

## Research Article

# Aerodynamic Optimal Shape Design Based on Body-Fitted Grid Generation

**Farzad Mohebbi and Mathieu Sellier**

*Department of Mechanical Engineering, The University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand*

Correspondence should be addressed to Farzad Mohebbi; [farzadmohebbi@yahoo.com](mailto:farzadmohebbi@yahoo.com)

Received 9 April 2014; Accepted 11 June 2014; Published 27 August 2014

Academic Editor: Caner Özdemir

Copyright © 2014 F. Mohebbi and M. Sellier. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper is concerned with an optimal shape design problem in aerodynamics. The inverse problem in question consists in finding the optimal shape an airfoil placed in a potential flow at a given angle of attack should have such that the pressure distribution on its surface matches a desired one. The numerical method to achieve this aim is based on a body-fitted grid generation technique (elliptic, O-type) to generate a mesh over the airfoil surface and solve for the flow equation. The O-type scheme is used due to its ability to generate a high quality (fine and orthogonal) grid around the airfoil surface. This paper describes a novel and very efficient sensitivity analysis scheme to compute the sensitivity of the pressure distribution to variation of grid node positions and both the conjugate gradient method (CGM) and a version of the quasi-Newton method (i.e., BFGS) are used as optimization algorithms to minimize the difference between the computed pressure distribution on the airfoil surface and desired one. The elliptic grid generation technique allows us to map the physical domain (body) onto a fixed computational domain and to discretize the flow equation using the finite difference method (FDM).

## 1. Introduction

Thanks to the advent of modern high speed computers over the last few decades, computational fluid dynamics (CFD) has been extensively employed as an analysis and as a design optimization tool. Among the methodologies often employed in shape optimization are gradient-based techniques. These techniques may be applied to minimize a specified objective function. In airfoil shape optimization, the objective function can be, for example, a measure of difference between the pressure distribution on the airfoil surface and a desired one, and it would be desirable to minimize this objective function. In this paper, we consider the 2D shape optimization of an airfoil in an irrotational and incompressible flow governed by the Laplace equation. The procedure employed is based on the elliptic grid generation, a novel sensitivity analysis (based on finite difference method), and an optimization method. The conjugate gradient method and an efficient version of quasi-Newton method (BFGS) will be used as the optimization algorithms. The airfoil surface is parameterized using the grid points and the Bezier curve. Three different types of design

variables were considered: the grid points, the Bezier curve control points, and the maximum thickness of NACA00xx airfoils. It will be represented that the use of the Bezier curve significantly improves the optimization performance to reach the optimal shape. Furthermore, it will be shown that the proposed sensitivity analysis method reduces the computation cost significantly even for large number of the design variables.

Some of the earliest studies using a combination of CFD with numerical optimization in aerodynamic were made by Hicks et al. [1] and Hicks and Henne [2]. In [1], a procedure for optimal design of symmetric low-drag, nonlifting transonic airfoils in inviscid flow is proposed. The proposed procedure uses an optimization program based on the method of the feasible directions coupled with an analysis program that utilizes a relaxation method to solve the partial differential equation that governs the inviscid, transonic, and small disturbance fluid flow. The drag minimization with geometric constraints is considered in this reference. In fluid dynamics, Pironneau was the first one to use the adjoint equations for design [3]. This is the first application of control

theory to design optimization. However, within the field of aeronautical computational fluid dynamics, Jameson was the first researcher who used the continuous adjoint formulation for aerodynamic shape optimization in transonic potential flows and flows governed by Euler equations [4–7]. Giles et al. made considerable contributions to the development of the discrete adjoint approach [8–11]. In [10], the adjoint equations are formulated for the transonic design applications for which there are shocks. The adjoint equations were already formulated for the incompressible or subsonic flows in which the assumption that the original nonlinear flow solution is smooth is valid. In [11], a number of algorithm developments are presented for adjoint methods using the “discrete” approach. In continuous adjoint method, the original partial differential equations are linearized, the adjoint partial differential equation and appropriate boundary conditions are formulated, and finally the equations are discretized. Unlike the continuous adjoint approach, in discrete adjoint approach the partial differential equations are discretized, the discrete equations are linearized, and then the transpose of the linear operator is used to form the adjoint problem. The adjoint equations have also been used by Baysal and Eleshaky to infer the optimal design for a scramjet-afterbody configuration which yields the maximum axial thrust [12] and by Taasan et al. to obtain an optimal airfoil shape [13]. Baysal and Eleshaky’s work was based on a computational fluid dynamics-sensitivity analysis algorithm (two different quasi-analytical approaches: the direct method and the adjoint variable method) to solve Euler equations for the inviscid analysis of the flow. Adjoint methods have been applied to incompressible viscous flow problems by Cabuk et al. [14] and Desai and Ito [15]. Cabuk et al. worked on the problem of determining the profile of a channel or duct that provides the maximum static pressure rise by solving the incompressible, laminar flow governed by the steady state Navier-Stokes equations. Early applications of discrete adjoint methods on unstructured meshes can be found in works by Elliott and Peraire in inviscid [16] and viscous flows [17] for 2D and 3D [18] configurations. In [16], an inverse design procedure for single- and multielement airfoils using unstructured grids and based on the Euler equations is presented. The discrete adjoint method is used to compute the sensitivities and the results are compared with corresponding finite difference values. It is shown that the use of the adjoint method practically eliminates the dependence of the objective function gradient computation on the number of design variables. The continuous adjoint approach for unstructured grids has been developed by Anderson and Venkatakrishnan [19]. In [19], aerodynamic shape optimization on unstructured grids using a continuous adjoint approach is developed and analyzed for inviscid and viscous flows. B spline and Bezier curves are employed to parameterize the airfoil surface. The objective functions considered include drag minimization, lift maximization, and matching a specified pressure distribution. The quasi-Newton optimization method is used to obtain the optimal design. Evolutionary algorithms, as methods that do not need the computation of the gradient, have recently gained much attention in the context of aerodynamic shape optimization [20–24]. Although they are of extremely high

computational cost, they have the advantage that they can escape from a “local minimum” (a major issue in using gradient based methods) and have the ability of finding globally optimum solutions amongst many local optima [25, 26]. A detailed study of many methods in shape optimization in fluid mechanics is given by Mohammadi and Pironneau [27].

The adjoint approach, as an alternative to the finite difference method to compute the gradient of functional with respect to the design variables, is computationally very efficient. Therefore, as far as the computational cost is concerned, it is the appropriate choice. This is the case when there are a large number of design variables which makes use of the finite difference method impractical. The differences between the adjoint method and the finite difference one (to compute the gradient of functional with respect to the design variables) can be summarized as follows.

*Adjoint Method.*  $N$  design variables, 1 flow solution, and 1 adjoint calculation.

*Finite Difference Method.*  $N$  design variables,  $N$  flow solutions. Because

$$\frac{\partial \mathcal{F}}{\partial \alpha_j} = \frac{[\mathcal{F}(\alpha_j + \delta \alpha_j) - \mathcal{F}(\alpha_j)]}{\delta \alpha_j}, \quad (1)$$

where  $\mathcal{F}$  is the objective function and  $\alpha_j$  are the design variables [28]. As can be seen, aerodynamic shape optimization with large number of design variables is computationally practical only when the adjoint method is used. However, as will be shown in this paper, a novel sensitivity analysis will be presented which makes use of the finite difference method comparable (from computation cost viewpoint) to the use of adjoint method. The numerical algorithm used in this paper is already employed in shape optimization problems in heat conduction [29, 30]. The numerical algorithm consists of three steps, namely, grid generation and flow equation solver to find the pressure on the airfoil surface, sensitivity analysis to compute the gradient of the objective function with respect to the design variables, and an optimization method to minimize the functional and reach optimum solution.

## 2. Governing Equation

For a two-dimensional incompressible flow, a stream function  $\psi$  can be defined such that

$$\begin{aligned} u &= \frac{\partial \psi}{\partial y}, \\ v &= -\frac{\partial \psi}{\partial x}, \end{aligned} \quad (2)$$

where  $u$  and  $v$  are the components of the velocity vector  $\mathbf{V}$ ; that is,  $\mathbf{V} = u\mathbf{i} + v\mathbf{j}$  ( $\mathbf{i}$  and  $\mathbf{j}$  are the unit vectors in  $x$  and  $y$  directions, resp.). Combining the above definitions with the irrotationality condition leads to the following Laplace equation for the stream function

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0. \quad (3)$$

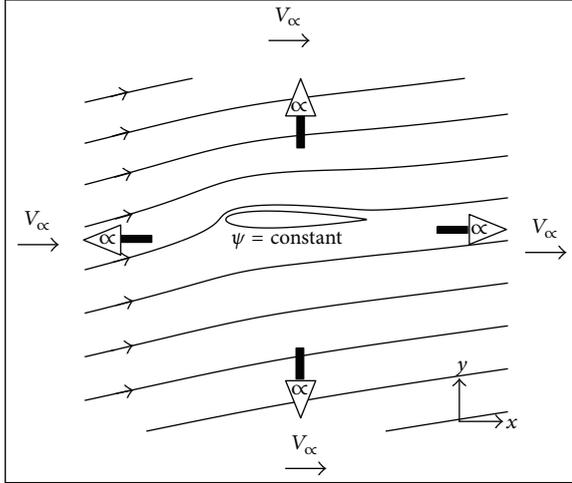


FIGURE 1: Boundary conditions at infinity and on the airfoil surface (no-penetration).

Consider an irrotational incompressible flow over an airfoil (Figure 1). The boundary conditions are as shown in Figure 1.

*Conditions at Infinity.* Far away from the airfoil surface (toward infinity), in all directions, the flow approaches the uniform freestream conditions. If the angle of attack (AOA) is  $\alpha$ , the free stream velocity  $V_\infty$ , the components of the flow velocity can be written as

$$\begin{aligned} u &= \frac{\partial \psi}{\partial y} = V_\infty \cos \alpha, \\ v &= -\frac{\partial \psi}{\partial x} = V_\infty \sin \alpha. \end{aligned} \quad (4)$$

*Condition on the Airfoil Surface.* The relevant boundary condition at the airfoil surface for this inviscid flow is the no-penetration boundary condition. Thus the velocity vector must be *tangent* to the surface. This wall boundary condition can be expressed by

$$\frac{\partial \psi}{\partial s} = 0, \quad \text{or } \psi = \text{constant}, \quad (5)$$

where  $s$  is tangent to the surface. In the problem of the flow over an airfoil, if the free stream velocity and the angle of attack are known, from the boundary conditions at infinity (see (4)) and the wall boundary condition (see (5)) one can compute the stream function  $\psi$  at any point of the physical domain (flow region). Then, by knowing  $\psi$ , one can compute the velocity of all points. Since for an incompressible flow, the pressure coefficient is a function of the velocity only, one can obtain the pressure of any point in the flow region, as will be shown.

*Pressure Coefficient.* The pressure coefficient  $C_p$  is defined as

$$C_p = \frac{p - p_\infty}{(1/2) \rho_\infty V_\infty^2} = 1 - \left( \frac{V}{V_\infty} \right)^2, \quad (6)$$

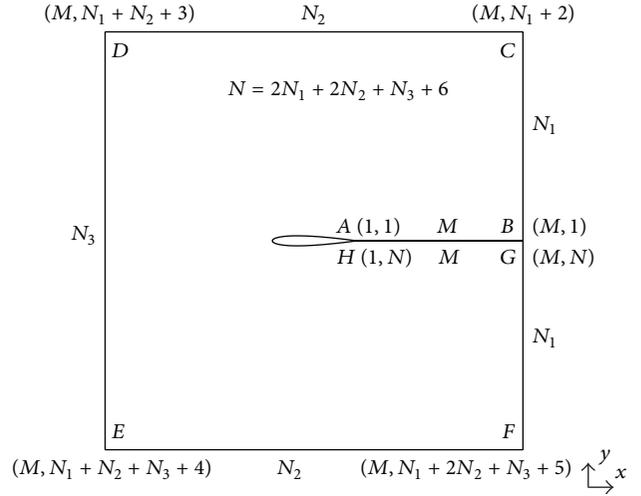


FIGURE 2: Physical domain showing the discretization of the boundaries used for O-type elliptic grid generation technique.

where  $V$  is the velocity of fluid at the point at which the pressure coefficient  $C_p$  is being evaluated. At standard sea level conditions,

$$\begin{aligned} \rho_\infty &= 1.23 \text{ kg/m}^3, \\ p_\infty &= 1.01 \times 10^5 \text{ N/m}^2, \end{aligned} \quad (7)$$

where  $\rho_\infty$  and  $p_\infty$  are the freestream density and pressure, respectively. From (6),

$C_p = 0$  indicates that the point at which the pressure coefficient  $C_p$  is being evaluated is located at infinity.

$C_p = 1$  indicates that the point at which the pressure coefficient  $C_p$  is being evaluated is a stagnation point (where  $V = 0$ ). For an incompressible flow, this is the maximum allowable value of  $C_p$  anywhere in the flow field.

And in regions of the flow where  $V > V_\infty$ ,  $C_p$  value will be negative.

### 3. Grid Generation and Flow Solver

To calculate the pressure at any point in the flow region, a grid should be generated over the region. The grid generation method considered in this study is the elliptic grid generation, which was proposed by Thompson et al. [31] and is based on solving a system of elliptic partial differential equations to distribute nodes in the interior of the physical domain by mapping the irregular physical domain from the  $x$  and  $y$  physical plane (Figure 2) onto the  $\xi$  and  $\eta$  computational plane (Figure 3), which is a regular region.

The O-type elliptic grid generation technique is employed here which results in a smooth and orthogonal grid over the airfoil surface. The discretization of the physical domain (flow region) and the corresponding computational domain are shown in Figures 2 and 3, respectively. In the computational domain,  $M$  and  $N = 2N_1 + 2N_2 + N_3 + 6$  are the number of nodes in the  $\xi$  and  $\eta$  directions, respectively. The resulting O-type grid scheme over an airfoil for the case  $N_2 = N_1$  and  $N_3 = 2N_1 - 1$  or  $N = 6N_1 + 5$  is shown in Figure 4.

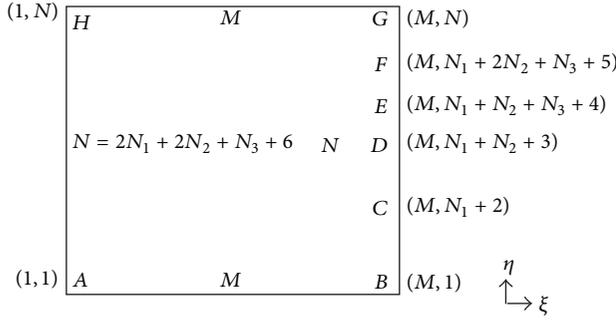


FIGURE 3: Computational domain showing the discretization of the physical domain boundaries.

The initial guess for the elliptic grid generation is performed using the transfinite interpolation (TFI) method. Since TFI method is an algebraic technique and does not require much computational time, it will be an appropriate initial guess for the elliptic grid generation method and accelerates convergence time for the elliptic method. Another advantage of using the TFI method as an initial guess is that it prevents the grids generated by the elliptic (O-type) method from folding.

