Two-Dimensional Gradient-Based Aerodynamic Shape Optimization with Two Geometry Parameterization Techniques using SU2 Code

Trung-Huy Nguyen, Gia-Long Hoang, Huy-Duc Nguyen, Quoc-Bao Nguyen, Van-Sang Pham [*]
Hanoi University of Science and Technology, Ha Noi, Vietnam
[*] *Corresponding author email: sang.phamvan@hust.edu.vn*

Abstract

The paper studies the effect of some shape parameterization techniques on automatic two-dimensional aerodynamic shape optimization using the discrete adjoint method. In this paper, the Hicks-Henne Bump Functions (HHBF) technique and the Free-form Deformation (FFD) control points technique are used to parameterize the shape of the NACA 0012 airfoil. First, this paper makes a full, detailed description of the shape optimization workflow, including Euler equations, geometry parameterization techniques, discrete adjoint method, gradient evaluation, optimization algorithm, and mesh deformation. Second, it explores how shape parameterization techniques are implemented in the optimization problem. Finally, the results are evaluated to compare the efficiency of the mentioned techniques. The results suggest that, in general, both techniques were shown to be equally effective as geometry parameterization methods for the shape optimization problem. However, it appears that the HHBF technique demonstrates better performance with fewer design iterations compared to that of FFD technique. On the other hand, FFD shows stability and a smoother decrease in drag values, while HHBF exhibits greater unsteadiness during the optimization process.

Keywords: Aerodynamic shape optimization, Hicks-Henne bump functions, Free-form Deformation.

1. Introduction

Aerodynamic shape optimization of aircraft has played a crucial role in the aviation industry. In this scenario, the optimization of an aircraft wing's geometric shape to minimize drag while maintaining its lift is widely studied, as it helps preserve transportation efficiency while reducing fuel consumption by preventing the engine from operating beyond necessity. This is especially important for transonic flights, where shock waves form on the wings of the aircraft, significantly increasing drag.

With advancements in numerical simulations, particularly computational fluid dynamics (CFD), the optimization of an aircraft wing's geometric shape is now formulated using mathematical models, which significantly reduce costs compared to physical testing. Moreover, drag is ensured to be minimized through the gradient evaluation of the objective function concerning the design parameters due to the accuracy and efficiency of optimization algorithms.

To represent mathematically an airfoil, several parameterization techniques are used to define the airfoil surface in mathematical form. Among these, the Hicks-Henne Bump Functions (HHBF) method employs a combination of bump functions to describe the airfoil shape, while the Free-form Deformation (FFD) method utilizes control points that represent specific locations on the airfoil surface in its parametric volume.

Earlier in the 20^{th} century, several works were conducted in the field of aerodynamic shape optimization, aiming to improve aerodynamic performance through systematic modification of geometry. These works were the foundation, as well as an inspiration, for this article to be completed. Among these, the Hicks-Henne bump function method, originally introduced by Hicks and Henne in 1978 [1], has been widely used due to its simplicity and effectiveness in controlling airfoil surface geometry with a limited number of design variables. In the work published in 1978, R. M. Hicks and P. A. Henne demonstrate this technique through three design problems on a swept wing, including shock drag reduction with volume constraint, lift-drag ratio increment, and good stall progression achievement, by modifying the wing's surface.

Later in 1986, the Free-form Deformation technique was first introduced by Sederberg and Parry [2]. This offered greater geometric flexibility by embedding the geometry in a lattice of control points, making it particularly suitable for complex three-dimensional shape modifications.

ISSN 2734-9373
https://doi.org/10.51316/jst.184.ssad.2025.35.3.6
Received: Mar 4, 2025; revised: May 15, 2025;
accepted: May 16, 2025

Later in the 21^{st} century, some researchers have also done similar work to this aeroelastic shape optimization topic. In 2017, D. A. Masters *et al.* tested seven parameterization techniques, including CSTs, B-Splines, Hicks-Henne bump functions, Radial Basis function (RBF) approach, Bezier surfaces, a singular value decomposition method (SVD), and the PARSEC method, within the analysis of 2000 airfoils [3].

Regarding to the use of SU2 code, in 2018, G. Yang and A. D. Ronch also studied the aerodynamic shape optimization of the transonic airfoils, including inviscid NACA 0012 and turbulent RAE 2822 with lift constrained, with the implementation of HHBF and FFD methods in the SU2 environment [4], except that they employed the continuous adjoint approach in their work.

