

Dynamic Mesh Based Airfoil Design Optimization

Zhu Xiong-feng, Hou Zhong-xi, Guo Zheng, Liu Zhao-Wei

Abstract—A method of dynamic mesh based airfoil optimization is proposed according to the drawbacks of surrogate model based airfoil optimization. Programs are designed to achieve the dynamic mesh. Boundary condition is add by integrating commercial software Pointwise, meanwhile the CFD calculation is carried out by commercial software Fluent. The data exchange and communication between the software and programs referred above have been accomplished, and the whole optimization process is performed in iSIGHT platform. A simplified airfoil optimization study case is brought out to show that aerodynamic performances of airfoil have been significantly improved, even save massive repeat operations and increase the robustness and credibility of the optimization result. The case above proclaims that dynamic mesh based airfoil optimization is an effective and high efficient method.

Keywords—unmanned air vehicles; dynamic mesh; airfoil optimization; CFD; genetic algorithm;

I. INTRODUCTION

AIRFOIL design is a crucial and core technology in the overall design process of flying vehicle, especially for the High Altitude Long Endurance (HALE) aircraft such as Gobaal Hawk. Usually high aspect ratio straight wing is employed for that vehicle, selecting an appropriate airfoil is extremely important for improving the overall aerodynamic performance for that. Recently, airfoil design methods can be categorized as forward design optimization and inverse design method. The forward design optimization method is achieved by continuously modifying the airfoil shape curve to reach optimal aerodynamic goal (lift coefficient, drag coefficient, lift-drag ratio, etc.), while aerodynamic condition is given initially for inverse design method, and meet the requirements aerodynamic goal by gradually changing the geometry and calculating gas flow equations [1-4].

The forward design optimization method is utilized for this paper, which the commonly procedure is surrogate model based optimization. A number of sample is available by Design of Experiment (DOE) after parameterize of airfoil, surrogate model (Kriging model, RSM model, etc.) is constructed based on that sample set.

Zhu Xiong-feng, Ph.D candidate, College of Aerospace and Material Engineering, National University of Defense Technology, Changsha, 410073, China (e-mail: zhuxiongfeng@yeah.net)

Hou Zhong-xi, professor, College of Aerospace and Material Engineering, National University of Defense Technology, Changsha, 410073, China (e-mail: cn_hzx@sina.com)

Guo Zheng, professor, College of Aerospace and Material Engineering, National University of Defense Technology, Changsha, 410073, China (e-mail: jason_god@sina.com@sina.com)

Liu Zhao-wei, Ph.D candidate, College of Aerospace and Material Engineering, National University of Defense Technology, Changsha, 410073, China (e-mail: vincentgrace@163.com)

Surrogate model rather than actual airfoil model is driven by some optimization algorithm, for which optimal airfoil can be obtain. However, the accuracy of surrogate model constructed is affected by the size of sampling, while the sampling scale is related to number of parameters. Difficulties of CFD operation and calculation in reality limit the size of sampling, thus accuracy and credibility of surrogate based optimization should be repeatedly verified. In this paper, CFD simulation is directly driven by optimization algorithm.

Firstly airfoil parameters are given by optimizer and airfoil shape is generated, secondary dynamic mesh is constructed and boundary condition is added to the mesh, finally CFD simulation process is carried out and the result is submit to the optimizer, a next iteration is programmed by optimizer according to its optimization strategy. The overall above process can be fully automated.

The final result of that process is that after airfoil parameters and constraints are given by designer and airfoil optimization goal is set, an optimized airfoil can be achieved by only starting the process and persisting a certain time.

II. THE OVERALL PROCESS

The basic process of dynamic mesh based airfoil optimization referred consists of four modules. Firstly is the dynamic mesh module, which is accomplished by c language programming. The import document for this module is the file Hicks.dat which describes the Hicks-Henne parameters. The export document is the file OutMesh.x, which is a Plot3D format grid file generated according to Hicks-Henne airfoil parameterize method. The initial aerofoil grid file inMesh.x is required, which is also a supporting Plot3D format document.