If  $V_\infty$  and  $\alpha$  are known, then from (4) one can obtain the stream function  $\psi$  at any point on the boundaries of the physical domain as follows:

$$\psi_2 = \psi_1 + (y_2 - y_1) V_\infty \cos \alpha, \quad (8)$$

$$\psi_2 = \psi_1 - (x_2 - x_1) V_\infty \sin \alpha, \quad (9)$$

where subscripts 1 and 2 refer to any two arbitrary grid points on the boundaries of the physical domain. Equations (8) and (9) are applied to vertical and horizontal boundaries of the physical domain, respectively. By knowing the values of the stream function  $\psi$  on the boundaries of the physical domain as well as on the airfoil surface, we can obtain the values of  $\psi$  over the physical domain by applying the Kutta condition [32, 33] and using the following formula (by mapping the physical domain onto the computational domain [29]):

$$\alpha \psi_{\xi\xi} - 2\beta \psi_{\xi\eta} + \gamma \psi_{\eta\eta} = -J^2 (P(\xi, \eta) \psi_\xi + Q(\xi, \eta) \psi_\eta), \quad (10)$$

where

$$\begin{aligned} \alpha &= x_\eta^2 + y_\eta^2, \\ \beta &= x_\xi x_\eta + y_\xi y_\eta, \\ \gamma &= x_\xi^2 + y_\xi^2, \\ J &= x_\xi y_\eta - x_\eta y_\xi, \quad (\text{Jacobian of transformation}). \end{aligned} \quad (11)$$

$P$  and  $Q$  are grid control functions which control the density of grids towards a specified coordinate line or about a specific

grid point. Equations (10) and (11) are discretized using the finite difference method. For more details, please refer to [29].

*Velocity Calculation.* There are three sections where the velocity must be known:

- (1) the outer boundaries (four sides CD, DE, EF, and FC of the rectangle shown in Figure 2);
- (2) the airfoil surface (AH in Figure 2);
- (3) the inside of the physical domain.

The velocity values on the outer boundaries are known from the conditions at infinity (using (4)). In other words,  $x$ -component of the velocity vector ( $u$ ) on all the outer boundaries is equal to  $V_\infty \cos \alpha$  and  $y$ -component of the velocity vector ( $v$ ) on all the outer boundaries is equal to  $V_\infty \sin \alpha$ . For the inside of the physical domain and the airfoil surface, we can use the flowing relationships to evaluate the velocity. These relationships are obtained by using the transformation relationships and chain rule in mapping the physical domain onto the computational one. Consider

$$u_{i,j} = \frac{\partial \psi}{\partial y} \Big|_{i,j} = \frac{1}{J} \left[ -(x_\eta)_{i,j} (\psi_\xi)_{i,j} + (x_\xi)_{i,j} (\psi_\eta)_{i,j} \right], \quad (12)$$

$$v_{i,j} = -\frac{\partial \psi}{\partial x} \Big|_{i,j} = -\frac{1}{J} \left[ (y_\eta)_{i,j} (\psi_\xi)_{i,j} - (y_\xi)_{i,j} (\psi_\eta)_{i,j} \right]. \quad (13)$$

The central and forward difference schemes are used for the inside of the physical domain and the airfoil surface, respectively. After obtaining the components of the velocity vector, the total velocity (velocity distribution) can be computed by

$$V_{i,j} = \sqrt{u_{i,j}^2 + v_{i,j}^2}. \quad (14)$$

As stated before, for an incompressible flow, the pressure coefficient can be expressed in terms of the velocity only. Thus (6) can be used to determine the pressure at any grid point in the domain. Therefore,

$$P_{i,j} = \frac{1}{2} \rho (V_\infty^2 - V_{i,j}^2) + p_\infty. \quad (15)$$

*Validation of the Results for the Pressure Distribution.* The results obtained here are compared with the results given in [34] which are obtained both analytically and by using the panel method (see Figure 7).

*Validation Case.* The pressure coefficient distribution ( $C_p$ ) over the NACA 0012 airfoil at an angle of attack  $\alpha = 9^\circ$  is plotted. The results are compared with the results from [34]. The O-type grid size used in the computation is  $155 \times 155$ . The computation time is 53 seconds.

#### 4. Airfoil Parameterization

So far, the airfoil surface is parameterized by grid points which result in accurate pressure distribution on the airfoil

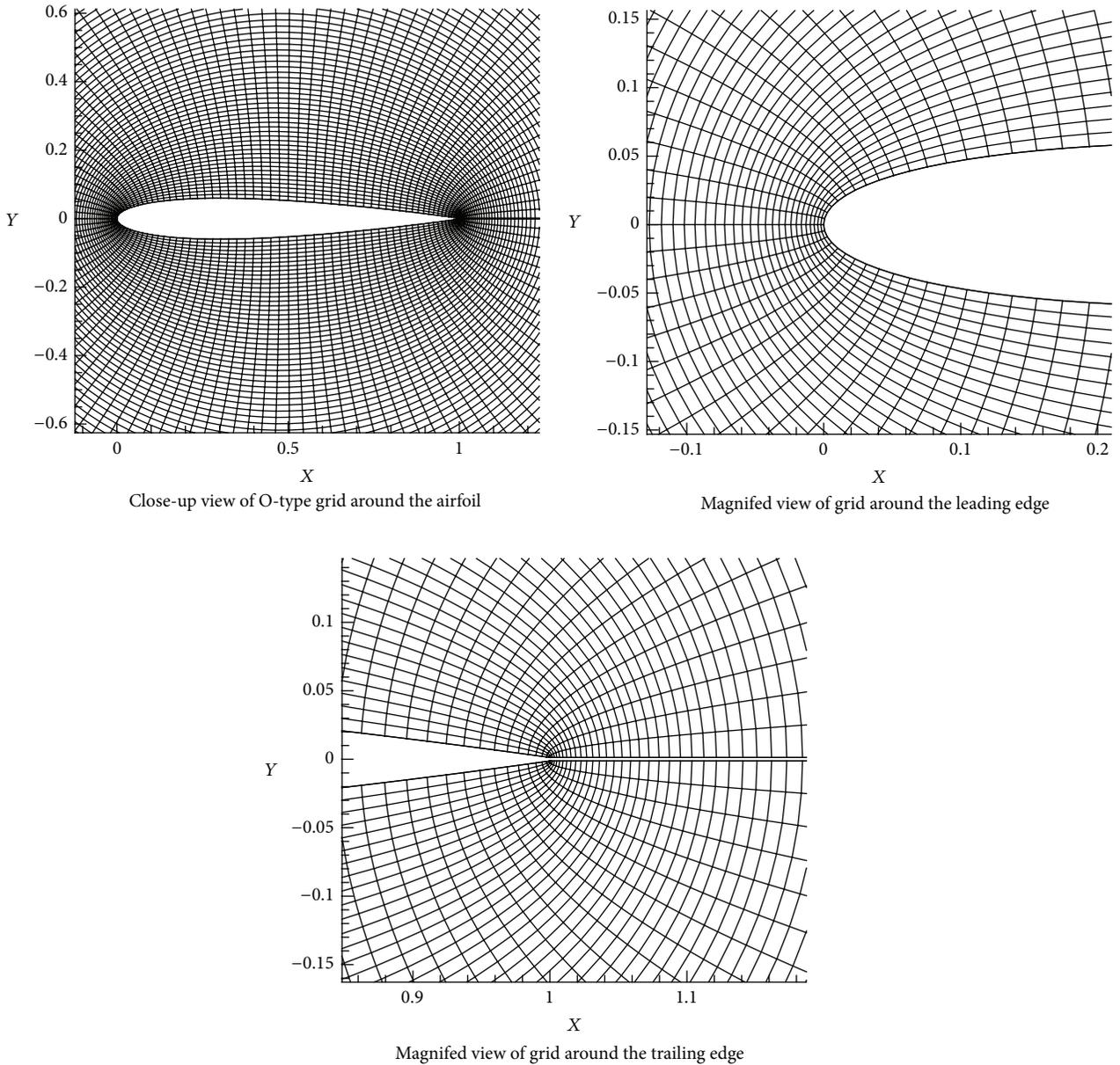


FIGURE 4: O-type grid (elliptic) around an airfoil. This close-up view of the grid shows orthogonality and smoothness of the gridlines especially near airfoil surface.

surface (see Figures 14, 19, and 25). However, a large number of grid points are needed to obtain such accurate results which in turn lead to high (see Figures 5 and 6) computation cost. The design variables are the coordinate (usually  $y$ -coordinate) of grid points. Therefore, the optimization process may be inappropriate if there are a large number of design variables since it is difficult to maintain a smooth geometry, the optimization problem will be difficult to solve, and the optimization strategy is likely to fail or be impractical [35]. Thus alternative methods of airfoil surface parameterization are needed. These methods should represent great flexibility in defining the airfoil surface with minimum design variables. In this paper, in addition to the grid points to represent the airfoil surface, Bezier curves (a special subset of B-spline) are

employed due to their ability to produce airfoil surfaces easily and precisely with only a few control points.

*Bezier Curve.* A Bezier curve is a special case of a B-spline curve and is mathematically defined by

$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t), \quad (16)$$

where

$$J_{n,i}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} \quad (17)$$

is Bernstein basis polynomial of degree  $n$ . By convention  $0^0 \equiv 1$  and  $0! \equiv 1$ . Here,  $n$ , the degree of the Bernstein

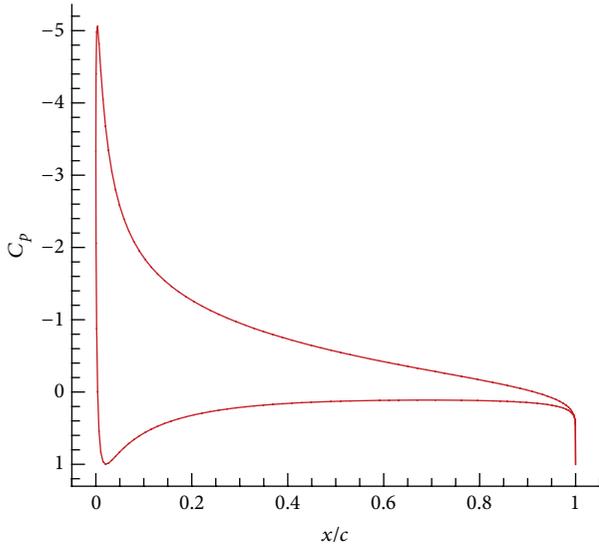


FIGURE 5: The pressure coefficient distribution over an NACA 0012 airfoil at a  $9^\circ$  angle of attack obtained numerically.

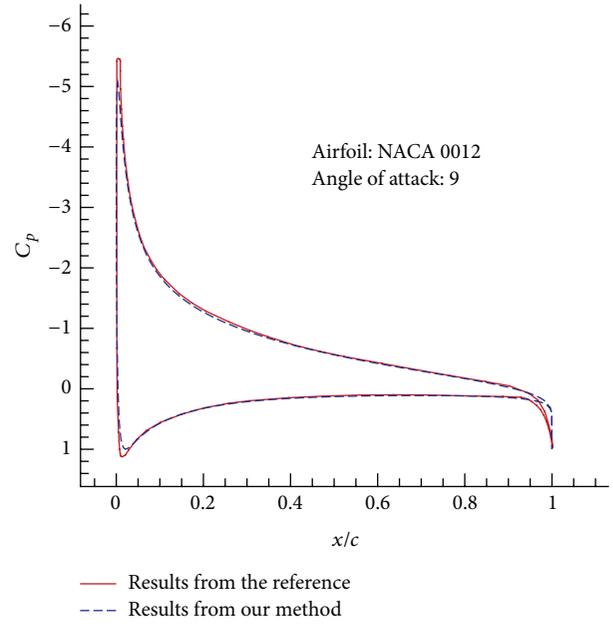


FIGURE 7: Comparison between the results from [34] and the results from our method for validation case. The figure shows an excellent agreement between the results.

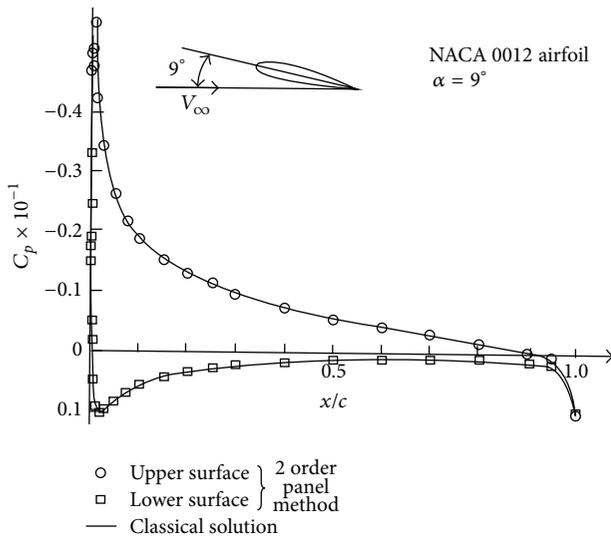


FIGURE 6: The pressure coefficient distribution over an NACA 0012 airfoil at a  $9^\circ$  angle of attack [34].

basis polynomial is one less than the number of points in the Bezier polygon. In other words, the number of control points is  $n + 1$ . The points  $B_i$  are the vertices of a Bezier polygon or the control points of a Bezier curve. The curve begins at  $B_0$  and ends at  $B_n$ . The order of a Bezier curve  $k$  is equal to  $n + 1$ . In other words, the order of a Bezier curve is equal to the number of the control points [36].

In this paper, two different Bezier curves of order 7 (degree = 6) and of order 11 (degree = 10) will be considered. As it will be shown, the Bezier curve of order 7 represents the better optimization performance due to its less design variables. However, this kind of Bezier curve is not able to produce very accurate airfoil shapes. Indeed, it is appropriate to NACA 00xx airfoils only. On the other hand, the Bezier

curve of order 11 can successfully generate any airfoil shape with a high degree of accuracy. Therefore, the formulation for the Bezier curve of order 11 only will be given here. The formulation for the Bezier curve of order 7 can be written in a similar fashion.

