

An inverse approach for airfoil design

M. T. Rahmati¹, G. A. Aggidis¹ & M. Zangeneh²

¹Lancaster University, Engineering Department, UK

²University College London, Mechanical Engineering Department, UK

Abstract

Inverse design methods directly compute geometry for specified design parameters such as surface pressure or velocity, which is related to the performance of an airfoil (or a blade) geometry. These methods replace the time consuming iterative procedure of direct methods in which a large number of different blade shapes are designed and analysed to find the one which creates the surface velocity or pressure distribution closest to the desired one. In this paper a viscous inverse method for airfoil design is described. The inverse design approach computes an airfoil shape based on the target surface pressure distribution. The re-design of an airfoil, starting from an initial arbitrary profile in subsonic flow regimes, demonstrates the merits and robustness of this approach.

Keywords: inverse method, CFD, airfoil design, RANS equations.

1 Introduction

The Computational Fluid Dynamics (CFD) codes can be directly used for airfoil shape design based on ‘trial and error’ approaches. By guessing an airfoil shape, the flow solution can be obtained using the CFD codes. The flow solution is then compared with the desired flow conditions. If these are not met then the airfoil geometry is altered. The whole process is repeated again, until the required flow conditions are achieved. In fact, these direct design procedures are very inefficient and time consuming. In order to reduce the development and design time and their associated costs, a more systematic method is required. The inverse method is an alternative approach that replaces the time consuming iterative procedure of direct methods. A pioneering airfoil inverse method based on conformal mapping was developed by Lighthill [1] in 1945. Since then many inverse methods for airfoil (or blade) design have been developed. These



methods are mainly based on potential flow equations [2, 3] or Euler flow equations [4, 5]. These methods provide inviscid geometries so many characteristics of real flow fields are ignored. Unfortunately the inverse design based on inviscid flow calculations cannot directly be extended to methods based on Navier-Stokes equations. This is because the blade modification algorithm of these inverse design methods requires a non-zero relative velocity on the surface whereas the relative wall velocity is zero in viscous flow calculations due to the non-slip condition. However, in certain flow fields accurate modelling of viscous flow by utilizing Navier-Stokes equations is essential in order to design the blade or airfoil shape more precisely [6].

The current viscous inverse design approach computes an airfoil shape based on the target surface pressure distribution. In order to determine the required geometrical modification to accomplish the target design specification the following steps are carried out. First, the surface pressure coefficients distribution of an initial blade is calculated using the viscous flow analysis algorithm. Then, the difference between the target and the initial surface pressure coefficient distribution is used for blade modification. Subsequently, the mesh of the domain is adapted to the modified blade. The modified airfoil is then considered as an initial airfoil and this procedure is carried out iteratively until the differences between the target and initial surface pressure coefficients are negligible. These steps and the application of the method for the design of an airfoil are described hereafter. It is then followed by the application of the method for airfoil shape design.

2 Flow analysis algorithm

Taking the time average of the steady incompressible Navier-Stokes equations in a Cartesian coordinate system and applying the Boussinesq hypothesis for closure of the equations yields the following RANS equations:

$$\frac{\partial}{\partial x_i}(\rho U_i) = 0 \quad (1)$$

$$\frac{\partial}{\partial x_j} \left[\rho U_i U_j - (\mu + \mu_t) \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \right] = -\frac{\partial P}{\partial x_i} \quad (2)$$

U and P are the time averaged components of velocity and pressure vectors respectively, μ is viscosity, μ_t is turbulent viscosity and ρ is density. The standard k - ε model is implemented for turbulence modelling as it has a good compromise in terms of accuracy and robustness [6]. In this model μ_t is given by:

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \quad (3)$$

An approximate transport equation for k and ε can be written in the following form:

$$\frac{\partial \rho U_i k}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\mu_t}{\sigma_k} \frac{\partial k}{\partial x_i} \right) + G_k - \rho \varepsilon \quad (4)$$



$$\frac{\partial \rho U_i \varepsilon}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\mu_t}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial x_i} \right) + C_1 \frac{\varepsilon}{k} G_k - C_2 \rho \frac{\varepsilon^2}{k} \quad (5)$$