Secondary is the boundary condition generated module, which is working through secondary development of commercial software Pointwise.

The import document is the export document of prior module which is a Plot3D format grid mesh file. The export document is a CAE file Airfoil.cas with boundary conditions that can be directly used for CFD calculation. An auxiliary Pointwise Glyph2 script file *.glf that define the boundary condition is require.

The third module is air flow field calculation module, which is working by secondary development of commercial software Fluent. The import document of this module is the export file of the prior module which is Airfoil.cas. The export document is the iteration history file of lift coefficient cl-history and drag coefficient cd-history.

The forth module is the optimize module, that a commercial integration platform iSIGHT is employed.

The import document is the export document of prior module which forms the optimization goals. The export file is the Hicks-Henne airfoil parameter file for the first module which

forms the optimization variables.

In general, the overall process of the design process is that, the dynamic mesh module works according to the airfoil shape define by optimize module, CAE file is created by boundary generation module, aerodynamic result is reach by flow field calculation module and submit to optimize module.

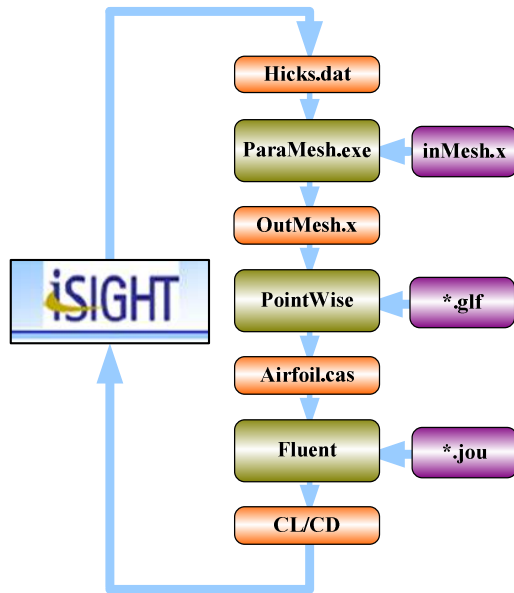


Fig. 1 The overall process

III. DYNAMIC MESH PROCESS

The optimized baseline for airfoil optimization is the standard airfoil available such as NACA series airfoil. Usually shape deformation in the optimization process appears to be small range, that a unify topology can be employ to describe the mesh. Whereas, an initial grid mesh is required for the dynamic mesh process, the shape deformation mesh is generated according to initial grid mesh and airfoil shape parameters.

A. Airfoil parameterization

Recently quite a number of airfoil parameterization methods are available, such as Hicks-Henne shape function method, orthogonal basis function method (OBF), CST method, Bezier curve parameterization method, PARSEC method, etc. For the above parametric approach, only Hicks-Henne shape function method is a perturbation based method, that a set of perturbation functions are superimposed to the baseline airfoil and forms a new airfoil. This particular principle makes Hicks-Henne method naturally applying to dynamic mesh process [5-7].

The shape of Hicks-Henne airfoil is constructed by baseline airfoil and perturbation function. Usually standard airfoil is adopted as baseline airfoil, and Hicks-Henne shape function adopted as the perturbation function $f_k(x)$.

$$f_k(x) = \begin{cases} x^{0.25}(1-x)e^{-20x} & (k=1) \\ \sin^3(\pi x^{e(k)}) & (k \geq 2) \end{cases} \quad (1)$$

Where $e(k)$ is given as $e(k) = \ln 0.5 / \ln x_k, 0 \leq x_k \leq 1$, $x_k (k=1, 2, 3, 4, 5, 6)$ as 0.15, 0.3, 0.45, 0.6, 0.75, 0.9.

Coefficients of perturbation function make up the design variables, together with the baseline airfoil to determine a new airfoil shape. The new airfoil coordinates are defined by 12 coefficients $D = [a_1, \dots, a_{12}]$ as follows

$$\begin{aligned} y_{up} &= y_{up0} + \sum_{k=1}^{k=6} a_k f_k(x) \\ y_{low} &= y_{low0} + \sum_{k=1}^{k=6} a_{k+6} f_k(x) \end{aligned} \quad (2)$$

Where y_{up}, y_{low} is the upper and lower surface vertical axis of new airfoil, and y_{up0}, y_{low0} the vertical axis of baseline airfoil, k is the coefficient numbers, a_k is the shape coefficient parameters, the 12 variable should be no more than ± 0.01 . Fig 2 shows the comparison of Hicks-Henne airfoil curve and the original airfoil curve.