The parametric Bezier curve of order 11 is as follows:

$$n = 10 \implies \text{number of control points} = 11$$

$$\begin{aligned} P(t) &= \sum_{i=0}^{10} B_i J_{10,i}(t) \\ &= B_0 J_{10,0}(t) + \dots + B_{10} J_{10,10}(t) \\ &= B_0 \frac{10!}{0!(10-0)!} t^0 (1-t)^{10-0} \\ &\quad + \dots + B_{10} \frac{10!}{10!(10-10)!} t^{10} (1-t)^{10-10}. \end{aligned} \quad (18)$$

Therefore,

$$\begin{aligned} P(t) &= B_0 (1-t)^{10} + B_1 10t(1-t)^9 + B_2 45t^2(1-t)^8 \\ &\quad + B_3 120t^3(1-t)^7 + B_4 210t^4(1-t)^6 \\ &\quad + B_5 252t^5(1-t)^5 + B_6 210t^6(1-t)^4 \\ &\quad + B_7 120t^7(1-t)^3 + B_8 45t^8(1-t)^2 \\ &\quad + B_9 10t^9(1-t)^1 + B_{10} t^{10}. \end{aligned} \quad (19)$$

In order to construct the airfoil surface, two Bezier curve will be considered corresponding to the upper and lower surfaces,

respectively. Here there are 11 control points (vertices) for each surface. Since the coordinates of the airfoil surface are known, the problem is to determine values for the control points  $B_i$  ( $i = 0, \dots, 10$ ). In other words, our problem is to specify the coordinates of the control points  $B_i$  so that the curve passes through the predetermined data points on the airfoil surface. Equation (16) can be written in matrix form as follows:

$$[P(t)] = [J(t)] [B]. \quad (20)$$

If the number of the chosen data points on the airfoil surface is  $m$  and the degree of Bezier curve is  $n$ , then  $[P(t)]$  is a  $m \times 2$  matrix,  $[J(t)]$  is a  $m \times (n + 1)$  matrix, and  $[B]$  is a  $(n + 1) \times 2$  matrix. Two columns of the matrix  $[P(t)]$  pertain to the  $x$ - and  $y$ -coordinates of the predetermined data on the airfoil surface. Equation (20) can be rewritten as

$$[P(t)]_{m \times 2} = [J(t)]_{m \times (n+1)} [B]_{(n+1) \times 2}. \quad (21)$$

If  $m = n + 1$ , the matrix  $[J(t)]_{m \times (n+1)}$  will be a square matrix and it can be inverted. In such a case, (21) can be written as follows to find the matrix  $[B]$ :

$$[B]_{(n+1) \times 2} = [J(t)]_{m \times (n+1)}^{-1} [P(t)]_{m \times 2}. \quad (22)$$

However, the number of the airfoil surface data points is usually more than the number of control points. In such a case, there are more equations than unknowns and the matrix  $[J(t)]_{m \times (n+1)}$  is no longer a square matrix. Hence it is required to convert it to a square matrix by multiplying both sides of (21) by the transpose of  $[J(t)]_{m \times (n+1)}$  as follows:

$$\begin{aligned} [J(t)]_{(n+1) \times m}^T [P(t)]_{m \times 2} \\ = [J(t)]_{(n+1) \times m}^T [J(t)]_{m \times (n+1)} [B]_{(n+1) \times 2}. \end{aligned} \quad (23)$$

Thus,

$$\begin{aligned} [B]_{(n+1) \times 2} = & \left[ [J(t)]_{(n+1) \times m}^T [J(t)]_{m \times (n+1)} \right]^{-1} \\ & \cdot [J(t)]_{(n+1) \times m}^T [P(t)]_{m \times 2}. \end{aligned} \quad (24)$$

NACA 0015 and TsAGI "B" 12% airfoils produced by Bezier curve with  $n = 10$  and  $m = 51$  and their comparison with conventional NACA 0015 and TsAGI "B" 12% airfoils are shown in Figures 8 and 9, respectively. There is an excellent agreement between two airfoils in each figure.

The predetermined data for the NACA airfoils can be extracted from, for example, the software JavaFoil [37] which is based on the analytical NACA formulations.

*NACA 00xx Symmetric Airfoils.* Since the maximum thickness of a NACA 00xx symmetric airfoil will be considered as a design variable, the equation for generating such airfoils is given as follows:

$$\begin{aligned} \pm y_t = \frac{t}{0.2} \left[ 0.2969 \sqrt{x} - 0.1260x \right. \\ \left. - 0.3516x^2 + 0.2843x^3 - 0.1015x^4 \right], \end{aligned} \quad (25)$$

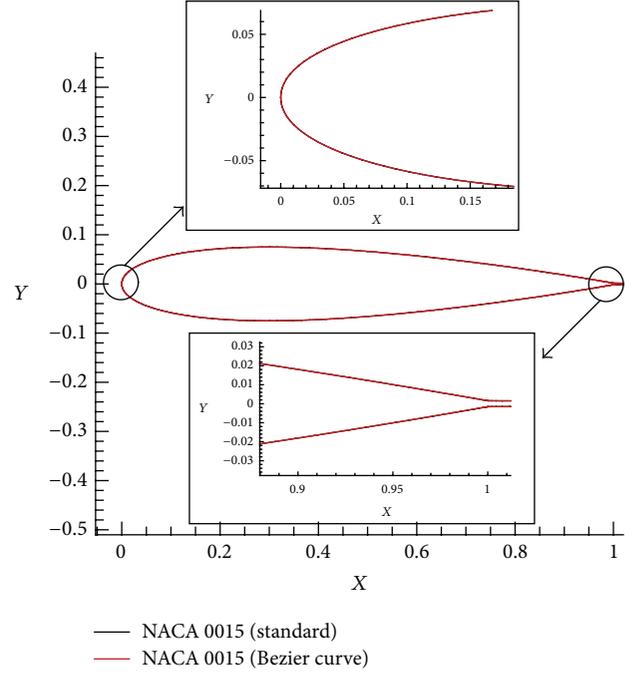


FIGURE 8: Comparison between the standard airfoil and the Bezier curve for a NACA0015.

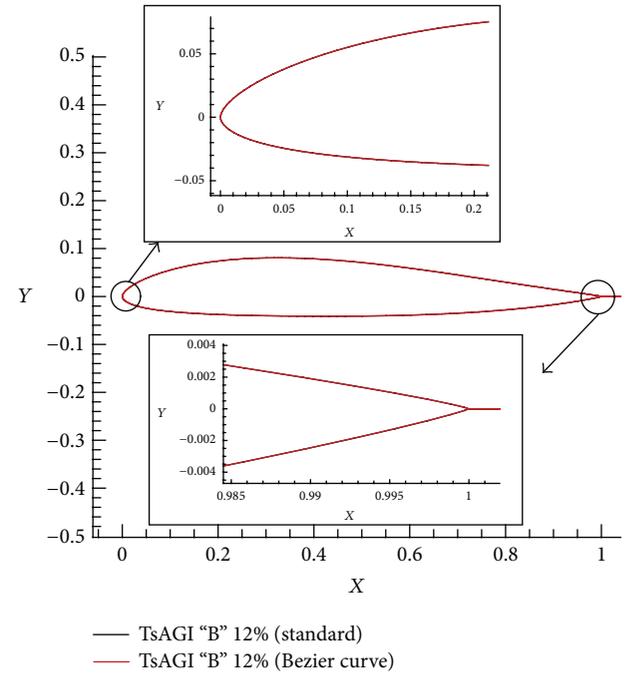


FIGURE 9: Comparison between the standard airfoil and the Bezier curve for a TsAGI "B" 12%.

where  $x$  is coordinates along the chord of the airfoil, from 0 to  $c$  ( $c$  is the *chord length* and is assumed equal to 1),  $y_t$  is the thickness coordinates above and below the line extending along the length of the airfoil, and  $t$  is maximum thickness of the airfoil in percentage of chord (i.e.,  $t$  in a %15 thick airfoil would be 0.15). Equation (25) can be used to find

the  $y$ -coordinates of a NACA 00xx symmetric airfoil by knowing the values for  $x$  and  $t$ . As will be shown, the maximum thickness of such airfoils will also be considered as a design variable. By optimizing the thickness, the optimal shape for such airfoils will be obtained. This kind of optimization problem, however, is not comprehensive and produces the optimal NACA 00xx symmetric airfoils only. In summary, three kinds of design variable will be considered in this paper for airfoil shape optimization which are grid points on a given airfoil surface extracted from, say, the software JavaFoil, the Bezier curve control points, and the maximum thickness of NACA 00xx symmetric airfoils.

## 5. Shape Optimization

Different objective functions may be considered for the aerodynamics shape optimization including maximizing the lift-drag ratio, maximizing the lift, and minimizing the drag. In the framework of this paper, the shape optimization problem will be to infer the shape an airfoil should have so that the pressure distribution on the airfoil surface matches a prescribed one (an inverse problem). In inverse design problem, the desired pressure distribution of the target design may be specified a priori.

*Design Variable (DV).* Here the airfoil grid points, the Bezier curve control points, and the maximum thickness of NACA00xx airfoils are considered as design variables. Therefore, one has the following:

*Case 1:* the airfoil grid points as design variable (see Figure 13);

*Case 2:* the Bezier curve control points as design variable;

*Case 3:* the maximum thickness of NACA00xx airfoils.

*Case 1.* The mathematical expression for the objective function considered for Case 1 can be stated as

$$\mathcal{J} = \sum_{j=2, j \neq (N+1)/2}^{N-1} (P_{1,j} - P_{d(1,j)})^2, \quad (26)$$

where  $P_{(1,j)}$  is the pressure at grid points  $F_{1,j}$  on the airfoil surface and  $P_{d(1,j)}$  is the desirable pressure at grid points  $F_{1,j}$  on the airfoil surface (Figure 10). The aim is to minimize  $\mathcal{J}$  and to reach the desirable pressure distribution by changing the position of the grid points on the airfoil surface. Since the  $x$ -coordinates of the grid points can be constant during the optimization process, only the  $y$ -coordinates of the grid points are considered as design variables. Two end points of airfoil, namely, *leading edge* ( $j = (N + 1)/2$ ) and *trailing edge* ( $j = 1, N$ ), are fixed. Thus they are not considered as design variables.

*Case 2.* The mathematical expression for the objective function considered for Case 2 can be stated as

$$\mathcal{J} = \sum_{i=1}^{2m-4} (P_{iB} - P_{iB_d})^2, \quad (27)$$

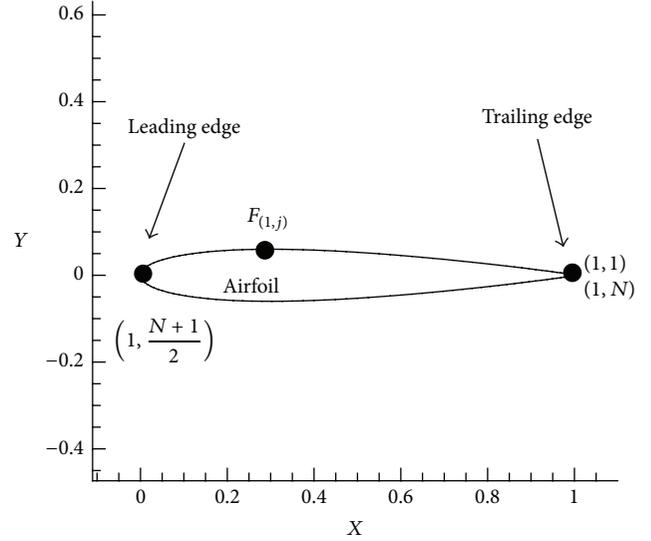


FIGURE 10: Illustration of the airfoil surface points to be optimized so that the objective function reaches a minimum.

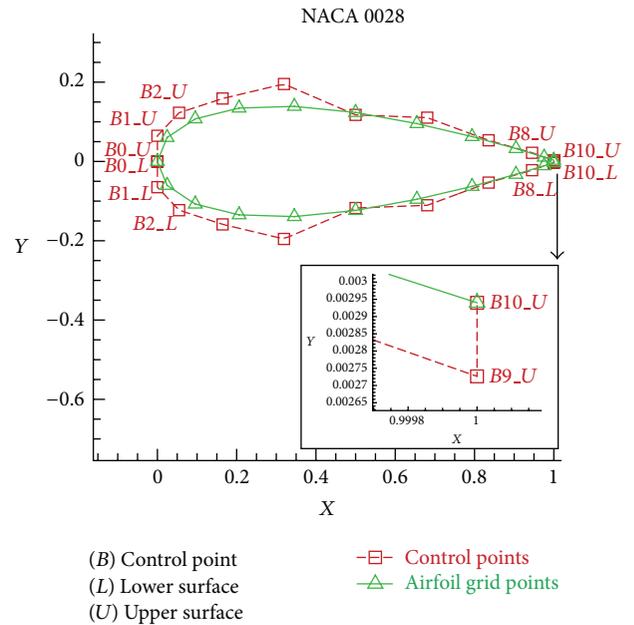


FIGURE 11: Illustration of the Bezier control points ( $B_i$ ) to be optimized so that the objective function (see (27)) reaches a minimum.

where  $m$  is the number of the predetermined data on each of the upper and the lower surfaces of the airfoil,  $P_{iB}$  is the pressure at point  $i$  of the airfoil surface generated by the Bezier curve, and  $P_{iB_d}$  is the desirable pressure at point  $i$ . Why does  $2m - 4$ ?  $m$  data points for the upper surface,  $m$  data points for the lower surface, and the leading and the trailing edges for two surfaces are considered fixed. The aim is to minimize  $\mathcal{J}$  and to reach the desirable pressure distribution by changing the  $y$ -position of the control points  $B_i$  ( $i = 1, \dots, 9$ ) on each of the upper and the lower surfaces of the airfoil (see Figure 11).  $B_0$  and  $B_{10}$ , which are concerned with the leading

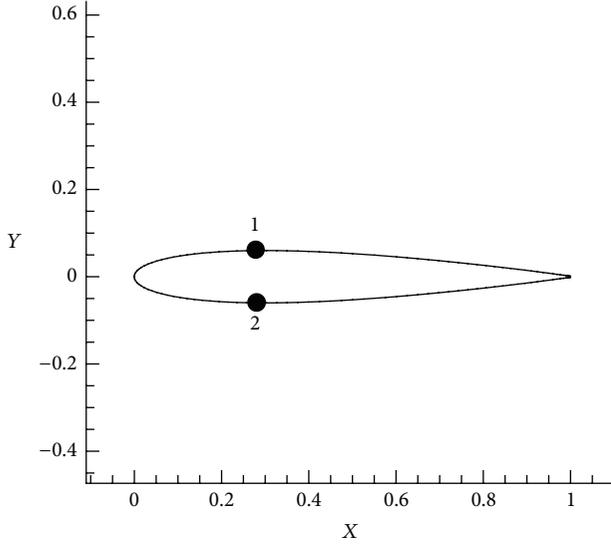


FIGURE 12: The location for the maximum thickness on the upper and lower airfoil surfaces.  $y$ -coordinates of the points 1 and 2 are considered as the design variables.

edge and the trailing edge, respectively, are considered fixed for both upper and lower surfaces. Therefore, for the shape optimization problem with a Bezier curve of order 11, we have  $2 \times (11 - 2) = 18$  design variables. For the shape optimization problem with a Bezier curve of order 7, we have  $2 \times (7 - 2) = 10$  design variables. The reason for considering these two kinds of the Bezier curve is twofold:

- (1) to show that the optimization problem will be more successful if we have less number of design variable;
- (2) to have a very accurate and flexible representation of the airfoil shapes, a degree of at least 10 should be used.

*Case 3.* The airfoil surface is generated by the analytical NACA formula (25) and the maximum thickness is considered as the design variable. To show the accuracy of the sensitivity scheme, the upper and lower airfoil surfaces are generated separately and hence the design variables will be two maximum thicknesses in the upper and lower airfoil surfaces. As shown in Figure 12, if the indices 1 and 2 denote the location of maximum thickness on the upper and lower airfoil surfaces, respectively, then the mathematical expression for the objective function considered for Case 3 is as follows:

$$\mathcal{F} = \sum_{i=1}^2 (P_i - P_{d(i)})^2. \quad (28)$$

## 6. Sensitivity Analysis

Suppose we wish to calculate the sensitivity of pressure of nodes on the airfoil surface (see Figure 10),  $P_{1,j}$  ( $j = 2, \dots, N - 1, j \neq (N + 1)/2$ ), to the  $y$ -position of the nodes on the airfoil surface,  $y_{1,j'}$  ( $j' = 2, \dots, N - 1, j' \neq (N + 1)/2$ ).