The discrete adjoint solver was integrated into the SU2 framework in 2016 by T. Albring, M. Sagebaum, and N. R. Gauger to explore new and interesting optimization problems [5]. In their work, they applied the discrete adjoint method to the aerodynamic design of 3D models.

In this work, we take on the task of studying the effect of HHBF and FFD methods, with the implementation of a discrete adjoint approach, on two-dimensional lift-constrained airfoil optimization. These geometric parameterization methods are already integrated into the SU2 code, an open-source CFD simulation and optimization tool, which we will use to support this research [6].

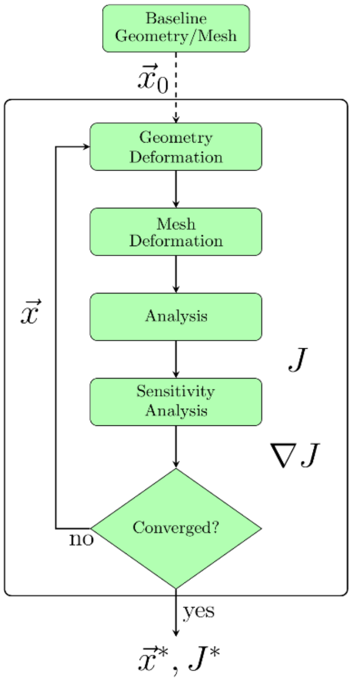

Fig. 1. Optimal design loop for NACA 0012

2. Numerical Approach

2.1. Optimization Framework

The flow solver analyzes aerodynamic forces on the NACA 0012 airfoil with initial conditions and a grid. The chosen objective function is drag, optimize the airfoil with minimum drag constraint, and the result is the optimized geometry of the NACA 0012 airfoil after reaching the specified converged criterion or the maximum number of iterations.

The optimization process starts with a baseline geometry and grid as input to the design loop, along with a chosen objective function (*J*) and a set of design variables (*x*). After analyzing the flow for the first time, if the analyzed sensitivity has converged, the process is completed. Otherwise, design variables will be optimized again until the objective function reaches the specified converged criterion, or the process reaches the maximum number of iterations. See Fig. 1 to understand the airfoil shape optimization workflow.

2.2. Numerical Flow Solver

The compressible Euler equations based on the Navier-Stokes equations with zero viscosity and zero thermal conductivity are used in this case:

$$\mathrm{R}(U) = \frac{\partial \mathrm{U}}{\partial \mathrm{t}} + \nabla \cdot \bar{F_c}(U) - \mathrm{S} = 0 \tag{1}$$

where the conservative variables are the working variables and are given by:

$$U = \{\rho, \rho\bar{v}, \rho E\}^T \tag{2}$$

and S is a generic source term. The convective flux is:

$$\bar{F_c} = \{\rho\bar{v}, \rho\bar{v} \otimes \bar{v} + \bar{\bar{I}}p, \rho E\bar{v} + p\bar{v}\} \tag{3}$$

where ρ is the fluid density, $\bar{v} = \{u, v, w\}^T \in R^3$ is the flow speed in the Cartesian system of reference, E is the total energy per unit mass, and p is the static pressure. Using the gas model, which is a perfect gas with an adiabatic index of γ and gas constant R, the fluid solver can close the system by determining the pressure from $p = (\gamma - 1)\rho[E - 0.5(\bar{v} \cdot \bar{v})]$ [7].

2.3. Geometry Parameterization Techniques

Parameterization methods are used to describe the surface of the airfoil. They are categorized as either constructive or deformative methods [3]. Hicks-Henne bump functions and Bezier surface FFD are two common deformative methods that have been integrated into SU2. The main idea is to manipulate the original airfoil shape to create a new one.

2.3.1. Hicks-Henne bump functions

Hicks-Henne bump functions are introduced to an existing airfoil baseline to mathematically deform it by 'bumping' at local positions where that function is defined. As mostly used in 2D problems, its equations are given as follows:

$$\begin{cases} y = y_{baseline} + \sum_{i=1}^{n} b_i(x) \\ b_i(x) = a_i \left[\sin\left(\pi x^{\frac{\log 0.5}{\log h_i}} \right) \right]^{t_i} , 0 \le x \le 1 \end{cases} \quad (4)$$

where *n* is the number of bump functions; $b_i(x)$ is the bump function proposed by Hicks and Henne [1]; a_i is the amplitude of the i^th^ function, which also serves as a design variable; h_i is the location of the maximum point of the bump on the *X*-axis and t_i is the width of the bump. In Fig. 2, each bump function reaches its maximum value at $x = h_i$ and that x location is represented by a vertical red dotted line.