Here G_k represents the generation of k and is defined as

$$G_k = \mu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \frac{\partial U_i}{\partial x_j} \quad (6)$$

Based on experimental data from a variety of turbulent flows Launder and Spalding [7] recommended the following values for the empirical constants which appear in equation (3) to equation (5):

$$C_1=1.44, C_2=1.92, C_\mu=.09, \sigma_k=1 \text{ and } \sigma_\varepsilon=1.3.$$

The pressure correction method developed by Patankar [8] has been used to solve the incompressible Navier-Stokes equations on unstructured meshes. A cell centred finite volume discretisation of the governing equation based on the work of Mathur and Murthy [9] was developed. The flow analysis algorithm and discretisation method is explained in details in Rahmati et al. [10].

3 Airfoil modifying algorithm

The blade surface pressure coefficient instead of surface velocity is used to modify the blade. This is because the velocity on the surface is considered zero to satisfy the non-slip conditions. The airfoil surface is treated as elastic membrane, which is modified based on the differences between target and calculated pressure coefficient. The airfoil modifying algorithm is based on the method of Dulikravich and Baker [11] who suggested the following linear partial differential equations can be used as a residual corrector to modify the top and bottom contour of the blade respectively:

$$-\beta_1 \partial (\Delta y_{top}) + \beta_2 \partial (\Delta y_{top}) / \partial s + \beta_3 \partial^2 (\Delta y_{top}) / \partial s^2 = \Delta Cp \quad (7)$$

$$\beta_1 \partial (\Delta y_{bottom}) + \beta_2 \partial (\Delta y_{bottom}) / \partial s - \beta_3 \partial^2 (\Delta y_{bottom}) / \partial s^2 = \Delta Cp \quad (8)$$

In equation (7) and equation (8), s is the airfoil contour following coordinate, L_E is the lower airfoil contour length, L is the total airfoil counter length and Δy_{top} and Δy_{bottom} are the blade normal displacement at the top and bottom counter of the airfoil respectively. $\beta_1, \beta_2, \beta_3$ in equation (7) and equation (8) are user specified coefficients while ΔCp is the local differences between the target and computed surface pressure coefficient i.e.

$$\Delta Cp = C_p^{Target} - C_p^{Calculated} \quad (9)$$

By using two different equations, appropriate boundary conditions for the airfoil shape design such as leading edge closure and stacking conditions can be implemented. The solution of this equation provides the geometry correction, which is used to modify the initial geometry to form a new geometry. If, after having checked the convergence, the design requirements are not satisfied, the design cycle is repeated with the new geometry. The process is repeated until the pressure coefficient differences are negligible.



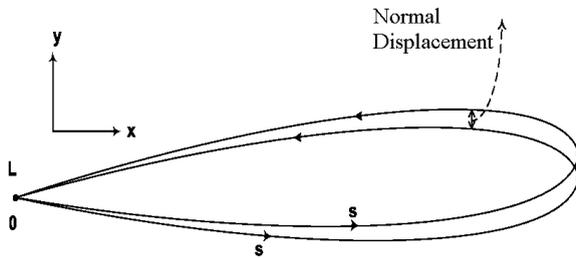


Figure 1: Blade modification method using elastic membrane concept.