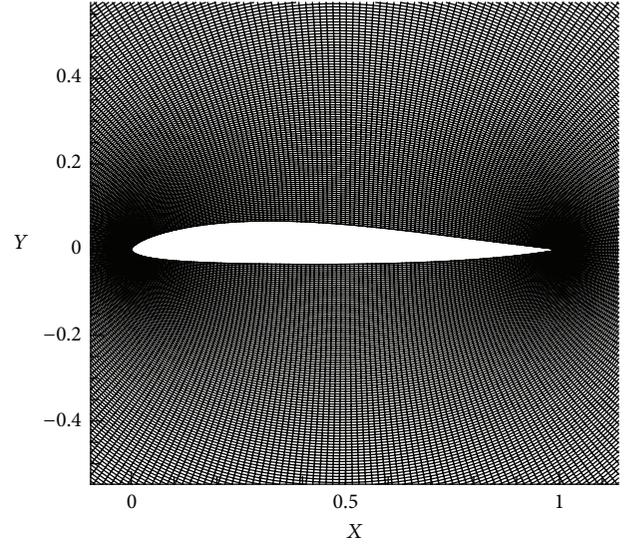


FIGURE 13: Grid used in Test Case 1 (around initial shape).

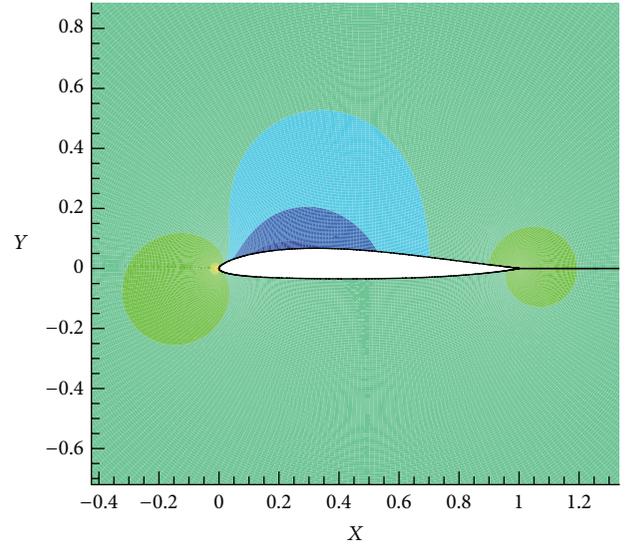


FIGURE 14: Pressure distribution around the airfoil surface (initial shape).

The sensitivity analysis can be performed by introducing small perturbations to the  $y$ -coordinate of each point on the airfoil surface, individually. The grid generation and flow problem may be solved for this perturbed shape to obtain the new values for the pressure  $P_{1,j}$ . Using these values for the pressure, the dependency of the pressure  $P_{1,j}$  to the perturbation of the  $y$ -position of points of coordinates  $(1, j')$ ,  $y_{1,j'}$ , can be evaluated. The finite difference method may be used to formulate these sensitivities as follows:

$$\frac{\partial P_{1,j}}{\partial y_{1,j'}} = \frac{P_{1,j}(y_{1,j'} + \epsilon y_{1,j'}) - P_{1,j}(y_{1,j'})}{\epsilon y_{1,j'}}, \quad (29)$$

where  $\epsilon$  may be, say,  $10^{-6}$ . The term  $\epsilon y_{1,j'}$  is the perturbation in the  $y$ -position of points of coordinates  $(1, j')$ ,  $y_{1,j'}$ .

Since the sensitivity of each pressure  $P_{1,j}$  ( $j = 2, \dots, N - 1, j \neq (N + 1)/2$ ) to each  $y$ -position of points of coordinates  $(1, j')$  ( $j' = 2, \dots, N - 1, j' \neq (N + 1)/2$ ) is required, the computation of the sensitivity coefficients using this method requires  $(N - 3)$  additional solutions of the flow problem. Therefore, this method is only suitable when the number of points on the airfoil surface is small. Thus for the airfoil shape optimization problem, which demand a fine grid to obtain accurate results, the perturbation method using the finite difference method will be of high computation cost. In this paper, we will expand the novel method used in evaluating the sensitivity matrix in the shape optimization of heat transfer problems. As will be shown, it requires only one solution of the flow problem (at each iteration) to compute all sensitivity coefficients.

With regard to (26), the following equation can be written in order to calculate the Jacobian matrix

$$\frac{\partial \mathcal{F}}{\partial y_{1,l}} = 2 \sum_{j=2, j \neq (N+1)/2}^{N-1} (P_{1,j} - P_{d(1,j)}) \frac{\partial P_{1,j}}{\partial y_{1,l}}, \quad (30)$$

where ( $j = 2, \dots, N - 1, j \neq (N + 1)/2$ ) and ( $l = 2, \dots, N - 1, l \neq (N + 1)/2$ ). The expression  $\partial P_{1,j}/\partial y_{1,l}$  in the above relation is called the Jacobian coefficient. In this case, the sensitivity matrix can be expanded as

$$\mathbf{J}_{\mathbf{a}_y} = \begin{bmatrix} \frac{\partial P_{1,2}}{\partial y_{1,2}} & \frac{\partial P_{1,2}}{\partial y_{1,3}} & \frac{\partial P_{1,2}}{\partial y_{1,4}} & \dots & \frac{\partial P_{1,2}}{\partial y_{1,N-1}} \\ \frac{\partial P_{1,3}}{\partial y_{1,2}} & \frac{\partial P_{1,3}}{\partial y_{1,3}} & \frac{\partial P_{1,3}}{\partial y_{1,4}} & \dots & \frac{\partial P_{1,3}}{\partial y_{1,N-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial P_{1,N-1}}{\partial y_{1,2}} & \frac{\partial P_{1,N-1}}{\partial y_{1,3}} & \frac{\partial P_{1,N-1}}{\partial y_{1,4}} & \dots & \frac{\partial P_{1,N-1}}{\partial y_{1,N-1}} \end{bmatrix}. \quad (31)$$

Since the physical domain is mapped onto the computational one, the chain rule may be used to correlate variables in the two domains. Therefore,

$$\frac{\partial P_{1,j}}{\partial x_{1,l}} = \frac{\partial P_{1,j}}{\partial \xi} \frac{\partial \xi}{\partial x_{1,l}} + \frac{\partial P_{1,j}}{\partial \eta} \frac{\partial \eta}{\partial x_{1,l}}, \quad (32)$$

$$\frac{\partial P_{1,j}}{\partial y_{1,l}} = \frac{\partial P_{1,j}}{\partial \xi} \frac{\partial \xi}{\partial y_{1,l}} + \frac{\partial P_{1,j}}{\partial \eta} \frac{\partial \eta}{\partial y_{1,l}}. \quad (33)$$

As pointed out before, the  $x$ -coordinate of the grid points are considered fixed and they are not included in the design variables. Thus (32) is written here to derive the required relations for the sensitivity coefficients. By interchanging  $x$  and  $\xi$ , and  $y$  and  $\eta$ , and solving the derived equations for  $\partial P/\partial x$  and  $\partial P/\partial y$ , we finally obtain

$$\frac{\partial P_{1,j}}{\partial y_{1,l}} = \frac{1}{J} \left[ -(x_\eta)_{1,l} (P_\xi)_{1,j} + (x_\xi)_{1,l} (P_\eta)_{1,j} \right], \quad (34)$$

where  $J = (x_\xi y_\eta - x_\eta y_\xi)_{1,l}$  is the Jacobian of the transformation. Using the finite difference method to discretize the equations in the computational domain, we can write appropriate

algebraic approximations for all partial derivatives involved in the above equation. Therefore,

$$(P_\xi)_{1,j} = \frac{-3P_{1,j} + 4P_{2,j} - P_{3,j}}{2}, \quad (35)$$

$$(P_\eta)_{1,j} = \frac{P_{1,j+1} - P_{1,j-1}}{2}, \quad (36)$$

$$(x_\xi)_{1,l} = \frac{-3x_{1,l} + 4x_{2,l} - x_{3,l}}{2}, \quad (37)$$

$$(x_\eta)_{1,l} = \frac{x_{1,l+1} - x_{1,l-1}}{2} \quad (38)$$

which are based on the central and the forward differences. Equations (35) through (38) are employed to calculate the sensitivity coefficients in (31).

*Bezier Control Points as Design Variables.* With regard to (27) and considering the control points of the Bezier curve as design variable, we can write

$$\frac{\partial \mathcal{F}}{\partial B_{y_l}} = 2 \sum_{i=1}^{2m-4} (P_{iB} - P_{iB_d}) \frac{\partial P_{iB}}{\partial B_{y_l}}. \quad (39)$$

Using the chain rule, we can write

$$\frac{\partial P_{iB}}{\partial B_{y_l}} = \frac{\partial P_{iB}}{\partial y_{i'B}} \frac{\partial y_{i'B}}{\partial B_{y_l}}, \quad (40)$$

where  $y_{i'B}$  ( $i' = 1, \dots, 2m - 4$ ) are the  $y$ -coordinate of the predetermined grid points to be passed by the Bezier curve and  $B_{y_l}$  ( $l = 1, \dots, 18$ ) are the  $y$ -coordinate of Bezier control points whose number is equal to 18 (9 for each of the upper and lower surfaces). The term  $\partial P_{iB}/\partial y_{i'B}$  can be computed by the expressions derived for Case 1 (see (31)). The size of the matrix formed by the arrays  $\partial P_{iB}/\partial y_{i'B}$  is  $(2m - 4) \times (2m - 4)$ . The term  $\partial y_{i'B}/\partial B_{y_l}$  can be easily evaluated by taking derivative of (19) with respect to the control points  $B$  (noting that  $[P(t)] \equiv [x(t), y(t)]$ ). The control points may be renumbered so that  $B_{y_1} = B_{y_{9U}}, B_{y_2} = B_{y_{8U}}, \dots, B_{y_9} = B_{y_{1U}}$  and  $B_{y_{10}} = B_{y_{1L}}, B_{y_{11}} = B_{y_{2L}}, \dots, B_{y_{18}} = B_{y_{9L}}$ . The indices  $U$  and  $L$  denote the upper and lower surfaces, respectively. The direction of numbering is from right to left for the upper surface and from left to right for the lower surface. The reason for this renumbering is the compatibility with the grid point data reading (most of the airfoil data are in this format) as well as the pressure reading to compute the objective function (see (27)). However, we should note that the Bezier curve evaluation is from left to right for both the upper and lower surfaces. The size of the matrix formed by the arrays  $\partial y_{i'B}/\partial B_{y_l}$  is  $(2m - 4) \times 18$ . Because the upper and lower surfaces are constructed separately, the variation of  $y$  of the upper surface with respect to the change in position of the lower surface control points as well as the variation of  $y$

of the lower surface with respect to the change in position of the upper surface control points is zero.

*Maximum Thickness as Design Variables.* In a similar derivation to Case 1, the sensitivity matrix for Case 3 will be

$$\mathbf{J}\mathbf{a} = \begin{bmatrix} \frac{\partial P_1}{\partial y_1} & \frac{\partial P_1}{\partial y_2} \\ \frac{\partial P_2}{\partial y_1} & \frac{\partial P_2}{\partial y_2} \end{bmatrix}. \quad (41)$$

## 7. Optimization Method

In this paper, two powerful optimization methods, namely, the conjugate gradient method and the quasi-Newton method will be used. For the airfoil grid points as a design variable (Case 1) both optimization methods will be employed. However, for the Bezier curve control points as a design variable (Case 2), only the quasi-Newton method will be used. For Case 3 (the maximum thickness of NACA00xx airfoils as a design variable), only the conjugate gradient method will be employed.

*Conjugate Gradient Method.* The conjugate gradient algorithm to obtain the optimal shape for the airfoil is as follows.

- (1) Specify the physical domain, the boundary conditions, the problem conditions such as Mach number and the angle of attack, and the desired airfoil surface pressure distribution.
- (2) Generate the boundary fitted grids using the grid generation methods described earlier.
- (3) Solve the direct flow problem of finding the pressure values at any grid points of the physical domain and hence the airfoil surface.
- (4) Using (26), compute the objective function ( $\mathcal{J}^{(k)}$ ).
- (5) If the value of the objective function obtained in step (4) is less than the specified stopping criterion, the optimization is finished. Otherwise, go to step (6).
- (6) Compute the sensitivity matrix ( $\mathbf{J}\mathbf{a}$ ) from (31).
- (7) Compute the gradient direction  $\nabla \mathcal{J}^{(k)}$  from (30).
- (8) Compute the conjugation coefficient  $\gamma^{(k)}$  from the following equation (the Polak-Ribiere formula):

$$\begin{aligned} \gamma^{(k)} &= \frac{[\nabla \mathcal{J}^{(k)}]^T (\nabla \mathcal{J}^{(k)} - \nabla \mathcal{J}^{(k-1)})}{\|\nabla \mathcal{J}^{(k-1)}\|^2} \\ &= \frac{[\nabla \mathcal{J}^{(k)}]^T (\nabla \mathcal{J}^{(k)} - \nabla \mathcal{J}^{(k-1)})}{[\nabla \mathcal{J}^{(k-1)}]^T \nabla \mathcal{J}^{(k-1)}}. \end{aligned} \quad (42)$$

For  $k = 0$ , set  $\gamma^{(0)} = 0$ .

- (9) Compute the direction of descent  $\mathbf{d}^{(k)}$  from the following:

$$\mathbf{d}^{(k)} = \nabla \mathcal{J}^{(k)} + \gamma^{(k)} \mathbf{d}^{(k-1)}. \quad (43)$$

- (10) Compute the search step size  $\beta^{(k)}$  from the following:

$$\beta^{(k)} = \frac{[\mathbf{J}\mathbf{a}^{(k)} \mathbf{d}^{(k)}]^T [P_{1,j} - P_{d(1,j)}]}{[\mathbf{J}\mathbf{a}^{(k)} \mathbf{d}^{(k)}]^T [\mathbf{J}\mathbf{a}^{(k)} \mathbf{d}^{(k)}]}. \quad (44)$$

- (11) Evaluate the new  $y$ -coordinates of the airfoil surface grid nodes as follows:

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} - \beta^{(k)} \mathbf{d}^{(k)}. \quad (45)$$

- (12) Set the next iteration ( $k = k + 1$ ) and return to step (2).

The above algorithm is for the airfoil grid points as a design variable (Case 1) only. The algorithm for Case 3 can be expressed in a similar way.

*Quasi-Newton Method.* Quasi-Newton method is another powerful optimization method used in this paper. In quasi-Newton method, the Hessian matrix (which is composed of the second partial derivatives) is replaced by an approximation of it. The approximation uses only the first partial derivatives. The *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) method is a quasi-Newton method for solving unconstrained nonlinear optimization. In the BFGS method, the Hessian matrix approximation,  $\mathbf{B}^{(k)}$ , is updated iteratively. The steps of BFGS method can be summarized as follows.