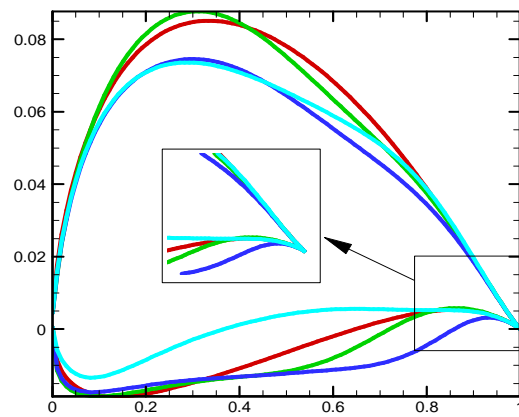


Fig. 2 Comparison of Hicks-Henne airfoil curve and Original airfoil curve

The core process of airfoil parameterization method:

- Input design variables $D = [a_1, \dots, a_{12}]$.
- Output the upper and lower curve y_{up}, y_{low} of new airfoil.

B. Initial Mesh generation

Topologies available that suitable for airfoil flow field calculation are "O" topology grid, "C" topology grid and "H" topology grid, etc. The initial mesh is use as the baseline of dynamic mesh process, which should has a good robustness as a consequence. Grid topology should be as simple as possible, in case the flow field can be definitely described. The grid topology should also be of favorable orthogonal. The "O" topology grid can meet all the need above and make it a better choice.

Commercial software Pointwise is utilized to generate the initial grid in this article.

- The core process of initial mesh generation:
- Input the geometric parameters of baseline airfoil.
- Output the Plot3D format mesh grid.

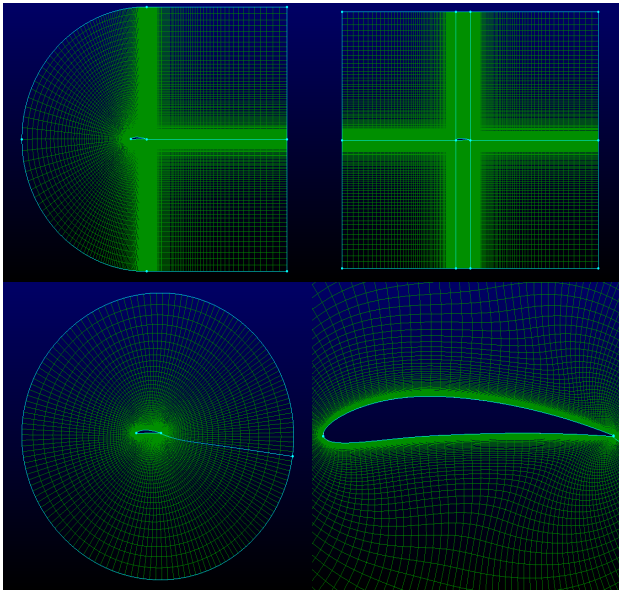


Fig. 3 Display of grid topology

C. Grid deformation

1. Plot3D format grid file

The Plot3D file is a format of structured grid file that was popularized by NASA's CFD visualization software. Only structured surface grids are supported for export. This file type will not export unstructured grids or i-blanking. Grid mesh is save to a *.x file analytically, which is logically clear and easy to read, and benefit for the follow-up dynamic mesh programming. In this article, initial Plot3D file is generated by Pointwise [8].