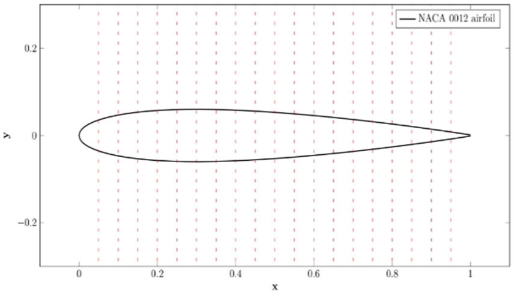

Fig. 2. Airfoil parameterized using HHBF

2.3.2. Free-form Deformation Bezier surface

FFD was first presented by Sederberg and Perry [2], which encapsulates an airfoil or a wing inside a box defined by lattices, serving as control points. By moving them, we deform the enclosed airfoil.

The problem's dimension requires the use of either a bivariate surface (2D) or trivariate volume (3D), which utilizes Bézier curves or B-Spline for the blending function.

The equations that govern the parameterized Bézier volume can be expressed as follows:

$$\boldsymbol{X}(\xi,\eta,\zeta) = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} \boldsymbol{P}_{ijk} B_i^l(\xi) B_j^m(\eta) B_k^n(\zeta) \quad (5)$$

where *l*, *m*, *n* are the degrees of the blending function; ξ, *η*, *ζ* are the coordinates in the parametric coordinate system; $\boldsymbol{P}_{ijk}$ are the Cartesian coordinates of the control point (*i, j, k*); ***X*** are the corresponding Cartesian coordinates for given Bézier volume coordinates; $B_i^j(\alpha)$ is a Bernstein polynomial [4], which is expressed as:

$$B_i^j(\alpha) = \frac{j!}{i!\,(j-i)!}\alpha^i(1-\alpha)^{j-i} \quad (6)$$

Fig. 3 describes how the NACA 0012 airfoil is encapsulated in an FFD box, defined by 40 control points in the *x*-direction and 1 in the *y*-direction.

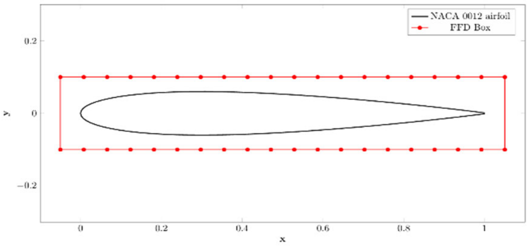

Fig. 3. Airfoil parameterized using FFD box

2.4. Discrete Adjoint Method

In the optimization framework, flow solution analysis, represented by the "Analysis" box in the flowchart in Fig. 1, is performed to evaluate how flow solutions affect the objective function. This calculation is called the adjoint solution. For the adjoint flow solver, a discrete adjoint approach is employed in this work. In SU2, the discrete adjoint method is an approach used for gradient-based aerodynamic shape optimization. It efficiently computes the gradients of an objective function (e.g., drag coefficient) concerning a large number of design variables. Unlike the continuous adjoint method, which derives adjoint equations from the continuous Navier-Stokes or Euler equations before discretization, the discrete adjoint method directly differentiates the discretized flow equations. This ensures that the adjoint solution is consistent with the numerical scheme used for solving the flow equations.

In this research, the primal flow equations are solved by discretizing them using the finite volume method (FVM). The solution provides pressure, velocity, density, and lift coefficient C_L and drag coefficient C_D. The discrete adjoint equations are derived by differentiating the discretized governing equations for the design variables.

$$\left[\frac{\partial R}{\partial U}\right]^T \boldsymbol{\lambda} = \frac{\partial J}{\partial U} \quad (7)$$

where R are the residuals of the flow equations, U are the flow variables (density, velocity, energy, etc.), J is the objective function (drag coefficient C_D), $\boldsymbol{\lambda}$ are the adjoint variables (Lagrange multipliers) [5].

By solving this linear system, the adjoint variables $\boldsymbol{\lambda}$ are obtained, providing information on how the objective function changes to small perturbations in flow variables.

