AIAA JOURNAL
Vol. 47, No. 3, March 2009

Inverse Approach to Turbomachinery Blade Design

Mohammad T. Rahmati*
Lancaster University, Lancaster, England LA1 4YR, United Kingdom

DOI: 10.2514/1.38856

This paper presents a novel viscous inverse method for blade design. In this method, viscosity is directly taken into account using a flow analysis code based on the Navier–Stokes equations. In this approach, the blade pressure loading and the blade thickness are prescribed and the corresponding blade profile is sought. By specifying the target pressure loading and the blade thickness, the camber line instead of the blade surface deforms. The prescription of the blade thickness ensures that a realistic blade shape with closed leading and trailing edges can be obtained. The redesign of an axial rotor and a stator blade, starting from an initial arbitrary profile in subsonic flow regimes, demonstrates the merits and robustness of this approach.

Nomenclature

a_n, b_n	=	Fourier coefficients
c	=	blade chord
f	=	blade camber
k	=	turbulent kinetic energy
k_{ij}	=	spring stiffness coefficient of edge ij
L_{ij}	=	length of edge ij
n_i	=	total number of neighboring vertices
P	=	time-averaged component of pressure
p	=	pressure
p'	=	fluctuating component of pressure
q_1, q_2	=	constants in complementary functions for the upper contour of blade
q_3, q_4	=	constants in complementary functions for the lower contour of blade
$\mathbf{r}_i$	=	position vector of vertex i
s	=	blade contour following coordinate
t_θ	=	blade thickness
U	=	time-averaged component of velocity
u	=	velocity
u'	=	fluctuating component of velocity
x, y	=	Cartesian coordinates
α	=	blade surface
β_i	=	arbitrary parameters in blade modification formulation ($i = 1, 2, 3$)
ΔP	=	pressure loading
δ_i	=	displacement of vertex i
δp	=	difference between the target and initial pressure loading
ε	=	dissipation of turbulence kinetic energy
μ	=	viscosity
μ_t	=	turbulent viscosity
ρ	=	density

Subscripts

new	=	new calculated value
old	=	value at the previous iteration

Superscripts

$+$	=	upper contour of blade
$-$	=	lower contour of blade

I. Introduction

A LARGE number of robust and sophisticated flow analysis codes are now available for the simulation of flow through turbomachinery blades [1–3]. These codes can be used directly for blade design based on "trial and error" approaches. In these methods, the designer relies on his previous experience to modify the blade shape at a certain location to improve the flowfield there. However, these methods are very inefficient and time consuming. To reduce the development and design time and their associated costs, a more systematic method is required. These direct methods can be automated by coupling an automatic optimization algorithm, a CAD-based blade generator, and a computational fluid dynamics (CFD) code [4]. These methods are, however, very expensive in terms of the required CPU time, particularly when they are coupled with a flow analysis algorithm based on the Navier–Stokes equations. The inverse design method is an alternative method that defines the blade shape for a prescribed local flow property, such as surface pressure. The physical model is used to derive the geometrical changes required for achieving the prescribed local flow conditions [5,6]. Inverse design methods are usually based on potential flow equations [7,8] or Euler flow equations [9–11]. The methods ignore fluid viscosity, and so many characteristics of the real flowfields are neglected. However, in certain flowfields, accurate modelling of viscous flow using the Navier–Stokes equations is essential to design the blade or airfoil shape more precisely [5]. Unfortunately, these inviscid inverse design methods cannot be extended directly to a method based on the Navier–Stokes equations. This is because the blade modification algorithms of these inverse design methods require a nonzero relative velocity on the surface, whereas the relative wall velocity is zero in viscous flow calculations due to the nonslip condition.