The following piece of pseudo-code describes the format the data found in a Plot3D file should take:

```

c.....nmax is number of networks
c.....ni(n) number of points in the i direction for the nth
network
c.....nj(n) number of points in the j direction for the nth
network
c.....nk(n) number of points in the k direction for the nth
network
integer ni(nmax), nj(nmax), nk(nmax)
real x(imax,jmax,nmax),
& y(imax,jmax,nmax),
& z(imax,jmax,nmax)
write(1) nmax
write(1) ( ( ni(n), nj(n), nk(n) ), n=1,nmax )
do 10 n = 1, nmax
write(1) ( ( x(i,j,n), i=1,ni(n) ), j=1,nj(n) ),
& ( ( y(i,j,n), i=1,ni(n) ), j=1,nj(n) ),
& ( ( z(i,j,n), i=1,ni(n) ), j=1,nj(n) )
Continue

```

2. Generation of dynamic mesh

The mesh deformation is small and grid topology does not change in the airfoil optimization process, thus a simple and efficient linear interpolated method is utilized for dynamic mesh generation.

The specific process is that: the initial Plot3D format airfoil

mesh is employed as baseline; the wall boundary is directly superimposed on the Hicks-Henne perturbation as equations (2) according to Hicks-Henne parameterize method, Hicks-Henne perturbation is then evenly distributed to the radial direction of the "O" topology grid.

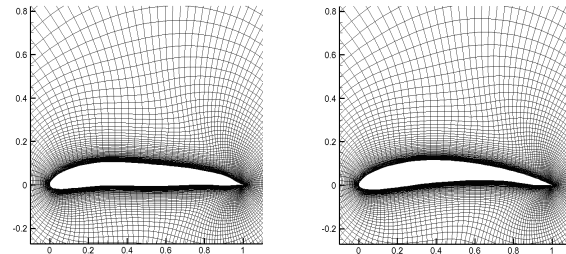


Fig. 4 Illustration of dynamic mesh

The core process of dynamic mesh generation:

- Input the Hicks-Henne parameters and the initial Plot3D format mesh grid.
- Output the new Plot3D format mesh grid.

IV. SETTING BOUNDARY CONDITIONS

The Plot3D format grid file can not be used for CFD simulation, boundary condition is required. Usually two types boundary conditions are needed for "O" topology grid, one is "wall boundary" which is set to airfoil boundary, the other is "pressure-far-field boundary" which is set to the external grid surface.

In this paper Pointwise software is utilized for setting boundary conditions. Because airfoil optimization is an automatic process, manually setting boundary condition is not proposed, whereas a piece of script code with the language Glyph2 is required. Glyph2 is the scripting language for Pointwise v16. It is an extension to the tcl programming language that allows access to the commands and entities of the Pointwise application, and offers the ability to automate all or part of the CFD meshing process [8].

It should be noted that using the following step can record a piece of Glyph2 draft code of setting boundary conditions: "Menu / Script /Begin Journaling". With the purpose of the program's simplicity and script execution reliability, the redundant syntax should be removed from the Glyph2 draft code, and refined script program is available as shown below.

Meanwhile, the automation process of setting boundary condition with Pointwise can use the following batch mode. Run Pointwise in batch mode by typing \$pointwise_home_path/hw/bin/tclsh MyScript.glf at the command line (Linux) or DOSprompt (Windows) where pointwise_home_path refers to the Pointwise installation directory and hw refers to the platform on which Pointwise is installed (i.e. win32, linux). An instance shows:

```
"$pointwise_home_path\Pointwise\PointwiseV16.04R1\win32\bin\tclsh.exe" "E:\AirfoilOptimal\imPortPlot3D.glf"
```

The core process of setting boundary conditions

- Input the Plot3D format grid file.
- Setting solvers and dimensions.
- Setting boundary conditions and adjusting orient

direction if negative volume appears.