2.5. Gradient Evaluation

Continuing with the workflow of aerodynamic optimization, sensitivity analysis is subsequent to flow solution analysis. In this module, the gradient of the objective function concerning the shape design variables is computed. The sensitivity analysis within the optimal shape design loop is obtained with the gradient evaluation using the discrete adjoint method, which can be described as the following equation:

$$\frac{dJ}{dx_i} = -\lambda \left[\frac{\partial R}{\partial x_i}\right]^T \quad (8)$$

where $R(U, x) = 0$ represents the flow governing equations (Euler) in SU2; λ is the adjoint variable, obtained by solving the adjoint equation [5].

This equation provides efficient gradient computation without requiring costly finite difference evaluations.

2.6. Sequential Least Squares Programming Optimization Algorithm

In this particular aerodynamic optimization using the SU2 code, this paper utilizes the Sequential Least Squares Programming Optimization Algorithm (SLSQP) to minimize the drag of the 2D NACA 0012 airfoil in transonic flight conditions while taking constraints into consideration. SLSQP optimizer is a sequential least squares programming algorithm which uses the Han–Powell quasi–Newton method with a Broyden–Fletcher–Goldfarb–Shanno (BFGS) update of the B–matrix and an L1–test function in the step–length algorithm [8].

Sequential Quadratic Programming (SQP) is one of the most successful classes of algorithms for solving non-linear optimization problems (NLP). It solves an NLP problem by iteratively formulating and solving a sequence of Quadratic Programming (QP) subproblems. The Sequential Least Squares Programming algorithm (SLSQP) has been one of the most widely used SQP algorithms since the 1980s [9].

In aerodynamic shape optimization, the problem is typically formulated as:

$$\min_x J(x) \quad (9)$$

subject to constraints:

$$g_i(x) \leq 0, \qquad i = 1, 2, \ldots, m$$

$$h_j(x) = 0, \qquad j = 1, 2, \ldots, p$$

where x represents design variables (e.g., shape parameters like Hicks-Henne bump amplitudes or FFD control points); $J(x)$ is the objective function (e.g., minimizing drag coefficient C_D); $g_i(x)$ are inequality constraints (e.g., thickness constraints, volume constraints); $h_j(x)$ are equality constraints (e.g., maintaining a fixed lift coefficient C_L).

SLSQP solves a sequence of Quadratic Programming (QP) subproblems. At each iteration k, it approximates the objective function and constraints using a quadratic model:

$$\min_d \nabla J(x^k)^T d + \frac{1}{2} d^T H d \quad (10)$$

subject to:

$$\nabla g_i(x^k)^T d + g_i(x^k) \leq 0, \qquad i = 1, \ldots, m$$

$$\nabla h_j(x^k)^T d + h_j(x^k) = 0, \qquad j = 1, \ldots, p$$

where $d = x^{k+1} - x^k$ is the step direction; $\nabla J(x^k)$ is the gradient of the objective function at iteration k, computed using the adjoint method in SU2; H is an approximate Hessian matrix, often updated using BFGS quasi-Newton methods; $\nabla g_i(x^k)$ and $\nabla h_j(x^k)$ are the gradients of the constraints [8].

For lift constraint enforcement, the following term is taken into consideration:

$$h(x) = C_L(x) - C_L^* \quad (11)$$

where C_L^* is the target lift coefficient. The gradient of C_L is also computed using the adjoint method:

$$\frac{dh}{dx_i} = \frac{dC_L}{dx_i} = -\lambda \left[\frac{\partial R}{\partial x_i}\right]^T \quad (12)$$

SLSQP ensures that $h(x) = 0$ is satisfied in each design iteration.

Once the quadratic subproblem is solved at each iteration k, the design variables are updated using:

$$x^{k+1} = x^k + \alpha_k d^k$$

where α_k is a step size determined by a line search strategy [8].

2.7. Mesh Deformation

Mesh deformation also plays an essential role in the optimization framework. Once the airfoil geometry is deformed by the optimization algorithm after gradient evaluation, the surrounding volume mesh needs to be deformed. The technique applied in SU2 is to model the mesh element as an elastic solid using the equations of linear elasticity (ELA).

ELA governs the small displacement vector $u(x) = (u, v, w)$ of an elastic solid subject to body forces and surface tractions [9], which can be written as:

$$\nabla \cdot \sigma = f \quad \text{on } \Omega \quad (13)$$

where f is body force, Ω is the computational domain and σ is the stress tensor given by the constitutive relation:

$$\sigma = \lambda Tr(\epsilon) I + 2\mu\epsilon \quad (14)$$

where ϵ is the strain tensor and Tr is its trace. λ and μ are the Lamé constants, depending on the properties of the elastic material, given as a function of Young's modulus E and Poisson's ratio ν:

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \quad (15)$$

$$\mu = \frac{E}{2(1+\nu)} \quad (16)$$

Young’s modulus E indicates the stiffness of the material, where large E means rigidity. Poisson’s ratio measures how much the material shrinks in the lateral direction as it extends in the axial direction ($-1 < \nu < \frac{1}{2}$ for physical materials).