- (1) Specify the physical domain, the boundary conditions, the problem conditions such as Mach number and the angle of attack, and the desired airfoil surface pressure distribution.
- (2) Generate the boundary fitted grids using the grid generation methods described earlier.
- (3) Solve the direct flow problem of finding the pressure values at any grid points of the physical domain and hence the airfoil surface.
- (4) Using (26), compute the objective function ( $\mathcal{J}^{(k)}$ ).
- (5) If value of the objective function obtained in step (4) is less than the specified stopping criterion, the optimization is finished. Otherwise, go to step (6).
- (6) Compute the sensitivity matrix ( $\mathbf{J}\mathbf{a}$ ) from (31).
- (7) Compute the gradient direction  $\nabla \mathcal{J}^{(k)}$  from (30).
- (8) The initial Hessian matrix approximation,  $\mathbf{B}^{(1)}$ , is taken as the identity matrix, namely,  $\mathbf{B}^{(1)} = \mathbf{I}$ .
- (9) Set  $\mathbf{S}^{(k)} = -\mathbf{B}^{(k)} \nabla \mathcal{J}^{(k)}$  and the iteration number as  $k = 1$ .
- (10) Compute the search step size  $\beta^{(k)}$  (from (44)) in the direction  $\mathbf{S}^{(k)}$  and set

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} - \beta^{(k)} \mathbf{S}^{(k)}. \quad (46)$$

- (11) Repeat the steps (2) to (7) with these new values of  $\mathbf{y}$  for the grid points  $y$ -coordinates to calculate  $\nabla \mathcal{J}^{(k+1)}$ .

TABLE 1: Data used for Test Case 1.

	Airfoil	Grid size	Angle of attack $\alpha$	Free stream velocity $V_\infty$
Initial	TsAGI“B” 10%	300 × 305	1°	70 m/s
Desired	NACA 0015	350 × 365	1°	70 m/s

(12) Update the Hessian matrix approximation as

$$\mathbf{B}^{(k+1)} = \mathbf{B}^{(k)} + \left( 1 + \frac{(\mathbf{g}^{(k)})^T \mathbf{B}^{(k)} \mathbf{g}^{(k)}}{(\mathbf{d}^{(k)})^T \mathbf{g}^{(k)}} \right) \frac{\mathbf{d}^{(k)} (\mathbf{d}^{(k)})^T}{(\mathbf{d}^{(k)})^T \mathbf{g}^{(k)}} - \frac{\mathbf{d}^{(k)} (\mathbf{g}^{(k)})^T \mathbf{B}^{(k)}}{(\mathbf{d}^{(k)})^T \mathbf{g}^{(k)}} - \frac{\mathbf{B}^{(k)} \mathbf{g}^{(k)} (\mathbf{d}^{(k)})^T}{(\mathbf{d}^{(k)})^T \mathbf{g}^{(k)}}, \quad (47)$$

where

$$\begin{aligned} \mathbf{d}^{(k)} &= \mathbf{y}^{(k+1)} - \mathbf{y}^{(k)} = -\beta^{(k)} \mathbf{S}^{(k)}, \\ \mathbf{g}^{(k)} &= \nabla \mathcal{J}^{(k+1)} - \nabla \mathcal{J}^{(k)}. \end{aligned} \quad (48)$$

(13) Set the new iteration number as  $k = k + 1$  and go to step (9).

## 8. Results

In this section, the results obtained for the shape optimization of an airfoil in the incompressible, irrotational, and inviscid flow under given boundary conditions are presented. Three kinds of the design variable (the airfoil grid points, the Bezier curve control points, and the maximum thickness of NACA 00xx airfoils) as well as two optimization methods (CG and BFGS) are considered. In all test cases in this paper which employ the Bezier curve, the number of predetermined airfoil data,  $m$ , is set equal to the Bezier curve order,  $n + 1$ .

*Test Case 1.* In this test case, the airfoil surface is parameterized by a Bezier curve of order 11. The total number of the design variable is 18, namely, 9 design variables for each of the upper and lower surfaces. At first, two parametric curves for two surfaces (upper and lower) are obtained using 11 grid points and then a fine grid is generated to obtain accurate results. The data for Test Case 1 is given in Table 1. The comparison of the initial and optimal airfoil shapes and some magnified parts of them are shown in Figures 15, 20, and 26. In this test case, BFGS optimization method is employed.

The convergence of the objective function is shown in Figures 17, 23, 29, 37, and 39. The initial and minimum values for the objective function are approximately 3517433 and 2877116, respectively, which shows %18.2 reduction in objective function (see Figures 16, 21, and 27). The minimum value for the objective function takes place in iteration 14. The optimization time spent on the 1st iteration (which is equivalent to one direct flow solution) is 11 minutes and 43 seconds and the total optimization time for 30 iterations is 11 minutes and 46 seconds which shows the proposed sensitivity

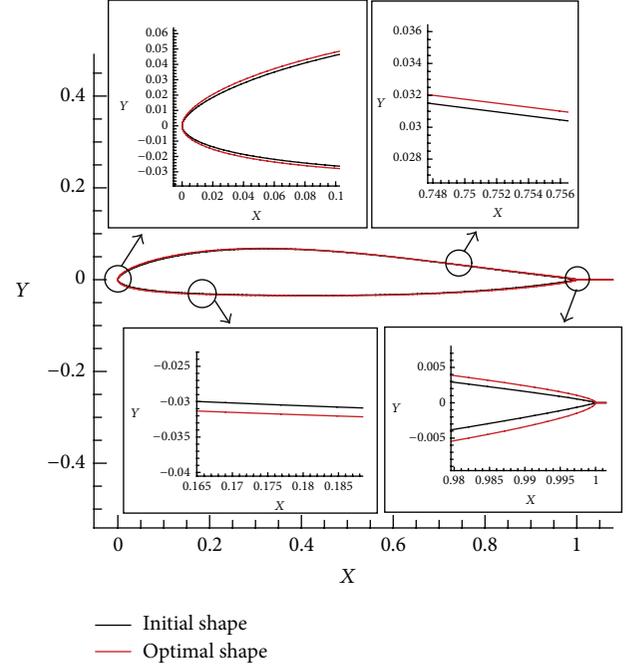


FIGURE 15: Comparison of the initial and optimal shapes and some magnified parts including the leading edge, middle parts, and the trailing edge.

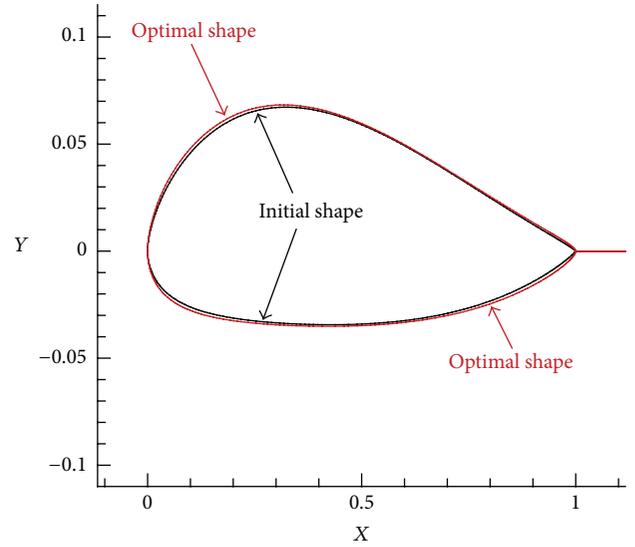


FIGURE 16: Comparison of the initial and optimal shapes. The  $y$ -axis has been greatly exaggerated to highlight difference in the airfoil shapes.

analysis efficiency. 30 iterations take only 3 seconds. The reason for the difference between the 1st iteration and the following ones is that the solution after the 1st iteration is a very good initial guess for the 2nd iteration and the direct solution converges quickly. In other words, what is a bit time consuming for the 1st iteration is the grid generation and stream function loops not the pressure calculation, sensitivity analysis, and optimization stages. Moreover, a fine

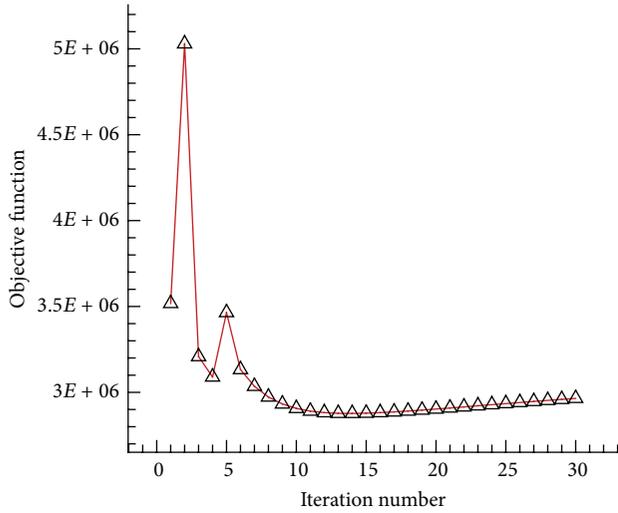


FIGURE 17: Objective function value versus the iteration number.

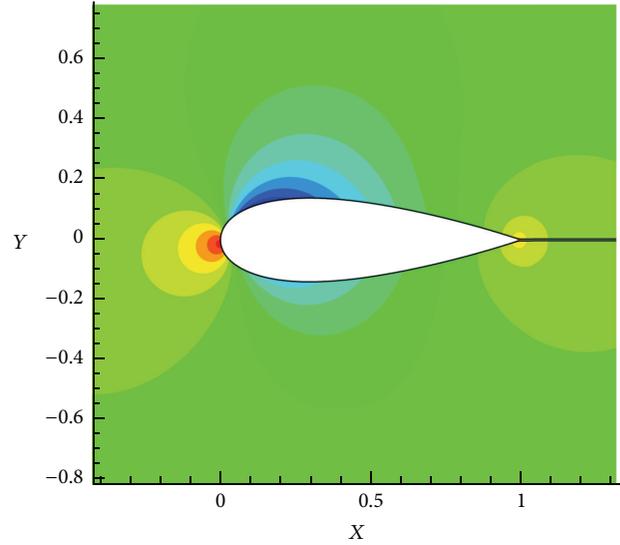


FIGURE 19: Pressure distribution around the airfoil surface (initial shape).

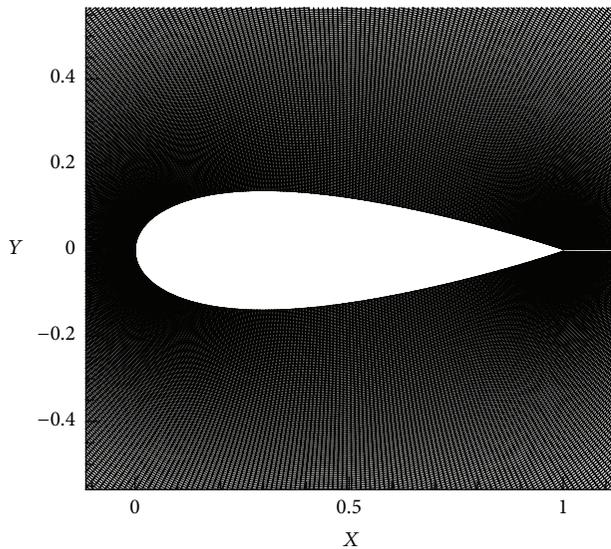


FIGURE 18: Grid used in Test Case 2 (around initial shape).

grid ( $300 \times 305$ ) and a tolerance of  $10^{-8}$  are used in the iterative loops which increase the computation time. The code is programmed in Fortran 77 using a Fortran compiler (Force 2.0) and the computations are run on a PC with Intel Pentium Dual 1.73 and 1G RAM. All the computations in the test cases in this paper are performed using the above mentioned compiler and PC. Therefore, there is no need to repeat it in the following test cases.

*Test Case 2.* Test Case 2 is similar to Test Case 1 but with different specifications (see Figure 18). The data for this test case is given in Table 2.

The explanation is similar to Test Case 1. Thus only the results will be given in Table 3.

Now an optimal shape design problem using a Bezier curve of order 7 is given. As it will be shown, it decreases the objective function value much bigger than when using

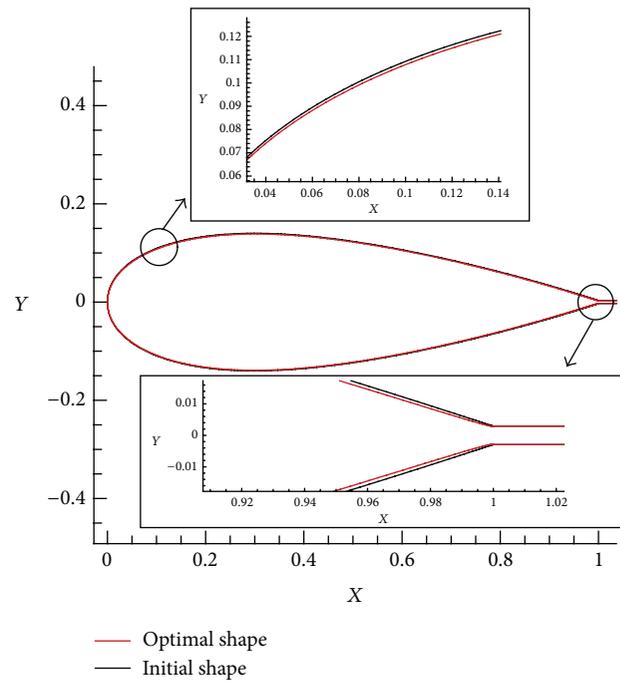


FIGURE 20: Comparison of the initial and optimal shapes and some magnified parts including the leading and trailing edges.

TABLE 2: Data used for Test Case 2.

	Airfoil	Grid size	Angle of attack $\alpha$	Free stream velocity $V_\infty$
Initial	NACA 0028	$400 \times 425$	$2^\circ$	70 m/s
Desired	NACA 0016	$350 \times 365$	$2^\circ$	70 m/s

a Bezier curve of order 11 as there is less number of design variables (10 design variables for a Bezier curve of order 7

TABLE 3: Results for Test Case 2.

Initial objective function	Minimum objective function (iteration 3)	Computational time, 1st iteration	Total computational time, 30 iterations	Percentage of reduction in objective function
10074507	5585758	35 m and 30 s	35 m and 34 s	44.5%

TABLE 4: Data used for Test Case 3.

	Airfoil	Grid size	Angle of attack $\alpha$	Free stream velocity $V_\infty$
Initial	NACA 0012	$300 \times 305$	$0^\circ$	70 m/s
Desired	NACA 0017	$250 \times 305$	$0^\circ$	70 m/s

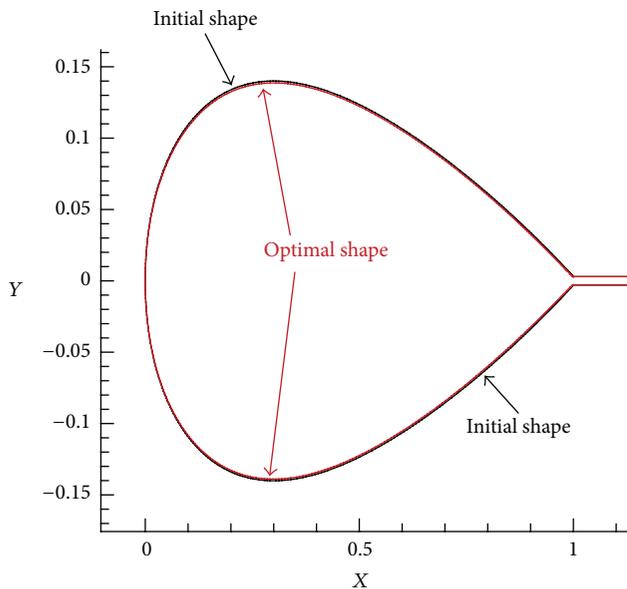


FIGURE 21: Comparison of the initial and optimal shapes. The  $y$ -axis has been greatly exaggerated to highlight difference in the airfoil shapes.

versus 18 design variables for a Bezier curve of order 11). However, as it will be shown, using a Bezier curve of order 7 is not comprehensive for all airfoil shapes and is suitable to NACA 00XX or similar airfoils only. In other words, it is not able to produce all airfoil shapes precisely.

