. *Structured mesh around airfoil.*

Note that each change control point creates a new CFD domain with the same mesh size and the number of elements of its is equal to the number of elements in the initial mesh.

The mesh was evaluated by four criteria as "Max Aspect Ratio (MAR)", “, “Min volume (MV)”, “max skewness (MS) ” and "Non Orthogonal quality (NO)”. Thus, according to the criteria mesh on OpenFOAM, the mesh has a good quality.

Table 2*. Mesh criteria.*

	MAR	MV	MS	NO
Mesh value	9.963	$1.3x10^{-9}$	0.938	Ave: 6.83
Threshold value	1000	$1x10^{-20}$	4	70
	OK	OK	OK	OK

3.1.2. Boundary conditions and turbulence model

The current model has the following boundary condition:

Table 3*. Boundary condition.*

	Velocity	Pressure
Inlet	50 m/s	Zero gradient
Outlet	Zero gradient	0 Pa
Top and Bottom	symmetryPlane	symmetryPlane
Airfoil	0 m/s	Zero gradient

The following codes contain the information to simulate the case using simpleFoam (steady-state solver for incompressible turbulent flow) and the k–ε turbulence model. According to Valerio Marra, marketing director at COMSOL, the technique offers good convergence and isn’t memory-intensive. Marra also explained that the model is typically used for external flows with complex geometry. These are common boundary conditions for the simulation of the wing.

3.2. Optimization

3.2.1. Objective Function

The selection of the objective function is very crucial for process optimization by algorithms. In this research, it is desirable to study the behavior of solutions when the drag coefficients are minimized.

$$F(x) = \text{minimum}\left(C_d,\ \forall x \in A\right) \quad (3)$$

Where A is the displacement space of all accepted geometries.

3.2.2. Choosing an Optimization method

There is a methodology for defining the optimal drag coefficient to be implemented through a coupled application of simulation and optimization models is analyzed. The fitness type was set as the merit function. The crossover was 20%, the mutation scale was 80%, and the number of evaluation was set as 400. The control file automatically generates new geometry and updates the mesh and the boundary conditions as described in the previous part. In addition, another script drives the CFD code run and exports the results for the next evaluation. The EA is chosen to optimize the value of the drag coefficient because of the desire to discover random characteristics and controlling the final simulation results to achieve values in global optimization.

3.2.3. Results

As observed from the final geometry, several points reach their limited variations such as the 6th and 8th points when modifying the y value only, and the 9th and 11th points when modifying both x and y values. Besides that, the control points in the middle (i.e., 3-5 for the upper surface and 8-10 for the lower surface) have large variance along y-axis in both cases but smaller variance along x-axis in cases of modifying both x and y values. For other points, the variances are smaller than the mentioned-above points. In general, the final geometry seems to increase the control points of the lower surface to the upper bound while decrease the control points of the upper surface to the lower bound. Let compare the simulation results of the drag coefficient of the initial geometry. The result in Manuel J. Garcia's paper is 0.380, while the average results of our simulations from the 450 to 650 are 0.381 which provides an error of less than 1%. This significantly small error proves that the proposed simulation model is reliable and accurate.

The final geometry of the two coordinate change methods is relatively similar (figure 10). However, when changing both x and y, the upper surface of the airfoil is smooth and better curved. The results show that the new design improves the drag coefficient of the airfoil about 2.4 and 3.1 times in comparison with the initial geometry with only change y coordinates and the case of change both x coordinates and y coordinates, respectively. The reduction of the drag coefficient for the following reasons:

− The optimal geometry has more aerodynamic shapes (curved and smooth on the upper surface near the trailing edge) especially with no fringe creating a large drag, so the drag coefficient produces less.

− The optimal geometry has TE more curved, so the separation of the boundary layer is delayed, as well as less splitting the velocity flow when traveling towards the TE.

− The new geometry has a considerably smaller thickness than the initial geometry, which also contributes to reducing the drag on the airfoil.