To quantify the deformation of a material fiber in an elastic body, the equation of the linear kinematic law is applied:

$$\epsilon = \frac{1}{2}(\nabla u + \nabla u^T) \qquad (17)$$

Then, the system is completed by applying Dirichlet boundary conditions, $u = g$ on $\partial\Omega$.

This set of equations is discretized using a Galerkin method based on the trial and test spaces. Since a complete explanation of this approach exceeds the article's purpose, the reader should refer to [10]**.**

To achieve robustness and accuracy when large displacements, K. Stein, Tayfun Tezduyar, and R. Benney have applied it with an elastic stiffness varying in inverse proportion to the cell volume, aiming to maintain quality mesh near the bodies (it could be boundary layers or high-resolution zones) [11]. To set up for E and ν, SU2 has several options, including the option called *inverse volume,* which is similar to the study [11].

By using the inverse volume method, SU2 ensures mesh quality and accuracy when deforming the mesh. This approach effectively maintains mesh quality and enables reliable simulations.

3. Optimization Model

In this paper, two shape parameterization methods and the SLSQP algorithm are applied to solve an important optimization problem in aerospace engineering, which is drag minimization for an airfoil (2D problem) or an aircraft wing (3D problem). Drag reduction is important in transonic flight because, in this flight condition, normal shock waves are formed on the surface of aircraft wings. See Fig. 4 to see how the pressure distribution around the NACA 0012 airfoil (basic profile) is disrupted in transonic flight conditions.

This phenomenon is also known as the term “wave drag” in aerodynamics, which means this normal shock waves cause a loss of total pressure, resulting in an increment in drag. In this scenario, the purpose of the optimization algorithm is to minimize this drag caused by shock waves by deforming the basic airfoil geometry and, as a result, terminating shock waves formed on the airfoil surface.

This paper focuses on studying the effect of different shape parameterization techniques on two-dimensional aerodynamic shape optimization. Initial freestream conditions, including Mach number, pressure, temperature, and angle of attack, are given in Table 1. The lift coefficient (C_L) obtained from the initial CFD simulation of the NACA 0012 airfoil is fixed in the optimization process. The thickness of the airfoil and pitching moment (M_Z) derived from this solution are also constrained.

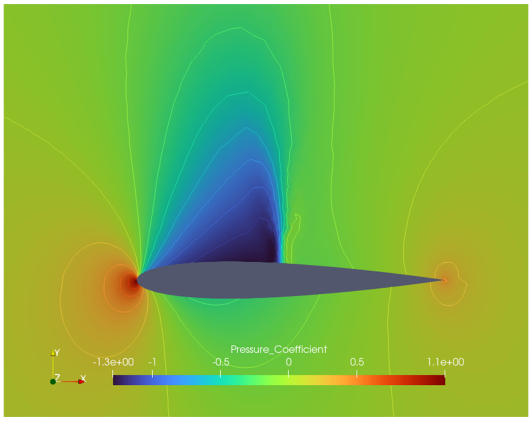

Fig. 4. Pressure contours around the basic profile

Table 1. Initial flow conditions for the design loop

Condition	Value
Mach number	0.75
Pressure (Pa)	101325
Temperature (K)	288.15
Angle of attack (degree)	2.00

During the design loop, the angle of attack of the airfoil varied due to shape deformation. This was done to ensure that the lift coefficient (obtained from the preliminary solution) remained the same throughout the design iterations. The results, i.e., the airfoil shape design such that the drag value is minimized, and the lift coefficient that was constrained, depended on the initial angle of attack.

This work performs airfoil optimization with many design variables (DVs) varying from 20 to 80 in order to see how it affects the optimization process. Optimal shape design cases use two shape parameterization techniques, including Hicks-Henne bump functions (HHBF) and FFD.

With the HHBF method, the bump position is set at 0.5 along the *Ox* axis, meaning it is located at the midpoint of the airfoil's chord. With the FFD method, a user-defined box of FFD control points is created surrounding the airfoil. This FFD box is then divided with control points based on the number of DVs. In both cases, the optimization is performed with an increasing number of DVs, specifically 20, 40, 60, and 80, resulting in deformed airfoil shapes that meet the requirements of the optimization problem.