*Test Case 3* (using a Bezier curve of order 7; see Figure 24 and Table 4). The results are given in Table 5.

Although there is a decrease of %59 in objective function, the %59 approach from the initial shape to desired one is not seen (see Figure 28). This indicates that the solution of the inverse problem is not unique. The reason for this can be found in the trailing edge configuration for the initial, optimal, and desired airfoil shapes (Figure 28).

Although the results of using the Bezier curve of order 7 is very promising, its drawback is that it is restricted to the simple and symmetric airfoil shapes such as NACA 00xx.

For other airfoil shapes, there can be seen some oscillations around the trailing edge (see Figure 30).

Moreover, the Bezier curve of order 7 fails to represent the NACA 00xx airfoils accurately. In other words, the Bezier curve of order 11 is the appropriate option to produce very accurate airfoils (Figure 31).

*Test Case 4.* In this test case, the airfoil surface is parameterized by grid points obtained from the analytical NACA formula (e.g., software JavaFoil). In this case, the number of the design variables is equal to the number of grid points minus three (one for leading edge and two for trailing edge). Therefore, we have an aerodynamic shape optimization problem with a high number of the design variables as we should have a fine grid to obtain sufficiently accurate results. It is known that the optimization process becomes more challenging by increasing the number of design variables. Hence we have a difficult shape optimization problem in Test Case 4. The data used for Test Case 4 is given in Table 6. A very fine grid ( $400 \times 425$ ) is used for the initial airfoil shape (NACA 0012). The number of the design variables is  $N - 3$  which is  $425 - 3 = 422$ . Therefore, a time consuming optimization problem is expected. However, by using the sensitivity analysis scheme proposed in this paper, the total time for the optimization problem in Test Case 4 using both CG and BFGS optimization methods is about 46 minutes for 8000 iterations. The computation time for the 1st iteration is about 25 minutes. The comparison of the computation times for the 1st iteration and 8000 ones indicates again the efficiency of the sensitivity analysis scheme. The summary of the results is presented in Table 7. The comparison of the initial, optimal, and desired airfoil shapes is given in Figure 32. As can be seen in the figure, the variation of the shape is minute. The convergence of the objective function to a local minimum when both the CG and BFGS optimization methods are used as well as a comparison of them is shown in Figures 33, 34, and 35, respectively. The plots reveal the better performance of the BFGS method in minimizing the objective function.

In Test Cases 5 and 6 the maximum thickness of the NACA 00xx is considered as a design variable. As mentioned previously, the conjugate gradient method is employed as the optimization method.

*Test Case 5.* The data for the problem including the conditions for the initial and desired airfoils are given in Table 8.

Figure 36 represents the comparison of the initial, optimal, and desired shapes for airfoils (also see Figures 22 and 38). The desired airfoil shape is a NACA0018 at conditions stated in Table 8. As can be seen, this shape is shown by

TABLE 5: Results for Test Case 3.

Initial objective function	Minimum objective function (iteration 34)	Computational time, 1st iteration	Total computational time, 100 iterations	Percentage of reduction in objective function
898639	368148	9 m and 32 s	9 m and 43 s	59%

TABLE 6: Data used for Test Case 4.

	Airfoil	Grid size	Angle of attack $\alpha$	Free stream velocity $V_\infty$
Initial	NACA 0012	400 × 425	0°	70 m/s
Desired	NACA 0014	400 × 425	0°	70 m/s

TABLE 7: Summary of the results of Test Case 4.

	Initial $\mathcal{J}$	Final $\mathcal{J}$	Number of iterations	Computation time (total)	Reduction in $\mathcal{J}$
CG	6351997	6053996	8000	~46 mins	4.7%
BFGS	6351997	6036943	8000	~46 mins	5%

TABLE 8: Data used for Test Case 5.

	Airfoil	Grid size	Angle of attack $\alpha$	Free stream velocity $V_\infty$
Initial	NACA 0011	80 × 125	2°	70 m/s
Desired	NACA 0018	80 × 125	2°	70 m/s

a black color line. The optimization process is started by a NACA0011 as an initial shape which is shown by a red color line. The optimal shape (shown by a blue color line), which is obtained by the conjugate gradient method, is in an excellent agreement (full matching) with the desired one. The objective function variation is shown in Figure 37. The initial and final values for the objective function are about 773597.45 and 15.48, respectively, which reveals an approximately %100 reduction in the objective function within 31 iterations. The total time for the optimization (for 31 iterations) is 2 minutes and 14 seconds. The tolerance used in iterative steps in the program is  $10^{-8}$ . Although such a tolerance value increases the computation time, it enhances the accuracy of the results. If the  $y$ -components of the maximum thickness in upper and lower airfoil surfaces are denoted by  $y_{thickU}$  and  $y_{thickL}$ , respectively, the value of the pressure for these two locations for initial, optimal, and desired shapes are reported in Tables 10 and 13. The difference values show the validity of the shape optimization process.

*Test Case 6.* The data for the problem is given in Table 11. The explanation for the results is similar to Test Case 5 (see Tables 9 and 12).

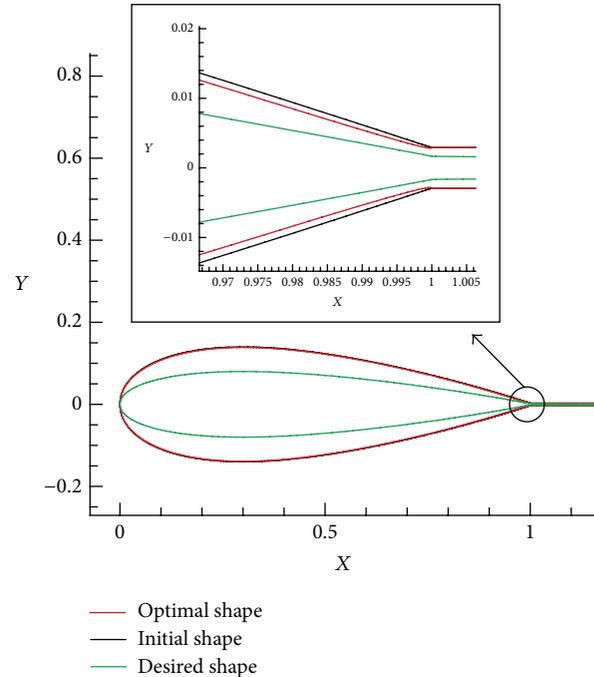


FIGURE 22: Comparison of the initial, optimal, and desired shapes.

TABLE 9: Results for Test Case 5.

	Initial $\mathcal{J}$	Final $\mathcal{J}$	Number of iterations	Computation time (total)	Reduction in $\mathcal{J}$
CG	773597.45	15.48	31	2 m : 14 s	~100%

### 9. Adjoint Method

As pointed out previously, for the aerodynamic shape optimization problems requiring a large number of design variables, the use of finite difference method to evaluate the gradient by introducing a small perturbation to each design variable separately and then solving the flow problem is of very high computational cost, because it requires a number of additional flow solutions equal to the number of design variables. For optimal shape design problems with a high number of design variables, the adjoint method [4] can compute the gradients of objective function much faster than the finite difference method.

The aerodynamic shape optimization problem of interest here can be expressed as

$$\begin{aligned}
 &\text{minimization of objective function } \mathcal{J} \\
 &\text{subject to constraint } \mathcal{R} = 0 \text{ (the governing equation)}.
 \end{aligned}
 \tag{49}$$

TABLE 10: Comparison of the pressure at the maximum thicknesses of the airfoil surface (upper and lower surfaces) for the initial, optimal, and desired shapes.

	Initial shape	Optimal shape	Desired shape	Difference (Pa)
Pressure at $y_{thickU}$ (Pa)	100080.04	99411.37	99410.06	Difference in initial and desired = 669.98 Difference in optimal and desired = 1.31
Pressure at $y_{thickL}$ (Pa)	100814.39	100248.25	100244.54	Difference in initial and desired = 569.85 Difference in optimal and desired = 3.71

TABLE 11: Data used for Test Case 6.

	Airfoil	Grid size	Angle of attack $\alpha$	Free stream velocity $V_\infty$
Initial	NACA 0015	$80 \times 125$	$2^\circ$	70 m/s
Desired	NACA 0035	$80 \times 125$	$2^\circ$	70 m/s

TABLE 12: Results for Test Case 6.

	Initial $\mathcal{J}$	Final $\mathcal{J}$	Number of iterations	Computation time (total)	Reduction in $\mathcal{J}$
CG	7986880.34	1.15	21	1 m : 08 s	~100%

TABLE 13: Comparison of the pressure at the maximum thicknesses of the airfoil surface (upper and lower surfaces) for the initial, optimal, and desired shapes.

	Initial shape	Optimal shape	Desired shape	Difference (Pa)
Pressure at $y_{thickU}$ (Pa)	99703.83	97552.39	97552.99	Difference in initial and desired = 2150.84 Difference in optimal and desired = 0.60
Pressure at $y_{thickL}$ (Pa)	100494.52	98662.17	98661.28	Difference in initial and desired = 1833.24 Difference in optimal and desired = 0.89

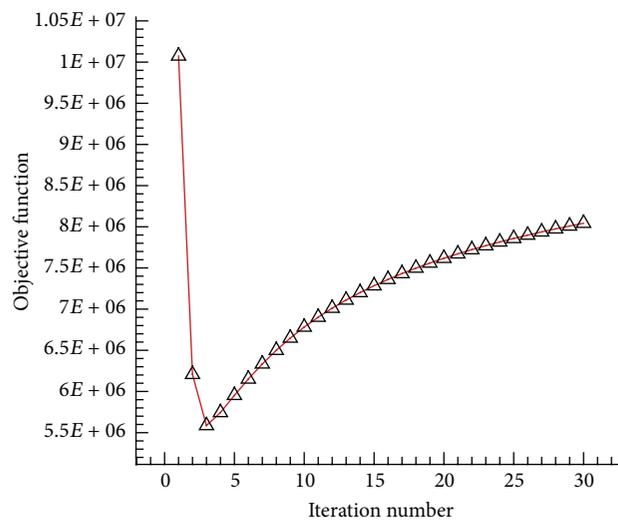


FIGURE 23: Objective function value versus the iteration number.

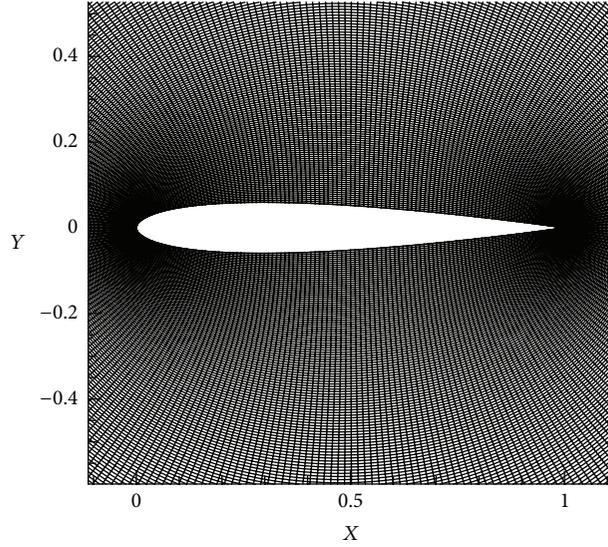


FIGURE 24: Grid used in Test Case 3 (around initial shape).

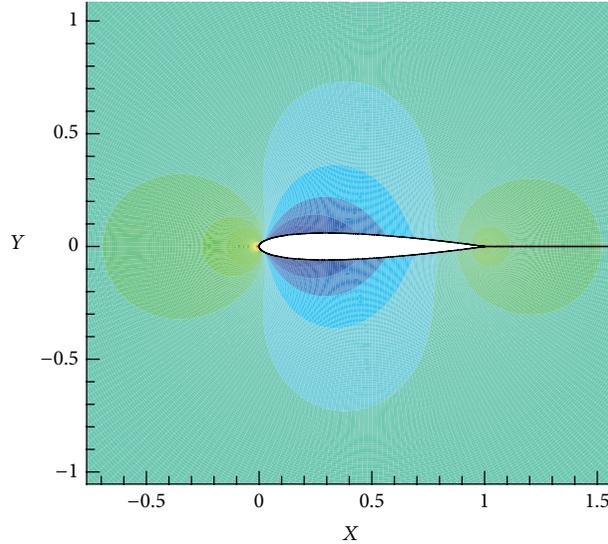


FIGURE 25: Pressure distribution around the airfoil surface (initial shape).

The objective function  $\mathcal{J}$  and the governing equation  $\mathcal{R} = 0$  depend on the flow variables  $\mathbf{W}$  and the geometry design variable  $\mathbf{X}_D$ :

$$\begin{aligned} \mathcal{J} &= \mathcal{J}(\mathbf{W}, \mathbf{X}_D), \\ \mathcal{R} &= \mathcal{R}(\mathbf{W}, \mathbf{X}_D) = 0. \end{aligned} \tag{50}$$

The derivative of the objective function  $\mathcal{J}$  with respect to the design variables  $\mathbf{X}_D$  can be expressed as

$$\frac{d\mathcal{J}}{d\mathbf{X}_D} = \frac{\partial \mathcal{J}}{\partial \mathbf{X}_D} + \frac{\partial \mathcal{J}}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial \mathbf{X}_D} \tag{51}$$

which states that a change in the objective function is due to a combination of a variation in the flow solution  $\partial \mathbf{W}$  and

a variation in the design variable (change in geometry)  $\partial \mathbf{X}_D$ . In a similar way, we have

$$\frac{d\mathcal{R}}{d\mathbf{X}_D} = \frac{\partial \mathcal{R}}{\partial \mathbf{X}_D} + \frac{\partial \mathcal{R}}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial \mathbf{X}_D} = 0. \tag{52}$$

If the sensitivity analysis is performed using (51) and (52), the problem is referred to as the “primal problem.” Solving the primal problem comes with the same difficulties as we encounter with use of the finite difference method. It requires the additional flow solutions proportional to the number of the design variables  $\mathbf{X}_D$ . Therefore, the adjoint method comes to the picture by introducing a vector of Lagrange multipliers

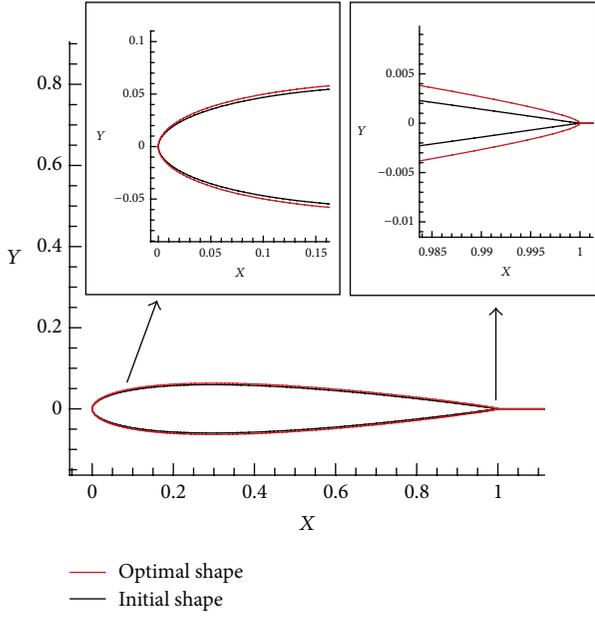


FIGURE 26: Comparison of the initial and optimal shapes and some magnified parts including the leading and trailing edges.