- Output the CAE file.

```

1 # Pointwise V16.
2 #####
3 package require 2 # ##### Begin
4 #####
5 pw::Application 4 package require FWI_Glyph 2.3
6 pw::Application 5 pw::Application reset
7 pw::Application 6 set report_var "Temp"
8
9 pw::Application 7 set scriptDir "E:/AirfoilOptimal/"
10
11 set_TMP(mode_1) 8 pw::Grid import -type PLOT3D -format ASCII \
12 $TMP(mode_1) in 9 -precision Single -report $report_var [file join $
13 $TMP(mode_1) res 10 ##### Set CAE solver and Dimension
14 $TMP(mode_1) con 11 set list1 [pw::Application getCAESolverNames]
15 $TMP(mode_1) end 12 pw::Application setCAESolver [lindex $list1 3] 2
16 unset_TMP(mode_1) 13 ##### Boundary condition
17 pw::Application 14 set Wall [pw::BoundaryCondition create]
18 15 $Wall setName "Wall"
19 pw::Application 16 $Wall setPhysicalType "Wall"
20 17 $Wall setId 3
21 18 set dom [pw::GridEntity getByName "dom-1"]
22 pw::Application 19 set con1 [pw::GridEntity getByName "con-1"]
23 20 $Wall apply [list [list $dom $con1]]
24 set_TMP(PW_1) [ 21 set Farfield [pw::BoundaryCondition create]
25 set_DM(1) [pw:: 22 $Farfield setName "farfield"
26 unset_TMP(PW_1) 23 $Farfield setPhysicalType "Pressure Far Field"
27 set_TMP(PW_2) [ 24 $Farfield setId 4
28 unset_TMP(PW_2) 25 set con3 [pw::GridEntity getByName "con-3"]
29 pw::Application 26 $Farfield apply [list [list $dom $con3]]
30 pw::Application 27 ##### Orientation Reverse
31 28 set dom [pw::GridEntity getByName "dom-1"]
32 set_CN(1) [pw:: 29 $dom setOrientation [Maximum JMinimum
33 set_TMP(PW_3) [ 30 ##### Export cae file
34 unset_TMP(PW_3) 31 #####
35 pw::Application 32 set dom [pw::GridEntity getByName "dom-1"]

```

Fig. 5 Illustration of Glyph2 code

V. THE AUTOMATION OF FLUENT

Fluent is a general flow field solver, CFD simulation is carried out for the CAE file, which achieve the lift coefficient, drag coefficient, lift-drag ratio of deformed airfoil. Just like the process of setting boundary condition, the CFD calculation requires automation rather than manually setting. In order to reach automation of Fluent, it can record a script in two ways: one is “Menu/File/Write/Start Journal” the other is “Menu/File/Write/Stat Transcript”.

The script can be reloaded to realize Fluent automation. However GUI interface should be open in the setting process for these two methods, which on one hand reduce the operating efficiency, on the other hand may interrupt the automation process in case of the operator’s operating windows GUI. In this paper FLUENT 6.3 Text Command is utilized, which is of high efficiency and does not rely to auxiliary GUI.

The drawback is that it can not generate the command line automatically, and should try several times to successfully fulfill setting.

Meanwhile the automated operation of Fluent can use the batch mode script as follows [9]:

```

"G:\Program Files\Fluent.Inc\ntbin\ntx86\fluent.exe" 2d -i
WingFluent.jou

```

The core Process of Fluent automatically execution

- Input CAE file.
- Check grid and scale.
- Set solver, materials, boundary conditions, operating environment, initialization, etc.
- Set reference values and output the lift coefficient and drag coefficient history values.

```

1 (cx-gui-do 1 (cx-gui-do cx-activate 1 file
2 (cx-gui-do c 2 (cx-gui-do cx-set-text 2 read-case
3 (cx-gui-do c 3 (cx-gui-do cx-activate 3 WingMesh.cas
4 (cx-gui-do c 4 (cx-gui-do cx-activate 4 q
5 (cx-gui-do c 5 (cx-gui-do cx-set-text 5 grid
6 Reading "E:2 6 (cx-gui-do cx-activate 6 scale
7 36516 2D 7 (cx-gui-do cx-set-text 7 71.000
8 204 2D 8 (cx-gui-do cx-activate 8 81.000
9 204 2D 9 (cx-gui-do cx-activate 9 q
10 18564 nod 10 (cx-gui-do cx-activate 10 define
11 18564 nod 11 (cx-gui-do cx-activate 11 models
12 18564 nod 12 (cx-gui-do cx-activate 12 viscous
13 Building... 13 (cx-gui-do cx-activate 13 spalart-allmaras
14 grid, 14 (cx-gui-do cx-activate 14 yes
15 materia 15 (cx-gui-do cx-activate 16 interfa 16 (cx-gui-do cx-activate 17 domains 17 (cx-gui-do cx-activate 18 mixture 18 (cx-gui-do cx-activate 19 zones, 19 (cx-gui-do cx-activate 20 farfield-5 20 (cx-gui-do cx-activate 21 wall-4 21 (cx-gui-do cx-set-list 22 interior-3 22 (cx-gui-do cx-activate 23 dom-1 23 (cx-gui-do cx-activate 24 shell c 24 (cx-gui-do cx-activate 25 Done, 25 (cx-gui-do cx-set-list 26 (cx-gui-do cx-activate 27 (cx-gui-do cx-activate 28 28 (cx-gui-do cx-activate 29 Grid Check 29 (cx-gui-do cx-activate 30 Domain Exte 30 (cx-gui-do cx-activate 31 x-coordin 31 (cx-gui-do cx-activate 32 no