Equation (7) and equation (8) are non-homogeneous second order equations with constant coefficients. The complementary functions of these equations for the upper and lower contours are:

$$\Delta y^{top} = C_1 e^{\lambda_1 s} + C_2 e^{\lambda_2 s} \tag{10}$$

$$\Delta y^{Bottom} = C_3 e^{-\lambda_1 s} + C_4 e^{-\lambda_2 s} \tag{11}$$

where

$$\lambda_{1,2} = \frac{-\beta_2 \pm \sqrt{\beta_2^2 + 4\beta_1 \cdot \beta_3}}{2\beta_3} \tag{12}$$

C_1, C_2, C_3 and C_4 are constants that will be computed from the boundary conditions at the leading and trailing edges. To find out the particular integral of equation (7) and equation (8) the surface distribution of ΔC_p is represented by utilising the Fourier series expansion as follow:

$$\Delta C_p(s) = a_0 + \sum_{n=1}^{n_{max}} [a_n \cos(N_n s) + b_n \sin(N_n s)] \tag{13}$$

where $N_n = 2n\pi/L$ and a_0, a_n and b_n are the Fourier series coefficients. The particular integral of equation (7) and equation (8) are represented in the form of Fourier series as:

$$\Delta y^{top} = A_0^{top} + \sum_{n=1}^{n_{max}} [A_n^{top} \cos(N_n s) + B_n^{top} \sin(N_n s)] \tag{14}$$

$$\Delta y^{bottom} = A_0^{bottom} + \sum_{n=1}^{n_{max}} [A_n^{bottom} \cos(N_n s) + B_n^{bottom} \sin(N_n s)] \tag{15}$$

The values of A_0, A_n and B_n are found by substitution of equation (14) and equation (15) into equation (7) and equation (8) respectively: These values are as follow:

$$A_n^{top} = \frac{a_n(\beta_1 + N_n^2 \beta_3) - b_n(\beta_2 N_n)}{(\beta_1 + N_n^2 \beta_3) + (\beta_2 N_n)^2} \quad 0 \leq n \leq n_{max} \tag{16}$$

$$B_n^{top} = \frac{b_n(\beta_1 + N_n^2\beta_3) + a_n(\beta_2 N_n)}{(\beta_1 + N_n^2\beta_3) + (\beta_2 N_n)^2} \quad 1 \leq n \leq n_{max} \quad (17)$$

$$A_n^{bottom} = -\frac{a_n(\beta_1 + N_n^2\beta_3) + b_n(\beta_2 N_n)}{(\beta_1 + N_n^2\beta_3) + (\beta_2 N_n)^2} \quad 0 \leq n \leq n_{max} \quad (18)$$

$$B_n^{bottom} = -\frac{b_n(\beta_1 + N_n^2\beta_3) - a_n(\beta_2 N_n)}{(\beta_1 + N_n^2\beta_3) + (\beta_2 N_n)^2} \quad 1 \leq n \leq n_{max} \quad (19)$$

Having found the particular integral and complementary functions of equations (7) and equation (8), the complete solution of these equations are given by the following equations:

$$\Delta y^{top} = C_1 e^{\lambda_1 s} + C_2 e^{\lambda_2 s} + \sum_{n=0}^{n_{max}} [A_n^{top} \cos(N_n s) + B_n^{top} \sin(N_n s)] \quad (20)$$

$$\Delta y^{bottom} = C_3 e^{-\lambda_1 s} + C_4 e^{-\lambda_2 s} + \sum_{n=0}^{n_{max}} [A_n^{bottom} \cos(N_n s) + B_n^{bottom} \sin(N_n s)] \quad (21)$$

Equation (20) and equation (21) contain four unknown constants C_1 , C_2 , C_3 and C_4 . To compute these constants, the following boundary conditions are applied: zero trailing edge displacement, trailing edge closure, leading edge closure, and smooth leading edge. Consequently, these boundary conditions, for the airfoil shown in figure 1, yield four equations with four unknown: C_1 , C_2 , C_3 and C_4 . These constants are found by the following matrix:

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & e^{-L\lambda_1} & e^{L\lambda_2} \\ e^{-\lambda_1 L_E} & e^{-\lambda_2 L_E} & -e^{\lambda_1 L_E} & -e^{\lambda_2 L_E} \\ -\lambda_1 e^{-\lambda_1 L_E} & -\lambda_2 e^{-\lambda_2 L_E} & -\lambda_1 e^{\lambda_1 L_E} & -\lambda_2 e^{\lambda_2 L_E} \end{bmatrix}^{-1} \times \begin{bmatrix} -\sum_{n=1}^{n_{max}} A_n^{bottom} \\ -\sum_{n=1}^{n_{max}} A_n^{top} \\ \sum_{n=1}^{n_{max}} [\Delta A_n \cos(N_n L_E) + \Delta B_n \sin(N_n L_E)] \\ \sum_{n=1}^{n_{max}} [-\Delta A_n N_n \sin(N_n L_E) + \Delta B_n N_n \cos(N_n L_E)] \end{bmatrix} \quad (22)$$

where

$$\Delta A_n = A_n^{top} - A_n^{bottom} \quad (23)$$

$$\Delta B_n = B_n^{top} - B_n^{bottom} \quad (24)$$



In the formulation of this method three arbitrary contents; β_1 , β_2 , β_3 are used. Ideally the design method for different applications should be independent of these user-specified constants. However, the choice of these arbitrary constants controls the convergence and stability of the inverse design. So optimising these values for successful convergence of iterative design procedure in reasonable time is essential. Though, the only method to specify these user-specified values for different applications is based on trail and error. This is the major disadvantage of the method as the success of inverse design depends on the specification of these arbitrary constants. For airfoil design the amount of $\beta_1 = 1.2$, $\beta_2 = 0$, $\beta_3 = 0.4$ was considered which are the same values which was proposed by Dulikravich and Baker [11].

4 Mesh movement algorithm

Mesh movement algorithm is an integral part of the current inverse design method as once the blade surface is modified by inverse algorithm the corresponding triangular mesh has to be adapted too. The mesh movement algorithm is based on the linear tension spring analogy concept of Batina [12]. In this method each unstructured meshes edges are modeled as springs with the stiffness inversely proportional to the length of the edges. By displacement of the boundaries of the domain, the spring forces are calculated. Then the displacement of every interior vertex is solved iteratively until all forces are in equilibrium. A linear tension spring analogy is applied here because only the nodal displacements are important and no purely elasticity variables such as local stress or strains. Also this method is a robust method with a low computational cost, see Rahmati [6].

5 Inverse design application

The task of inverse design is usually to improve the performance of a known blade or create a new design based on earlier design that operates under the same conditions. Thus, the initial blade is usually known prior to the design. In the case of inverse design of airfoil, a known airfoil, NACA0012 with zero angle of incidence, is used as an initial blade. The blade modification algorithms of the inverse design method require an initial condition, which is given by a stacking point that remains fixed throughout the iterative design procedure. In this case the stacking point is a point at the trailing edge of the airfoil. This initial condition is required to satisfy the boundary conditions of equation (20) and equation (21).

The NACA0012 airfoil with zero angle of incidence is utilised as an initial blade to re-design of NACA0012 with 6 degree angle of incidence by imposing the target surface pressure coefficient distribution. The air flow velocity is 55 m/s. Figure 2 and figure 3 show the initial airfoil, the airfoil shape after five and ten iterations, the designed airfoil and their corresponding surface pressure coefficients. The meshes of these airfoils, which contain 10216 triangular cells, are shown in figure 4 to figure 7. Twenty five calls to the analysis code are



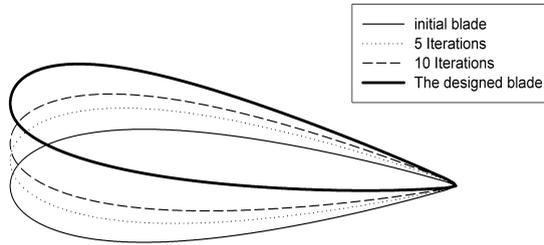


Figure 2: Reproducing NACA0012 airfoil with 6 degree angle of incidence.