An inverse design method based on the elastic membrane concept, which does not require the surface velocity for blade modification, was first proposed by Garabedian and McFadden [12]. The method can thus be used with the Navier–Stokes equations. This method was modified using a Fourier series formulation by Dulikravich and Baker [13]. In this approach, the surface of an aerodynamic body deforms under the aerodynamic load similar to an elastic membrane. The aerodynamic load is considered to be the local difference between the target and initial surface pressure coefficients. The elastic membrane under the aerodynamic load deforms smoothly until the calculated surface pressure coefficient matches the target surface pressure coefficient. Two linear partial differential equations are used as a residual corrector to modify the top and bottom contours of the airfoil. However, this method is based on the prescription of the surface pressure on the upper and lower contours of airfoils. The surface pressure distribution allows for good control over the flowfield but, in practical applications, defining a target pressure distribution on the blade that leads to a physically realistic blade, or any blade at all, cannot be guaranteed. In general, a blade that is designed using an inverse approach that is based on the specification

Received 31 May 2008; revision received 10 December 2008; accepted for publication 12 December 2008.

*Research Associate, Engineering Department.

of surface pressure on it can have physically nonrealistic geometries, such as an open leading edge or crossover trailing edge, as shown by [5]. Furthermore, the extension of these methods to the inverse design of three-dimensional blades is a challenge because the prescription of blade surface pressure in three-dimensional flows is extremely difficult. This is because the pressure distribution at the hub and shroud of the blade cannot be specified independently [11]. To produce realistic blades by using inverse design methods, another choice of target design specification instead of surface pressure distribution is required.

A new two-dimensional design approach based on this concept is developed for the inverse design of a blade based on the specification of pressure loading distribution and blade thickness. The pressure loading distribution is defined as the difference between the surface pressures on the two sides of the blade. By specifying target pressure loading, unlike in the previous method, the camber line instead of the blade surface deforms under the aerodynamics load. It smoothly deforms during design iterations until the calculated surface pressure loading matches the target surface pressure loading. An immediate consequence of using this method is that the prior determination of the blade thickness ensures that a realistic blade shape with closed leading and trailing edges can be obtained.

II. Inverse Design Method

The current inverse method consists of three integrated parts to determine the required geometrical modification to accomplish the target design specification. The first is the flow analysis algorithm, the second is the blade modifying algorithm, and the third is the mesh movement algorithm. The procedure involves coupling these three algorithms to compute the blade shape. The method assumes that the initial configuration of the turbomachinery, such as velocity input and rotational speed, have already been decided. The basic steps in the two-dimensional design procedure are given in Fig. 1.

First, the pressure loading of an initial blade is calculated using the flow analysis algorithm. Then, the difference between the target and the initial pressure loading distribution is used for blade modification. The modified blade is then considered as an initial blade, and this procedure is carried out iteratively until the differences between the target and initial pressure loadings are negligible. These steps and the application of the method to blade design are described below.

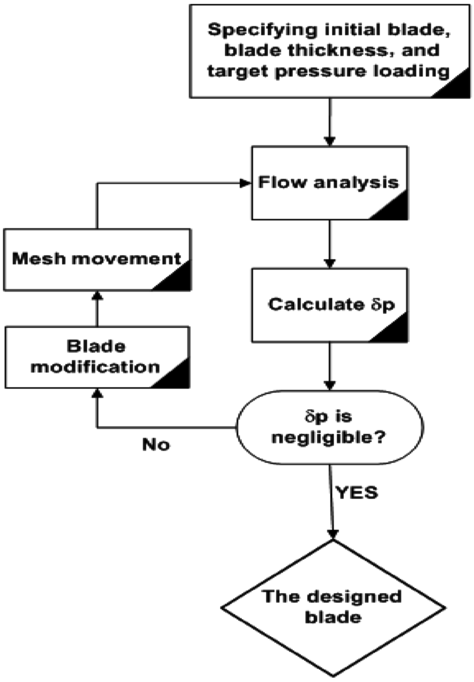

Fig. 1 Inverse design flowchart.