```

Fig. 6 FLUENT 6.3 Text Command

VI. iSIGHT BASED INTEGRATION

The airfoil optimization in this article is a single object optimization problem, with Hicks-Henne parameters as optimization variables and lift-drag ratio’s maximum as single object, which is typical simulation software based discrete optimization problems. Commonly, iSIGHT is utilized for the integration of numbers of software, and performing optimization process.

$$\begin{aligned}
 & \text{find : } Hick_i \\
 & \text{max : } K \\
 & \text{s.t.} \{ -0.01 < Hick_i < 0.01, i = 1, 2, \dots, 12
 \end{aligned}
 \tag{3}$$

iSIGHT, a platform developed by Engineous Software, Inc., is one of the most powerful computer aided engineering (CAE) software. It efficiently integrates digital technology, reasoning techniques and design exploration technology. A lot of traditional manual work can be complete automated by this software, which substitute for huge repetitive, error-prone digital processing and design process of engineering designers, that iSIGHT is known as “software robot” [10].

As shown in Fig 7, “ParaMesh” is dynamic mesh module, “Pointwise” is boundary setting module, “Fluent” is flow field solver module, “Cl/Cd” is the lift-drag ratio calculation module, “Save History” is post-processing module which saving the CFD simulation results and calling Tecplot script to finish graphics processing, “AirfoilOptimal” module is the optimization module.

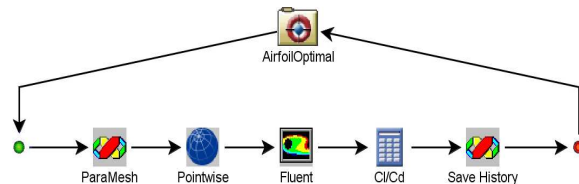


Fig. 7 Integration process

The core process of iSIGHT based integration:

- Set input and output parameters and automatically execution script for all modules.
- Set parameters exchange relations for all modules.
- Set optimizer.

VII. OPTIMIZATION CASE STUDY

The baseline airfoil of this optimization case is NACA6412. The flight environment is sea level with 2° angle of attack. Initial mesh grid is “O” topology structured grid. Solver of Fluent is “Spalart-Allmaras”, the rest of flight conditions are set in the following table.

TABLE I
FLIGHT CONDITIONS

Parameter	Quantity
Density	$\rho = 1.225 \text{ kg/m}^3$
Chord Length	$L = 1 \text{ m}$
Velocity	$v = 20 \text{ m/s}$
Dynamic viscosity	$\mu = 1.78938\text{e-}005 \text{ Ns/m}^2$
Reynolds number	$Re = 1.3692\text{e+}006$
Pressure	101325 Pa
Sound of speed	340.294 m/s
Mach number	0.05877

A. Design of experiment

Sensitivities of all the 12 parameters on the overall lift-drag ratio is find out using design of experiment (DOE) method, some parameters will be removed if the sensitivity is small. Firstly, 100 Latin-Hypercube samples are collected, and then sensitivity is analyzed. The result is shown below. It can be illustrated from the chart that, the sensitivity of 12th Hicks-Henne parameter (a_{12} shown in equation (2)) is greatest, while 7th Hicks-Henne parameter (a_7 shown in equation (2)) is smallest. However none of these parameters is of enough small sensitivity that can be ignored. Thus, all the 12 parameters should be included in the options of follow-up optimization.