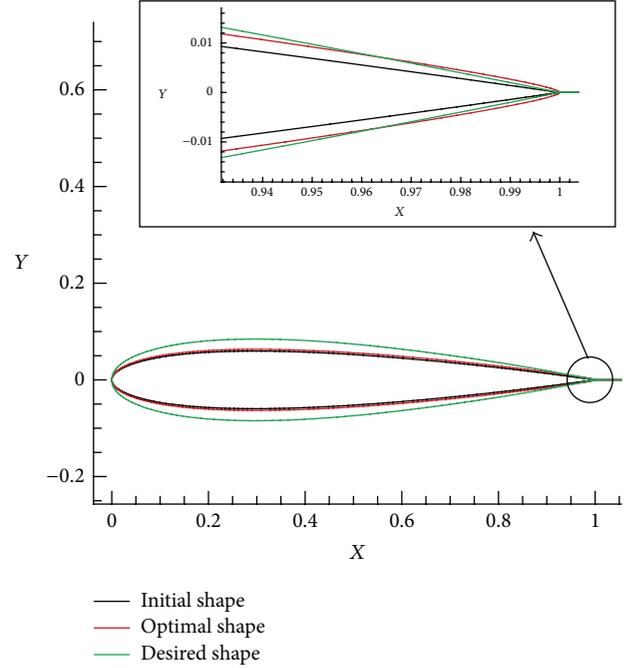


FIGURE 28: Comparison of the initial, optimal, and desired shapes.

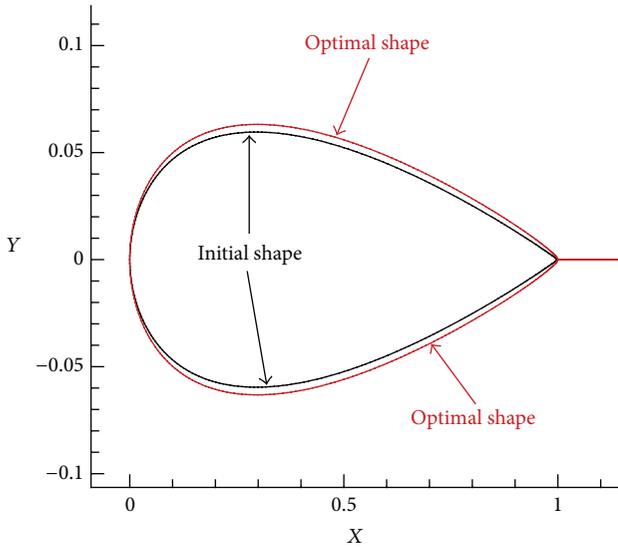


FIGURE 27: Comparison of the initial and optimal shapes. The y-axis has been greatly exaggerated to highlight difference in the airfoil shapes.

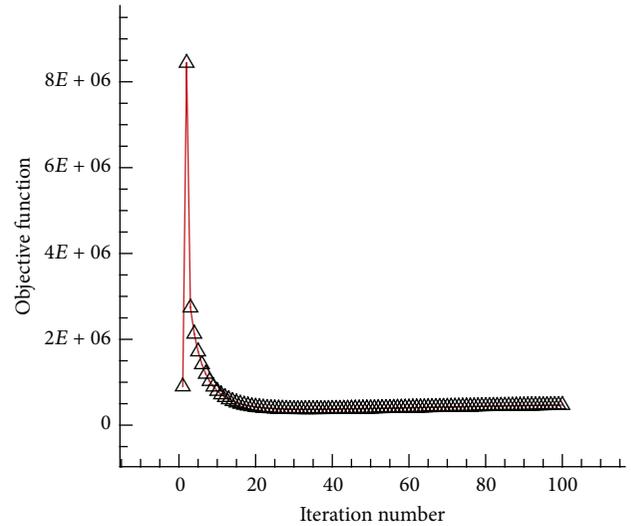


FIGURE 29: Objective function value versus the iteration number.

$\Psi$ . By adding (52) as a constraint to the sensitivity equation (51), we obtain

$$\frac{d\mathcal{F}}{d\mathbf{X}_D} = \frac{\partial \mathcal{F}}{\partial \mathbf{X}_D} + \frac{\partial \mathcal{F}}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial \mathbf{X}_D} - \Psi^T \left\{ \frac{\partial \mathcal{R}}{\partial \mathbf{X}_D} + \frac{\overset{=0}{\partial \mathcal{R}}}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial \mathbf{X}_D} \right\}. \quad (53)$$

Rearranging the terms inside (53), we get

$$\frac{d\mathcal{F}}{d\mathbf{X}_D} = \left( \frac{\partial \mathcal{F}}{\partial \mathbf{X}_D} - \Psi^T \frac{\partial \mathcal{R}}{\partial \mathbf{X}_D} \right) + \left( \frac{\partial \mathcal{F}}{\partial \mathbf{W}} - \Psi^T \frac{\partial \mathcal{R}}{\partial \mathbf{W}} \right) \frac{\partial \mathbf{W}}{\partial \mathbf{X}_D}. \quad (54)$$

If

$$\frac{\partial \mathcal{F}}{\partial \mathbf{W}} - \Psi^T \frac{\partial \mathcal{R}}{\partial \mathbf{W}} = 0, \quad (55)$$

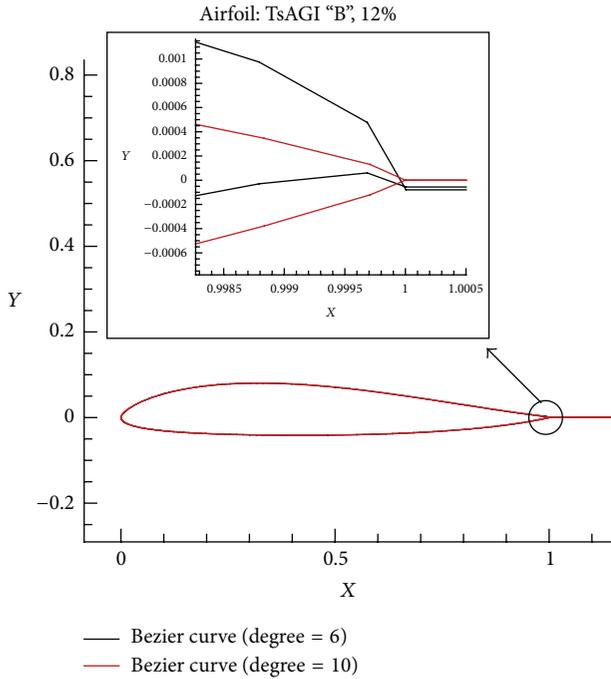


FIGURE 30: Oscillations around the trailing edge.

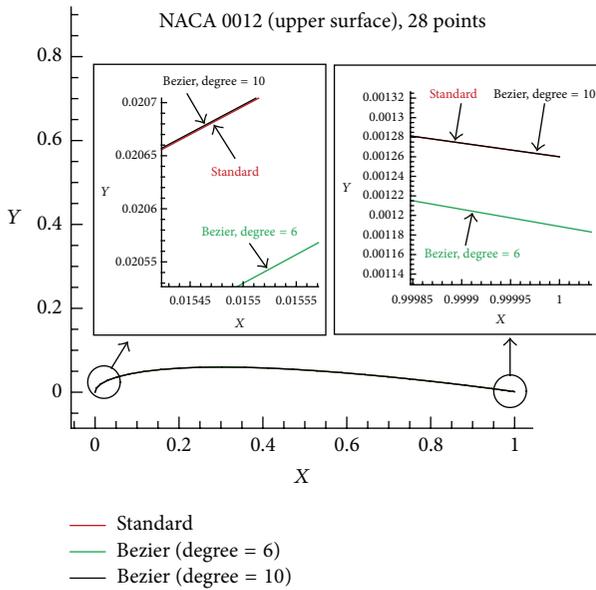


FIGURE 31: Comparison of an analytical NACA 0012 airfoil (upper surface only) with one obtained by using the Bezier curves of orders 7 and 11. The plots represent an excellent agreement between the analytical NACA and the Bezier of order 11.

then (54) reduces to

$$\frac{d\mathcal{J}}{d\mathbf{X}_D} = \frac{\partial \mathcal{J}}{\partial \mathbf{X}_D} - \Psi^T \frac{\partial \mathcal{R}}{\partial \mathbf{X}_D}. \quad (56)$$

Equation (55) is the *adjoint equation* and the vector  $\Psi$  is the *adjoint variables*. Equations (55) and (56) are referred to as the “dual problem.” The adjoint equation is a linear system

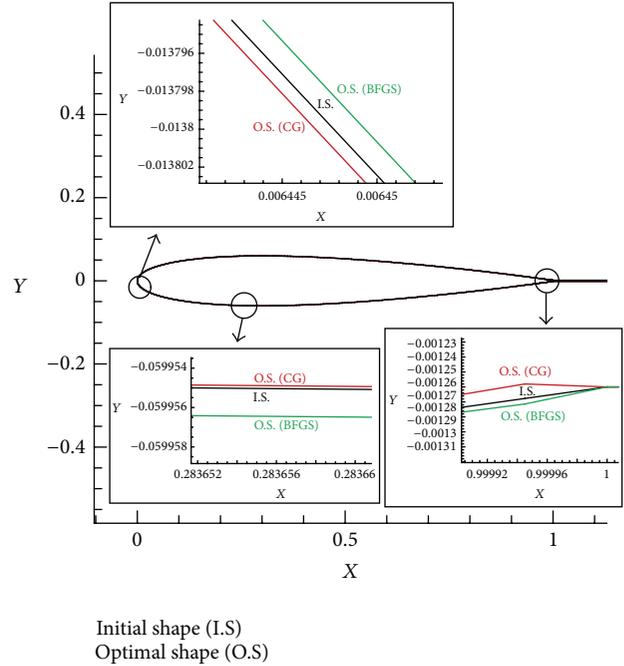


FIGURE 32: Comparison of the initial and optimal airfoil shapes with the magnified sections of them to show the variation of the shape. Both BFGS and CG are used in optimization process.

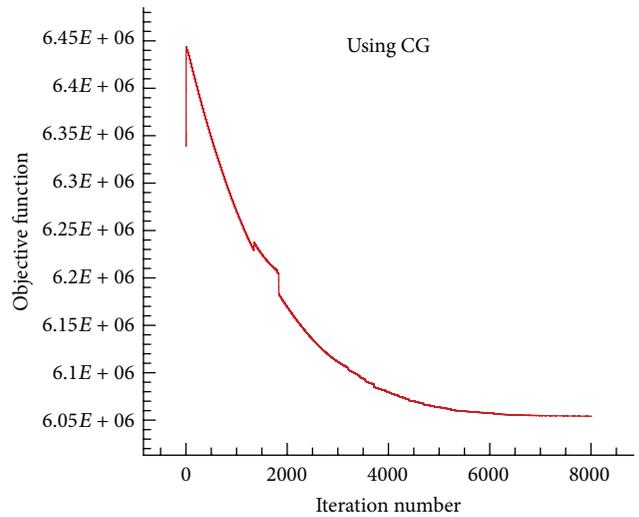


FIGURE 33: Convergence of the objective function. CG is used as the optimization method.

and can be solved to obtain  $\Psi$ . Then the determined  $\Psi$  can be substituted into (56) to obtain the gradient of the objective function. It can be seen that the gradient of the objective function can be determined without the need for additional flow solutions. The computational cost of solving the adjoint equation is comparable to that of solving the flow equation. Therefore, the computational cost of evaluating the objective function gradient is roughly equal to the computational cost of two flow equation solutions, independent of the number of design variables [38–41].

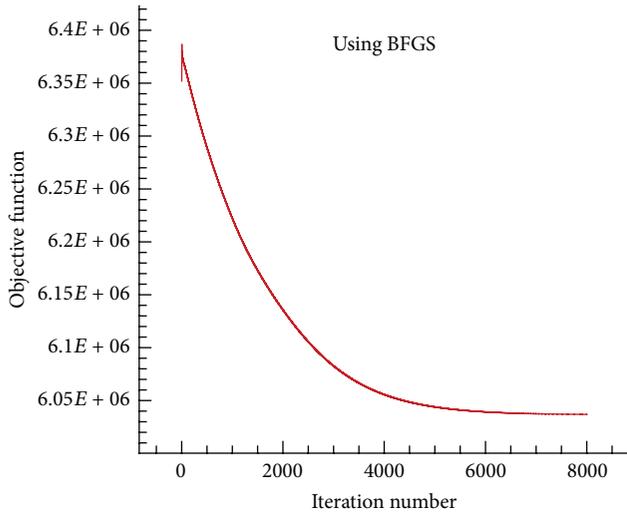


FIGURE 34: Convergence of the objective function. BFGS is used as the optimization method.

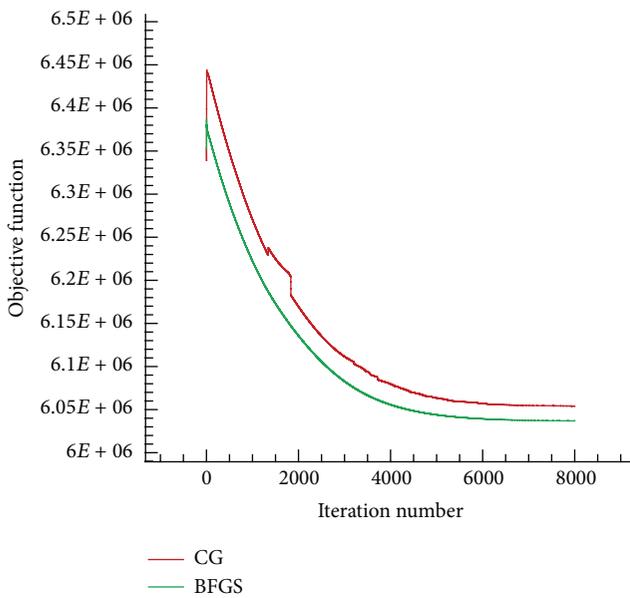


FIGURE 35: Comparison of the CG and BFGS methods in decreasing the objective function.