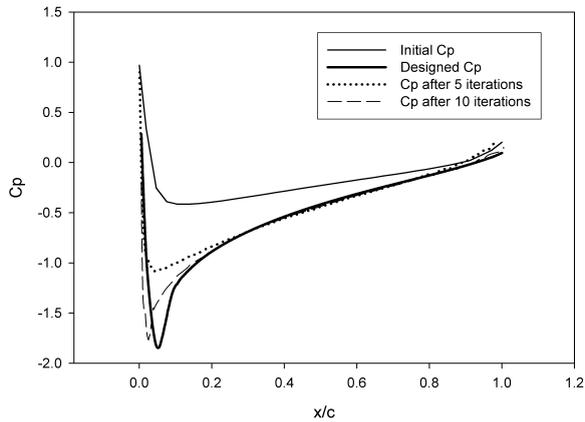


Figure 3: Initial, intermediates and target surface pressure coefficient.

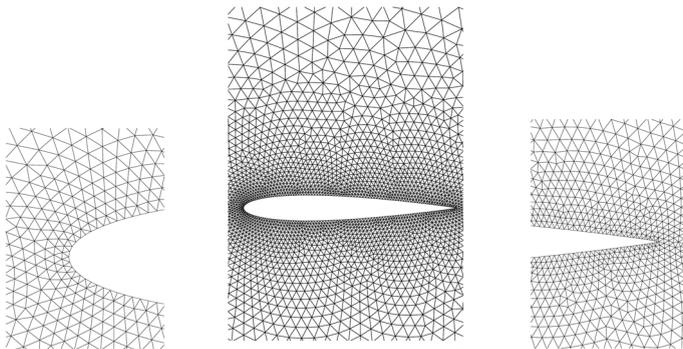


Figure 4: The initial mesh around the blade.

required for the convergence of the geometry. The flow analysis algorithm is considered to be converged when the normalised residuals of the governing equations had reduced to 10^{-4} . The design process for the computation of blade geometry is considered to have converged when the calculated design parameters

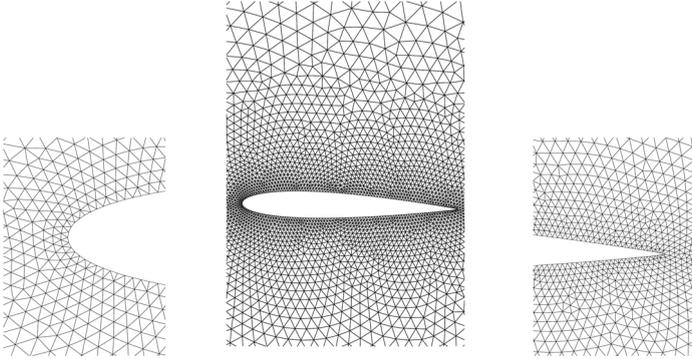


Figure 5: The mesh around the blade after 5 iterations.

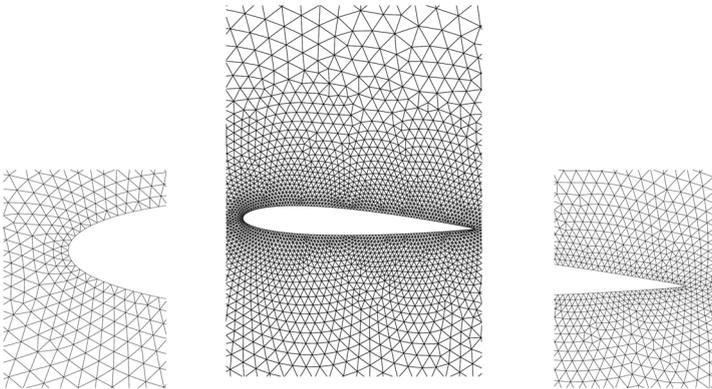


Figure 6: The mesh around the blade after 10 iterations.