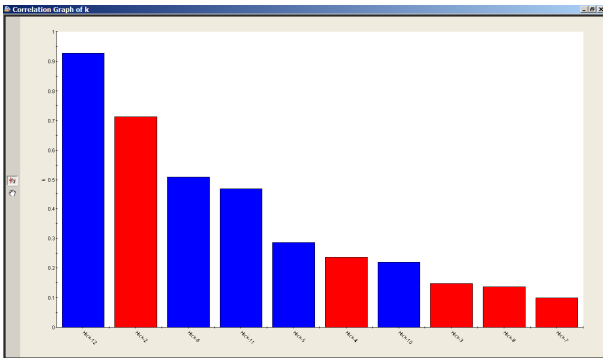


Fig. 8 Sensitivities of parameters

B. Global optimization

Optimization options of the optimizer that is “AirfoilOptimal” module is setup as: Multi-Island GA, Sub-Population Size 10, Number of Generations 100. Taking into account that the optimization process above is a serial process and can not execute parallel, the Number of Islands is chosen 1. The overall iteration number is 1000 and total duration is 25:38:58. Figure below shows the history of lift-drag ratio.

What can be illustrated is that lift-drag ratio is increasing by iteration number, and the increasing quantity become slight after 500th iterations. The optimal lift-drag ratio reaches at 866th iteration.

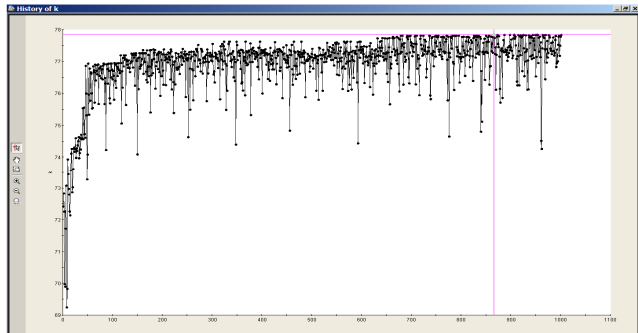


Fig. 9 History of lift-drag ratio

Table II show the Hicks-Henne parameters when lift-drag ratio reaches optimal.

TABLE II
OPTIMAL HICKS-HENNE PARRAMETERS

Parameter	Quantity	Parameter	Quantity
Hick1	0.00825	Hick7	-0.00129
Hick2	0.004787	Hick8	0.004161
Hick3	0.008506	Hick9	0.004237
Hick4	-0.0001	Hick10	0.004959
Hick5	0.004904	Hick11	-0.00246
Hick6	0.007081	Hick12	0.009582

Table III is the comparison of lift coefficient, drag coefficient, lift-drag ratio of initial airfoil and optimized airfoil. It can be illustrated that the lift-drag ratio increases 7.9340% after optimization.

TABLE III
COMPARISON OF INTIAL AND OPTIMIZED AIRFOIL

Parameter	Initial airfoil	Optimized airfoil
Lift coefficient (Cl)	0.88372	0.95587
Drag coefficient (Cd)	0.01225	0.01228
Lift-drag ratio (K)	72.11171	77.83303
Lift-drag ratio increase		7.9340%

Fig 10 is the comparison of initial airfoil curve and optimized airfoil curve, it can seen from the chart that curvature increases, thickness decreases, trailing edge becomes more curved for the optimized airfoil. These changes also meet the potential flow theory about airfoil.

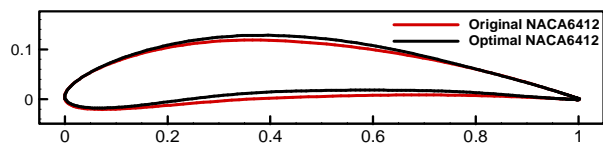


Fig. 10 Comparison of initial airfoil curve and optimized airfoil curve

Fig 11 shows the pressure distribution and x component velocity of initial airfoil and optimized airfoil.