4. Results and Comparison

The optimization of the NACA 0012 airfoil using both the FFD and Hicks-Henne Bump Function (HHBF) parameterization techniques yielded significant reductions in drag coefficient (C_D), with both methods demonstrating effectiveness in different aspects. Fig. 5 illustrates the comparison of each parameterization technique's performance on the drag reduction process.

It appears that the HHBF technique demonstrates better performance in terms of design iteration efficiency compared to that of FFD. Specifically, the HHBF technique appears to achieve comparable levels of drag coefficient (C_D) reduction with a lower number of design iterations in most cases. Especially, with 20 DVs, the HHBF method required only 20 iterations to reach a minimum C_D of 0.001123, whereas the FFD method needed 101 iterations to achieve a slightly higher minimum C_D of 0.001204.

However, the C_D graph of FFD shows stability and a smooth decrease, while HHBF exhibits greater oscillation during the optimization process. This indicates that FFD finds the optimal solution much more steadily and consistently than HHBF.

Fig. 6 illustrates the pressure field of optimal solutions using HHBF and FFD techniques. Subfigures (a), (b), (c), and (d) are solutions obtained using the HHBF technique with 20, 40, 60, and 80 DVs, respectively. Similarly, solutions using the FFD technique with the same order of DV numbers are shown in subfigures (e), (f), (g), and (h).

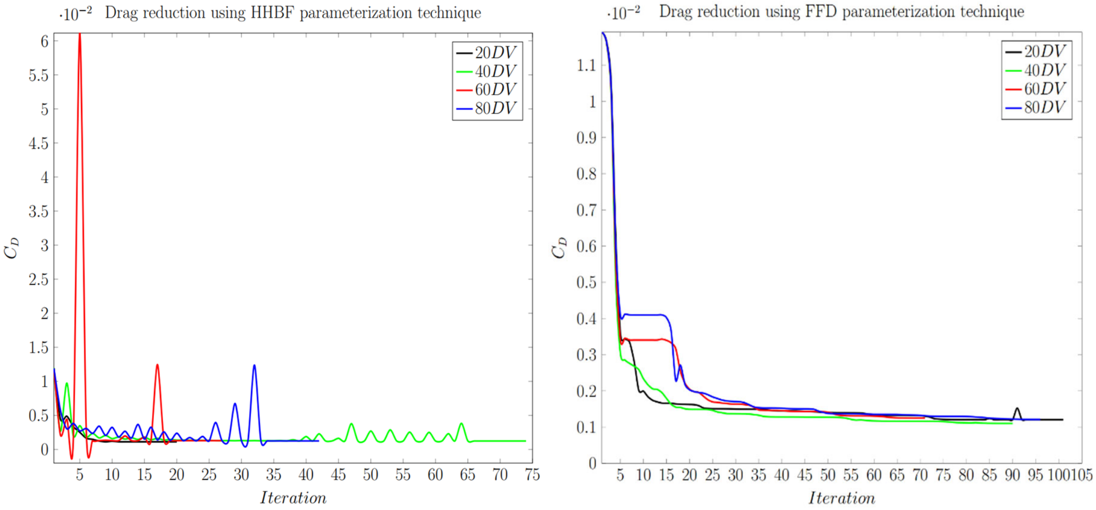

Fig. 5. Comparison of drag reduction using two different shape parameterization techniques