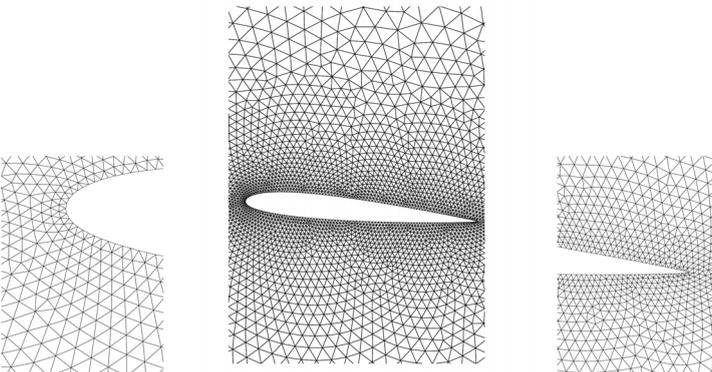


Figure 7: The mesh around the designed blade.

match the target design parameter within the specific tolerance of 1%. This tolerance is defined as the percentage of the discrepancy between the target and calculated design specification.

6 Conclusion

A Navier-Stokes inverse design method based on the specification of surface pressure coefficient distribution on the blade is developed. One of the improvements of the present methods over previous methods is that the flow field is treated as viscous turbulent flow. Also one of the advantages of the current design method and flow analysis algorithm is that they are based on the unstructured meshes. So most complex fluid regions can be meshed automatically which yields to significant reduction in the time and effort required to generate meshes. The capabilities of this design methodology have been verified by reproducing an airfoil based on a target surface pressure coefficient distribution. The numerical results show the efficiency of the method for airfoil inverse design. However the convergence rate of the shape design process relies on the specifications of three arbitrary constants. These arbitrary constants are required in the formulation of the blade modifying algorithm. The method can be extended for three-dimensional inverse design of wings by rewriting equation (7) and equation (8) for three-dimensional surfaces using more arbitrary constants. So, the convergence rate of the shape design process for three-dimensional applications will depend on the specifications of more than three arbitrary constants.

References

- [1] Lighthill, M.J., A New Method of Two Dimensional Aerodynamics Design, *Aeronautical Research Council*, London R& M, 2112, 1945.
- [2] Sobieczky, H., Fung, N.J., & Seebass, R.S., New Method for Designing Shock-Free Transonic Configuration, *AIAA Journal*, (17): pp. 722–729, 1978.
- [3] Zangeneh, M., A Compressible Three-Dimensional Design Method for Radial and Mixed Flow Turbomachinery Blades, *International Journal of Numerical Methods in Fluids*, pp. 599–624, 1991.
- [4] Demeulenaere, A.R., & Braembussche, V.D, Three-Dimensional Inverse Method for Turbomachinery Blading Design, *Trans. ASME, Journal of Turbomachinery*, (120), pp. 247–255, 1998.
- [5] Dang, T., & Jiang, J., Design Method for Turbomachine Blades With Finite Thickness By The Circulation Method, *Journal of Turbomachinery*, (119), 1997.
- [6] Rahmati, M.T., Incompressible Navier-Stokes Inverse Design Method Based on Unstructured Meshes, PhD *thesis*, University College London, 2006.
- [7] Launder, B.E., & Spalding, D.B., Lectures in Mathematical Models of Turbulence, *Academic Press*, London, 1972.



- [8] Patankar, S.V., Numerical Heat Transfer and Fluid Flow, *Taylor and Francis*, U.S.A, 1980.
- [9] Mathur, S.R., & Murthy, Y.T., A Pressure-Based Method for Unstructured Meshes, *Numerical Heat Transfer*, Part B. 31, pp. 195–215, 1997.
- [10] Rahmati, M.T., Charlesworth, D., & Zangeneh, M., Incompressible Navier-Stokes Inverse Design Method Based On Adaptive Unstructured Mesh, *The 13th Annual Conference on Computational Fluid Dynamics*, St. John's, Canada, 2005.
- [11] Dulikravich, G. S., & Baker, D.P., Fourier Series Solution for Inverse Design of Aerodynamic Shape, *Inverse Problems in Engineering*, pp. 427–436, 1998.
- [12] Batina, J.T., Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes, *AIAA Journal*, 28(8), pp. 1381–1388, 1990.