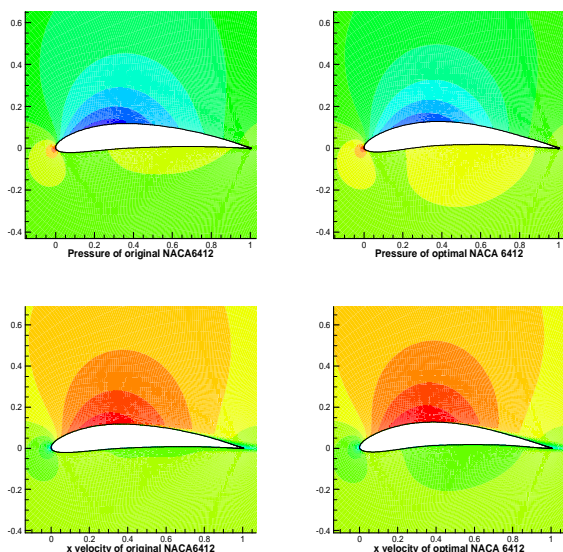


Fig. 11 Comparison of aerodynamics between initial airfoil and optimized airfoil

VIII. CONCLUSION

Standard airfoil such as NACA series airfoil is employed as the optimization baseline in this article. The new grid mesh is generated by linear interpolation of initial grid mesh. Pointwise is used for automatically setting boundary conditions, and Fluent is utilized for CFD simulation, some other post-process have been complete in this airfoil optimization platform, which including data communication between multiple procedure and commercial software. The entire process is supported by iSIGHT platform. Compared with traditional surrogate model based airfoil optimization, operation process proposed in this article can reduce huge amounts of manual sample collection, and assign the repetitive work to “software robots” iSIGHT, which enable the designers concentrate higher levels research such as increasing the CFD calculation accuracy. The optimization process is clearer in this operation, and the optimize result is more credible, which is prove to be an effective method for airfoil optimization.

REFERENCES

- [1] Oyama, A., “Multidisciplinary Optimization of Transonic Wing Design Based On Evolutionary Algorithms Coupled With CFD Solver, in European Congress on Computational Methods,” European Congress on Computational Methods in Applied Sciences and Engineering. Barcelona, pp. 1–17, September 2000.
- [2] Gopalarathnam, A. and M.S. Selig, “Low-Speed Natural-Laminar-Flow Airfoils: Case Study in Inverse Airfoil Design,” *Journal Of Aircraft*, vol. 38, 2001, pp. 57-63.
- [3] Shahrokhi, A. and A. Jahangirian, “Airfoil Shape Parameterization For Optimum Navier–Stokes Design With Genetic Algorithm,” *Aerospace Science and Technology*, 2007, pp. 443-450.
- [4] Oyama, A., S. Obayashi and T. Nakamura, “Real-coded Adaptive Range Genetic Algorithm Applied to Transonic Wing Optimization,” *Applied Soft Computing*, 2001, pp. 179-187.
- [5] Hicks, R. and P.A. Henne, “Wing Design by Numerical Optimization,” *AIAA Aircraft Systems & Technology Meeting*, Seattle, Washington, pp. 1-7, 1977.
- [6] Kulfan, B.M., “Universal Parametric Geometry Representation Method,” *Journal Of Aircraft*, vol. 45, 2008, pp. 142-158.
- [7] Derksen, A. R.W. and Rogalsky, B.T, “Bezier-PARSEC: An Optimized Aerofoil Parameterization for Design,” *Advances in Engineering Software*, 2010, pp. 923-930.
- [8] Pointwise, Inc., *Pointwise User Manual*. 2010.
- [9] Fluent, Inc., *FLUENT 6.3 Documentation*. 2006.
- [10] Dassault Systèmes Simulia, Corp., *Isight 5.0 Design Gateway Online Help*. 2011.