III. Flow Analysis Algorithm

For steady, incompressible viscous flow in a Cartesian coordinate system, the conservation laws of mass and momentum are described by the following equations:

$$\frac{\partial}{\partial x_i}(u_i) = 0 \tag{1}$$

$$\frac{\partial}{\partial x_j}\left[\rho u_i u_j - \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] = -\frac{\partial p}{\partial x_i} \tag{2}$$

The primitive variables are split into their fluctuating and time-averaged components, that is,

$$u = U + u' \tag{3}$$

$$p = P + p' \tag{4}$$

U and P are the time-averaged components, and u' and p' are the fluctuating components of velocity and pressure, respectively. Substituting Eqs. (3) and (4) into Eqs. (1) and (2), taking the time average of each equation, and applying the Boussinesq hypothesis for closure of the equations yields the following equations:

$$\frac{\partial}{\partial x_i} U_i = 0 \tag{5}$$

$$\frac{\partial}{\partial x_j}\left[\rho U_i U_j - (\mu + \mu_t)\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right] = -\frac{\partial P}{\partial x_i} \tag{6}$$

The standard k–ε model is implemented for turbulence modelling. This model is one of the most common turbulence models used in turbomachinery CFD codes because of its numerical stability. It is also a good compromise in terms of accuracy and robustness. In this model, μ_t is the turbulent viscosity and is given by:

$$\mu_t = \rho C_\mu (k^2/\varepsilon) \tag{7}$$

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

$$\frac{\partial \rho U_i k}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial x_i}\right) + G_k - \rho\varepsilon \tag{8}$$

$$\frac{\partial \rho U_i \varepsilon}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\frac{\mu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_i}\right) + C_1\frac{\varepsilon}{k}G_k - C_2\rho\frac{\varepsilon^2}{k} \tag{9}$$

Here G_k represents the generation of k and is defined as

$$G_k = \mu_t\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\frac{\partial U_i}{\partial x_j} \tag{10}$$

Based on experimental data from a variety of turbulent flows [14], the following values are recommended for the empirical constants that appear in Eqs. (8) and (9):

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

The flow analysis algorithm is based on the work of Charlesworth [15], who developed a Navier–Stokes flow analysis code in a stationary frame. In this algorithm, the pressure correction method developed by Patankar [16] has been used to solve the incompressible Navier–Stokes equations on unstructured meshes. A cell-centered finite volume discretization of the governing equation based on the work of Mathur and Murthy [17] was developed. The second-order discretization of the convective and diffusive terms in the governing equations is used. The pressure correction equation is solved using an adapted form of algebraic multigrid for unstructured meshes. Global normalized residuals are defined for the governing equations to allow a global convergence criterion to be specified. Such criteria are set to 10^{-5} for all discretized equations. The times taken for flow solution for various geometries on a Compaq (NUM) computer show that the code has a reasonable numerical computational cost in comparison with other codes [15]. The flow analysis algorithm is verified, updated, and extended by Rahmati [5] for flow analysis and the inverse design of turbomachinery blades. The efficiency and accuracy of the current incompressible flow analysis algorithm has been validated for various complex geometries, such as axial turbomachinery [5] and a marine propeller [18]. The results of the numerical calculations are in good agreement with the published data. This data are both experimental and numerical and so allow a comparison not only with the real flow patterns but also with other codes that have been developed for the solution of the Navier–Stokes equations.

IV. Mesh Movement Algorithm

A mesh movement algorithm is an integral part of the current inverse design method because, once the blade surface is modified by the inverse algorithm, the corresponding triangular mesh also needs to be changed. The mesh movement algorithm is based on the method of Batina [19], who proposed a mesh movement method based on the linear tension spring analogy. In this method, each of the unstructured mesh 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 for every vertex by mesh-point movement. At static equilibrium of the system, the force at every vertex must be zero, that is,

$$\sum_{j=1}^{n_i} k_{ij}(\boldsymbol{\delta}_j - \boldsymbol{\delta}_i) = 0 \tag{11}$$

where k_{ij} is the spring stiffness coefficient at edge ij, which is defined as

$$k_{ij} = \frac{1}{L_{ij}} = \frac{1}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \tag{12}$$

$\boldsymbol{\delta}_i$ and $\boldsymbol{\delta}_j$ are the displacements of vertex i and its neighboring vertex j, respectively, and n_i is the total number of neighboring vertices (see Fig. 2).