The behavior of the drag coefficient through the iterative process can be observed in Figure 9. There is a particular fluctuation in the C_d Graph illustrated by Dakota. The moving a control point is a tool to evaluate the drag coefficient increased or decreased and use it to obtain the direction of the gradient. With the gradient direction, a modified shape can be obtained and finally, an optimized shape throughout the iterations is reached. In addition, because DAKOTA displays in the evaluation step. For more details, the calculation process is implemented by changing the coordinates so that they will be closer to the optimal area of the previous loop. After one step the shape will be changed according to the result and the most optimal area will be chosen and kept for the next loop. However, as many coordinates are being controlled simultaneously, the result of the optimal area in the latter loop will fluctuate within an amplitude close to the previously chosen optimal area which may increase the drag. So, the value of the drag coefficient can fluctuate and the result of the optimal area in the latter loop will fluctuate within an amplitude close to the previously chosen optimal area. However, with the trend graph, despite the oscillation, but the graph tends to converge into a point in which the drag changes within the range from 0.12 to 0.15 and the fluctuation is stable. The value of the last loop is similar and much better than the first ones.

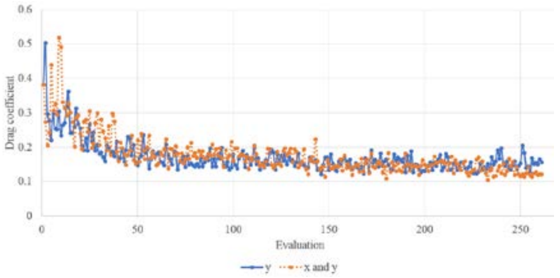

Figure 9. *Graph of the drag coefficient in the optimization process.*

Finally, comparing the results of the final geometry in the model with the final results of Manuel J. Garcia's paper, it is clear that different optimal geometry results are obtained. This difference is located mainly on the lower surface near the leading edge (control points 10 and 11) and near the trailing edge on the upper surface (control points 5 and 6) (figure 10). The reason is explained by the simulation model with optimal area size, the optimal parameters are different because Manuel J. Garcia's paper does not specify the set value of their algorithms. In EA, there are important parameters such as population size, crossover rate, mutation rate... These parameters are not published in the paper. Note that, the optimized geometry varies depending on the operation conditions derived from the CFD boundary conditions and optimal parameter selection for EA. For the same shape under different conditions, the optimum geometry achieved is different.

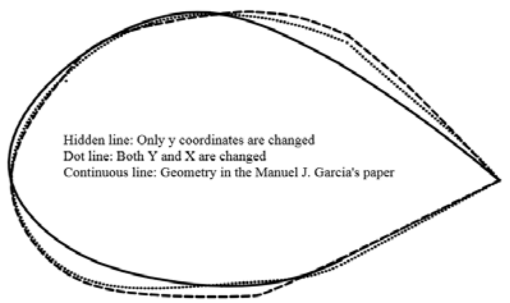

Figure 10. Comparison of optimal geometry results.

4. Conclusions and Future Work

4.1 Conclusions

The paper describes the development and validation of a shape optimization model based on two open-source software OpenFOAM and Dakota. Besides, a optimization model for the drag coefficient of the airfoil were also researched. The optimal design shows much improvement in comparison with the initial design. Although only a single geometry is produced, this framework can easily expand to multi-geometry. At the same time, there may be different solutions depending on the choice of optimization criteria, variables, objective function, and constants.This model can be applied to the design of UAV, hovercraft...

4.2 Future Works

Building optimization model lift coefficient and lift - drag coefficient ratio.Further validation and simplification of this method in shape optimization problems with shapes other than airfoils. Towards an optimal 3D modeling tool. Develop shape optimization problems with different optimization algorithms and other methods❑

References

[1] Manuel J. Garcia, Pierre Boulanger, Santiago Giraldo (2008), *CFD based wing shape optimization through gradient-based method*, International Conference on Engineering Optimization. Rio de Janeiro, Brazil.

[2] Jong-Taek Oh and Nguyen Ba Chien (2018), *Optimization design by coupling computational fluid dynamics and genetic algorithm*, Computational Fluid Dynamics - Basic Instruments and Applications in Science, Chapter 5, Publisher INTECH open science.

[3] Todd A. Johansen (2011), Optimization of a low reynold's number 2D inflatable airfoil section, Graduate thesis: Utah State University.

[4] Versteeg. H. K and Malalasekera. W, (2007), *An Introduction to Computational Fluid Dynamics*, Chapter 2, Pearson, England.

[5] Eiben. A. E, Smith. J. E (2003), *Introduction to Evolutionary Computing*, Chapter 2, Natural Computing Series.

Ngày nhận bài: 30/12/2019
Ngày chuyển phản biện: 2/1/2020
Ngày hoàn thành sửa bài: 22/1/2020
Ngày chấp nhận đăng: 30/1/2020