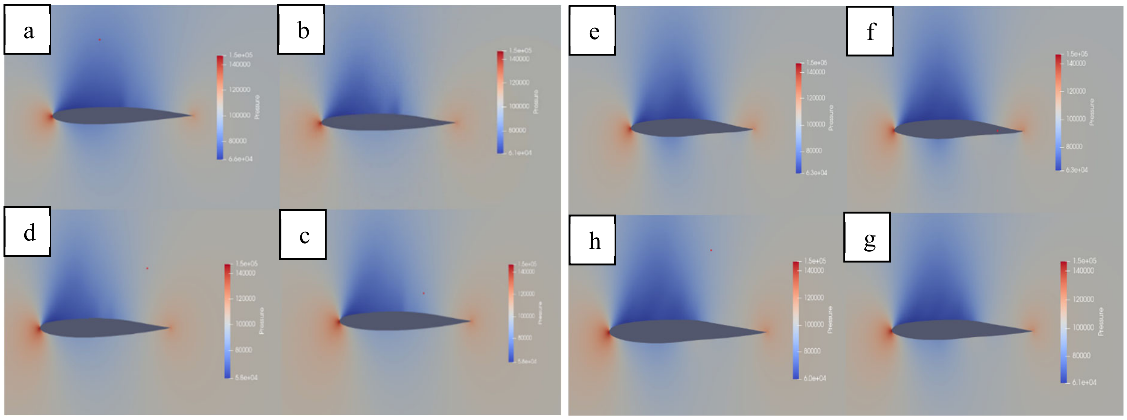

Fig. 6. Pressure fields around each optimal airfoil solutions by HHBF (left) and FFD (right)

The HHBF method yielded smoother and more regular airfoil profiles, suggesting a potential benefit in terms of manufacturing feasibility and aerodynamic performance. Fig. 7 reveals distinct differences in the optimized airfoil profiles produced by the two parameterization techniques. The FFD method appears to produce a more "kinked" or non-smooth airfoil profile compared to the HHBF method. Conversely, the HHBF method results in a smoother and more regular airfoil profile, due to its bump functions, resulting in a more consistently smooth airfoil surface. View Fig. 8 to see the evolution of the original airfoil throughout the shape C_D optimization process.

Most importantly, optimal solutions, including minimum value and decrement of C_D concerning parameterization techniques and number of DVs, are shown in Table 2. Both techniques have demonstrated significant reductions in drag coefficient for the NACA 0012 airfoil through a profile optimization process. On average, an impressive 89.85% reduction in drag was achieved across all cases. Notably, the HHBF method exhibited a distinct advantage in convergence efficiency, requiring fewer design iterations to achieve comparable drag reduction levels.

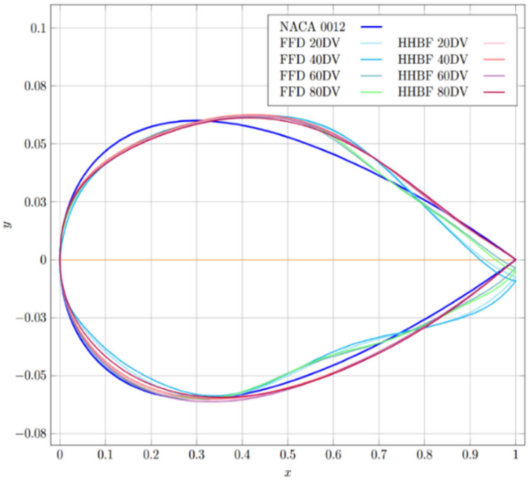

Fig. 7. Basic profile (NACA 0012) and optimized profiles by FFD/HHBF methods

Table 2. Drag coefficients evaluation of optimal solutions by FFD and HHBF method

DVs	20		40		60		80	
Method	FFD	HHBF	FFD	HHBF	FFD	HHBF	FFD	HHBF
Initial	0.0119129							
Minimum	0.001204	0.001123	0.001103	0.001244	0.001251	0.001289	0.001212	0.001252
Reduction	0.010709	0.010790	0.010810	0.010669	0.010662	0.010624	0.010701	0.010661
Reduction (%)	89.8	90.5	90.7	89.5	89.4	89.1	89.8	89.4
Design iterations	101	20	90	59	71	27	96	42