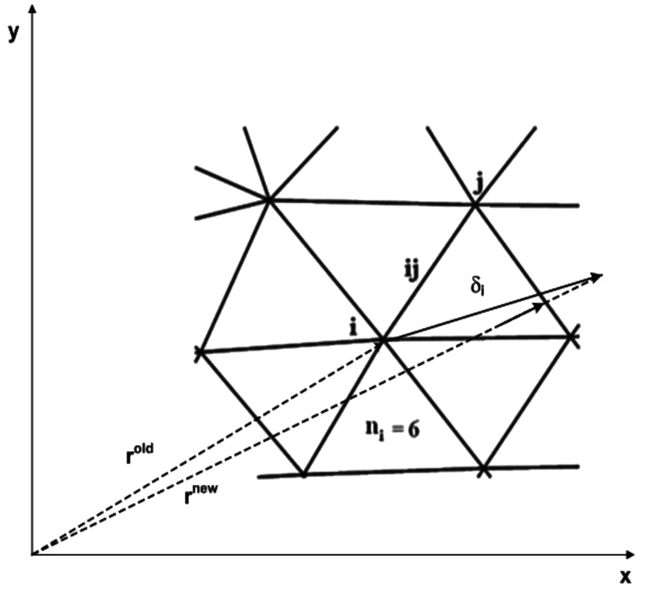

Fig. 2 Spring analogy mesh movement method for triangular meshes.

By solving the static equilibrium equations of the spring forces iteratively at each interior vertex, the final displacement of vertex i after n iterations is as follows:

$$\boldsymbol{\delta}_i^{n+1} = \frac{\sum_{j=1}^{n_i} k_{ij}\boldsymbol{\delta}_j^n}{\sum_{j=1}^{n_i} k_{ij}} \tag{13}$$

$\boldsymbol{\delta}_i^{n+1}$ is the final displacement of vertex i, and so the new position of vertex i) is determined as follows:

$$\mathbf{r}_i^{\text{new}} = \mathbf{r}_i^{\text{old}} + \boldsymbol{\delta}_i^{n+1} \tag{14}$$

where $\mathbf{r}_i^{\text{old}}$ and $\mathbf{r}_i^{\text{new}}$ are the old and new positions of vertex i, respectively. A linear tension spring analogy is applied here because only the nodal displacements are important and not purely elastic variables, such as local stress or strain. Also, this method is a robust method and has a low computational cost. The efficiency and robustness of the method for the inverse design of various blades and airfoils has been validated [6,18].

V. Blade Modifying Algorithm

For blade modification, the pressure loading, ΔP, and blade thickness are prescribed along an initial blade. The blade surface is treated as an elastic membrane that is modified according to the differences between the target and calculated pressure loading. The following linear partial differential equations are used as a residual corrector to modify the top and bottom contours of the blade, respectively:

$$-\beta_1\partial(\Delta y_{\text{top}}) + \beta_2\partial(\Delta y_{\text{top}})/\partial s + \beta_3\partial^2(\Delta y_{\text{top}})/\partial s^2 = \delta p^+ \tag{15}$$

$$\beta_1\partial(\Delta y_{\text{bottom}}) + \beta_2\partial(\Delta y_{\text{bottom}})/\partial s - \beta_3\partial^2(\Delta y_{\text{bottom}})/\partial s^2 = \delta p^- \tag{16}$$

Following Dulikravich and Baker [13], the following values are used for the arbitrary constants β_1, β_2, and β_3 in there equations: $\beta_1 = \Delta p_m$, $\beta_2 = 0$, and $\beta_3 = 0.4\ \Delta p_m$. Δp_m is the difference between the maximum and minimum magnitudes of the target pressure loading values. These arbitrary constants control the convergence of the inverse design. In Eqs. (15) and (16), L_E is the lower blade contour length, s is the blade contour following coordinate, L is the total blade counter length, and Δy_{top} and Δy_{bottom} are the normal displacements at the top and bottom counters of the blade, respectively (see Fig. 3).