From the accuracy of the derivatives view point, the finite difference method (based on the perturbation scheme) is compared to the adjoint method [42–44]. The comparison shows a very good agreement between two methods. Therefore, our aim here is to compare our novel shape sensitivity method to the adjoint method from the efficiency view point only. As mentioned above, the computational cost of solving the adjoint equation is comparable to that of solving the flow equation whereas the computational cost of our novel method is comparable to that of computation of an algebraic expression for arrays of a matrix. As seen in Test Case 4, the computation time for iterations 2 to 8000 is 46 – 25 =

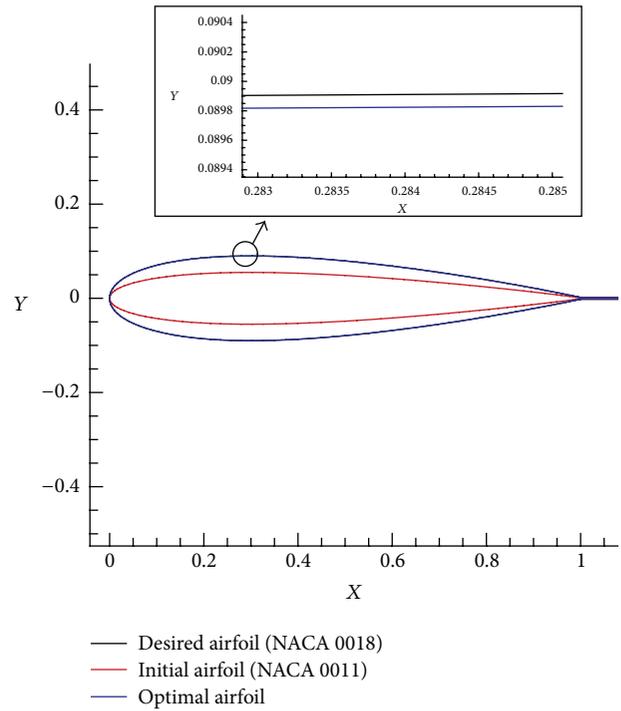


FIGURE 36: The initial, optimal, and desired shapes for the airfoil. There is an excellent agreement between the optimal and desired airfoil shapes.

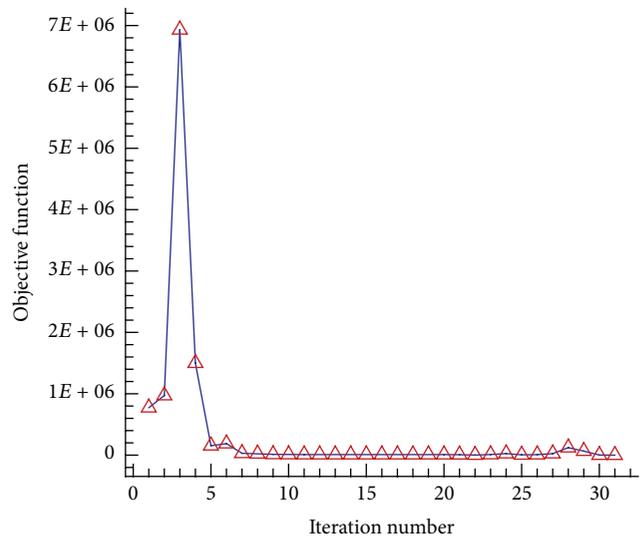


FIGURE 37: Objective function value versus the iteration number.

21 minutes (about 7 iterations per second) which reveals the efficiency of the proposed sensitivity analysis.

### 10. Conclusion

This paper addressed the aerodynamic shape optimization for an airfoil in an irrotational and incompressible flow governed

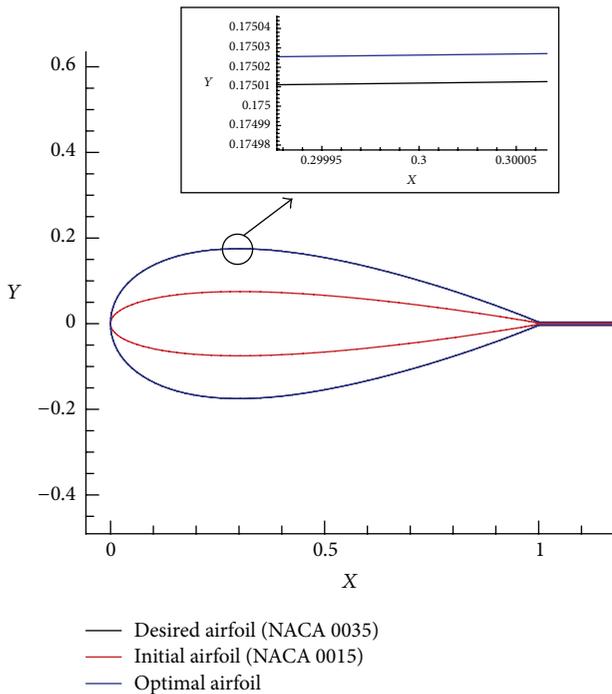


FIGURE 38: The initial, optimal, and desired shapes for the airfoil. There is an excellent agreement between the optimal and desired airfoil shapes.

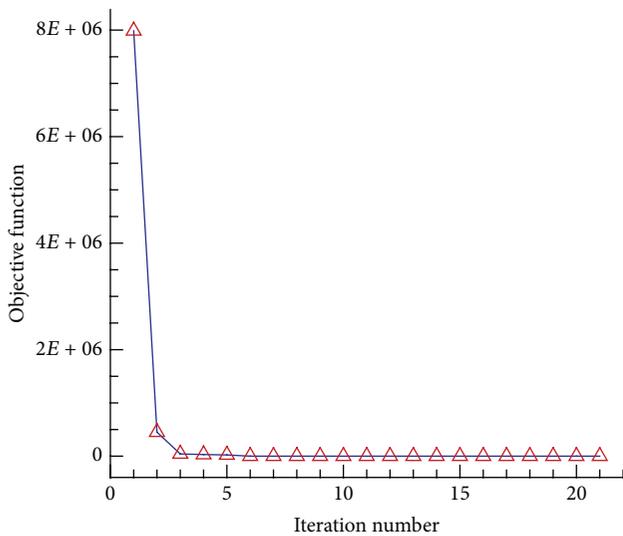


FIGURE 39: Objective function value versus the iteration number.

by Laplace equation using a type of the elliptic grid generation (O-type), a novel and very efficient sensitivity analysis method, and the conjugate gradient and BFGS optimization methods. The airfoil was parameterized using the grid points and the Bezier curve. Three different types of design variable were considered: the grid points, the Bezier curve control points, and the maximum thickness of NACA00xx airfoils. It was represented that the use of the Bezier curve significantly improves the optimization performance to reach the optimal

shape. The results obtained in test cases presented in this paper show that the proposed sensitivity analysis method reduces the computation cost even for large number of the design variables (Test Case 4) and confirm accuracy and efficiency of the proposed shape optimization algorithm.

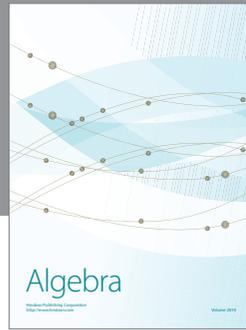
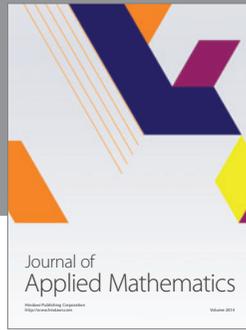
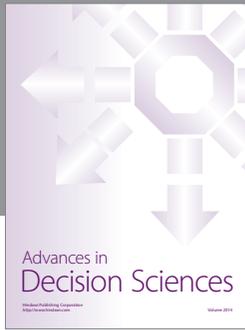
### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### References

- [1] R. M. Hicks, E. M. Murman, and G. N. Vanderplaats, *An Assessment of Airfoil Design by Numerical Optimization*, 1974.
- [2] R. M. Hicks and P. A. Henne, "Wing design by numerical optimization," *Journal of Aircraft*, vol. 15, no. 7, pp. 407–412, 1978.
- [3] O. Pironneau, "On optimum design in fluid mechanics," *Journal of Fluid Mechanics*, vol. 64, pp. 97–110, 1974.
- [4] A. Jameson, "Aerodynamic design via control theory," *Journal of Scientific Computing*, vol. 3, pp. 233–260, 1988.
- [5] A. Jameson, "Computational aerodynamics for aircraft design," *Science*, vol. 245, no. 4916, pp. 361–371, 1989.
- [6] A. Jameson, "Optimum aerodynamic design using CFD and control theory," AIAA Paper 95-1729, 1995.
- [7] A. Jameson and J. Reuther, *Control Theory Based Airfoil Design Using the Euler Equations*, Research Institute for Advanced Computer Science, NASA Ames Research Center, 1994.
- [8] M. B. Giles and N. A. Pierce, "Adjoint equations in CFD: duality, boundary conditions and solution behaviour," *AIAA Paper*, vol. 97, p. 1850, 1997.
- [9] M. B. Giles and N. A. Pierce, "On the properties of solutions of the adjoint Euler equations," *Numerical Methods for Fluid Dynamics*, pp. 1–16, 1998.
- [10] M. B. Giles, *Discrete Adjoint Approximations with Shocks*, Springer, New York, NY, USA, 2003.
- [11] M. B. Giles, M. C. Duta, J. Müller, and N. A. Pierce, "Algorithm developments for discrete adjoint methods," *AIAA Journal*, vol. 41, no. 2, pp. 198–205, 2003.
- [12] O. Baysal and M. E. Eleshaky, "Aerodynamic design optimization using sensitivity analysis and computational fluid dynamics," *AIAA Journal*, vol. 30, no. 3, pp. 718–725, 1992.
- [13] S. Ta'asan, G. Kuruvila, and M. Salas, "Aerodynamic design and optimization in one shot," in *Proceedings of the 30th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nev, USA, 1992.
- [14] H. Cabuk, C.-H. Sung, and V. Modi, "Adjoint operator approach to shape design for internal incompressible flows," in *Proceedings of the 3rd International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-3 '91)*, pp. 391–404, 1991.
- [15] M. Desai and K. Ito, "Optimal controls of Navier-Stokes equations," *SIAM Journal on Control and Optimization*, vol. 32, no. 5, pp. 1428–1446, 1994.
- [16] J. Elliott and J. Peraire, "Aerodynamic design using unstructured meshes," AIAA Paper, 1996.
- [17] J. Elliott and J. Peraire, "Aerodynamic optimization on unstructured meshes with viscous effects," *AIAA Paper*, vol. 97, p. 1849, 1997.
- [18] J. Elliott and J. Peraire, "Practical three-dimensional aerodynamic design and optimization using unstructured meshes," *AIAA Journal*, vol. 35, no. 9, pp. 1479–1485, 1997.

- [19] W. K. Anderson and V. Venkatakrishnan, "Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation," *Computers and Fluids*, vol. 28, no. 4-5, pp. 443–480, 1999.
- [20] L. Gonzalez, E. Whitney, K. Srinivas, and J. Périaux, "Multidisciplinary aircraft design and optimisation using a robust evolutionary technique with variable Fidelity models," in *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, vol. 4625, 2004.
- [21] I. C. Parmee and A. H. Watson, "Preliminary airframe design using co-evolutionary multiobjective genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1657–1665, 1999.
- [22] S. Obayashi, "Multidisciplinary design optimization of aircraft wing planform based on evolutionary algorithms," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3148–3153, October 1998.
- [23] A. Oyama, M.-S. Liou, and S. Obayashi, "Transonic axial-flow blade shape optimization using evolutionary algorithm and three-dimensional Navier-Stokes solver," in *Proceedings of the 9th AIAA/ISSMO Symposium and Exhibit on Multidisciplinary Analysis and Optimization*, Atlanta, Ga, USA, 2002.
- [24] H.-S. Chung, S. Choi, and J. J. Alonso, "Supersonic business jet design using knowledge-based genetic algorithm with adaptive, unstructured grid methodology," in *Proceedings of the 21st Applied Aerodynamics Conference*, 2003.
- [25] A. Jameson and K. Ou, "Optimization methods in computational fluid dynamics," in *Encyclopedia of Aerospace Engineering*, John Wiley & Sons, New York, NY, USA, 2010.
- [26] D. Thévenin and G. Janiga, *Optimization and Computational Fluid Dynamics*, Springer, 2008.
- [27] B. Mohammadi and O. Pironneau, *Applied Shape Optimization for Fluids*, Oxford University Press, Oxford, UK, 2009.
- [28] P. Castonguay and S. K. Nadarajah, "Effect of shape parameterization on aerodynamic shape optimization," in *Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit*, pp. 8–11, January 2007.
- [29] F. Mohebbi and M. Sellier, "Optimal shape design in heat transfer based on body-fitted grid generation," *International Journal for Computational Methods in Engineering Science and Mechanics*, vol. 14, no. 3, pp. 227–243, 2013.
- [30] F. Mohebbi and M. Sellier, "Three-dimensional optimal shape design in heat transfer based on body-fitted grid generation," *International Journal for Computational Methods in Engineering Science and Mechanics*, vol. 14, no. 6, pp. 473–490, 2013.
- [31] J. F. Thompson, F. C. Thames, and C. W. Mastin, "Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies," *Journal of Computational Physics*, vol. 15, no. 3, pp. 299–319, 1974.
- [32] M. W. Kutta, *Lifting Forces in Flowing Fluids*, 1902.
- [33] F. Mohebbi and M. Sellier, "On the Kutta condition in potential flow over airfoil," *Journal of Aerodynamics*, vol. 2014, Article ID 676912, 10 pages, 2014.
- [34] J. D. Anderson, *Fundamentals of Aerodynamics*, McGraw-Hill, 2001.
- [35] J. A. Samareh, "A survey of shape parameterization techniques," in *NASA Conference Publication*, pp. 333–344, Citeseer, 1999.
- [36] D. F. Rogers, *An Introduction to NURBS: With Historical Perspective*, Morgan Kaufmann Publications, 2001.
- [37] M. Hepperle, "Javafoil—analysis of airfoils," 2008, <http://www.mh-aerotoools.de/airfoils/javafoil.htm>.
- [38] A. Jameson, "Aerodynamic shape optimization using the adjoint method," Lectures at the Von Karman Institute, Brussels, Belgium, 2003.
- [39] A. Jameson, L. Martinelli, and N. A. Pierce, "Optimum aerodynamic design using the Navier-Stokes equations," *Theoretical and Computational Fluid Dynamics*, vol. 10, no. 1–4, pp. 213–237, 1998.
- [40] M. H. Straathof, *Shape parameterization in aircraft design: a Novel method, based on B-splines [Dissertation]*, Delft University of Technology, 2012.
- [41] M. B. Giles and N. A. Pierce, "An introduction to the adjoint approach to design," *Flow, Turbulence and Combustion*, vol. 65, no. 3-4, pp. 393–415, 2000.
- [42] W. K. Anderson and D. L. Bonhaus, "Airfoil design on unstructured grids for turbulent flows," *AIAA Journal*, vol. 37, no. 2, pp. 185–191, 1999.
- [43] T. D. Economou, P. Francisco, and J. J. Alonso, "Optimal shape design for open rotor blades," in *Proceedings of the 30th AIAA Applied Aerodynamics Conference*, pp. 1414–1436, June 2012.
- [44] S. K. Nadarajah, *The Discrete Adjoint Approach to Aerodynamic Shape Optimization*, Citeseer, 2003.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