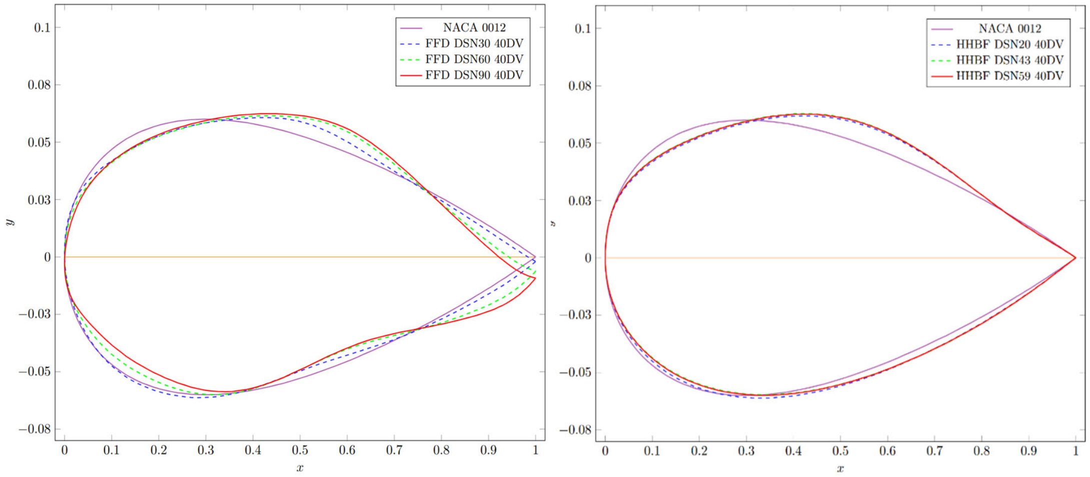

Fig. 8. Airfoil shape deformation during the optimization process by HHBF compared to FFD

5. Conclusion

This work focused on investigating the effects of shape parameterization techniques on numerical shape optimization of a two-dimensional airfoil. In this work, the open-source SU2 code was applied to perform gradient-based aerodynamic shape optimization using the discrete adjoint method. Two shape parameterization techniques were tested in the same transonic flight conditions, including Hicks-Henne Bump Functions and FFD.

Several conclusions can be derived from this study. In terms of drag reduction, Hicks–Henne bump functions and Free–form Deformation were shown to be equally effective as geometry parameterization methods for both optimization problems. In terms of design iterations, it appears that the HHBF technique demonstrates better performance in terms of design iteration efficiency compared to the FFD method. However, when it comes to stability, FFD shows stability and a smoother decrease in drag values, while HHBF exhibits greater unsteadiness during the optimization process.

References

[1] R. M. Hicks and P. A. Henne, Wing design by numerical optimization, Journal of Aircraft, vol. 15, iss. 7, pp. 407-412, Jul. 1978, https://doi.org/10.2514/3.58379

[2] T. W. Sederberg and S. R. Parry, Free-form deformation of solid geometric models, ACM SIGGRAPH Computer Graphics, vol. 20, iss. 4, pp. 151-160, 1986, https://doi.org/10.1145/15886.15903

[3] D. Masters, N. J. Taylor, T. Rendall, C. Allen, and D. Poole, A geometric comparison of aerofoil shape parameterisation methods, AIAA Journal, Jan. 2016, https://doi.org/10.2514/6.2016-0558

[4] G. Yang and A. D. Ronch, Aerodynamic shape optimisation of benchmark problems using SU2, in 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Kissimmee, 8-12 Jan. 2018, Florida, USA, 2018, https://doi.org/10.2514/6.2018-0412

[5] T. Albring, M. Segebaum, and N. R. Gauger, Efficient aerodynamic design using the discrete adjoint method in SU2, in 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Washington D.C, 13-17 Jun. 2015, https://doi.org/10.2514/6.2016-3518

[6] F. Palacios, M. R. Colonno, A. C. Aranake, A. Campos, S. R. Copeland, T. D. Economon, A. K. Lonkar, T. W. Lukaczyk, T. W. R. Taylor, and J. J. Alonso, Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design, , in 51st AIAA Aerospace Sciences Meeting Conference, including the New Horizons Forum and Aerospace Exposition, Grapevine (Dallas/Ft. Worth Region), Texas, January 2013, [Online] Available at: https://arc.aiaa.org/doi/10.2514/6.2013-287

[7] SU2 – Multiphysics Simulation and Design Software, Governing equations in SU2. (2023, Jun. 23). [Online]. Available: https://su2code.github.io/docs_v7/Theory/

[8] D. Kraft, A software package for sequential quadratic programming, Tech. Rep. DFVLR-FB 88-28, DLR German Aerospace Center – Institute for Flight Mechanics, Koln, Germany, 1988.

[9] A. J. Joshy and J. T. Hwang, PySLSQP: a transparent Python package for the SLSQP optimization algorithm modernized with utilities for visualization and post-processing, The Journal of Open-Source Software, vol. 9, no. 103, Nov. 2024, https://doi.org/10.21105/joss.07246

[10] R. P. Dwight, Robust mesh deformation using the linear elasticity equations, in Computational Fluid Dynamics 2006, Springer, pp. 401-406, Jan. 2009, https://doi.org/10.1007/978-3-540-92779-2_62

[11] K. Stein, T. Tezduyar, and R. Benney, Mesh moving techniques for fluid-structure interactions with large displacements, Journal of Applied Mechanics, vol. 70, pp. 58-63, Jan. 2003, https://doi.org/10.1115/1.1530635