The superscripts + and − are used to denote the upper and lower surfaces, respectively; δp^+ and δp^- are the aerodynamic loads, which are the difference between the calculated and target surface

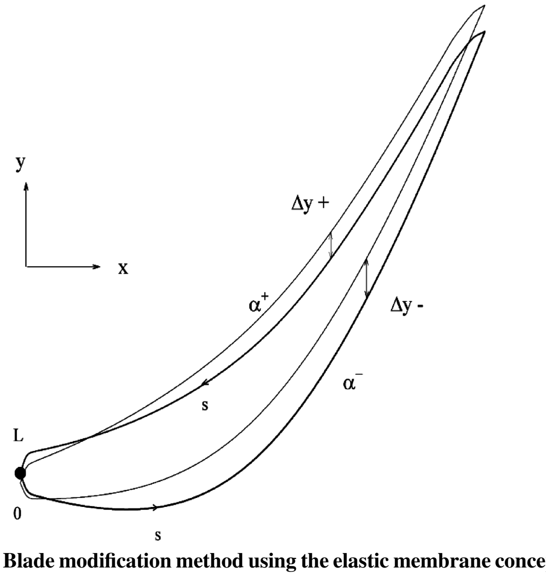

Fig. 3 Blade modification method using the elastic membrane concept.

pressures, that is,

$$\delta p^+ = p_t^+ - p_c^+ \tag{17}$$

$$\delta p^- = p_t^- - p_c^- \tag{18}$$

where p_t and p_c are the target and calculated surface pressures, respectively. The value of the target surface pressures on the top and bottom contours of the blade, p_t^+ and p_t^-, are unknown before blade design, but the target pressure loading, Δp_t, is prescribed. Based on the value of Δp_t, the following surface pressure distribution is used as the target pressure:

$$p_t^+ = (\Delta p_t / \Delta p_c) \times p_c^+ \tag{19}$$

$$p_t^- = (\Delta p_t / \Delta p_c) \times p_c^- \tag{20}$$

where Δp_c is the calculated pressure loading, defined as

$$\Delta p_c = p_c^+ - p_c^- \tag{21}$$

By using Eqs. (15) and (16) as the target pressure, the target pressure loading always remains equal to the prescribed target pressure loading, Δp_t. The substitution of Eq. (19) into Eq. (17), and Eq. (20) into Eq. (18), yields the following equations for the aerodynamics loading on the blade:

$$\delta p^+ = (\Delta p_t / \Delta p_c) \times p_c^+ - p_c^+ \tag{22}$$

$$\delta p^- = (\Delta p_t / \Delta p_c) \times p_c^- - p_c^- \tag{23}$$

These equations are rearranged as follows:

$$\delta p^+ = [(\Delta p_t / \Delta p_c) - 1] \times p_c^+ \tag{24}$$

$$\delta p^- = [(\Delta p_t / \Delta p_c) - 1] \times p_c^- \tag{25}$$

Equations (15) and (16) are nonhomogeneous second-order equations with constant coefficients. The complementary functions of these equations for the top and bottom contours are

$$\Delta y^{\text{top}} = q_1 e^{\lambda_1 s} + q_2 e^{\lambda_2 s} \tag{26}$$

$$\Delta y^{\text{bottom}} = q_3 e^{-\lambda_1 s} + q_4 e^{-\lambda_2 s} \tag{27}$$

where

$$\lambda_{1,2} = \frac{-\beta_2 \pm \sqrt{\beta_2^2 + 4\beta_1.\beta_2}}{2\beta_3} \tag{28}$$

q_1, q_2, q_3, and q_4 are constants that are computed from the boundary conditions at the leading and trailing edges. To calculate the particular integral of Eqs. (15) and (16), the surface distribution of δp is represented by using the Fourier series expansion as follows:

$$\delta p(s) = a_0 + \sum_{n=1}^{n_{\max}} a_n \cos(N_n s) + b_n \sin(N_n s) \tag{29}$$

where $N_n = 2n\pi/L$, and α_0, α_n, and b_n are the Fourier series coefficients. The particular integrals of Eqs. (15) and (16) are represented in the form of Fourier series as

$$\Delta y^{\text{top}} = A_0^{\text{top}} + \sum_{n=1}^{n_{\max}} A_n^{\text{top}} \cos(N_n s) + B_0^{\text{top}} \sin(N_n s) \tag{30}$$

$$\Delta y^{\text{bottom}} = A_0^{\text{bottom}} + \sum_{n=1}^{n_{\max}} A_0^{\text{bottom}} \cos(N_n s) + B_0^{\text{bottom}} \sin(N_n s) \tag{31}$$

The values of A_0, A_n, and B_n are found by the substitution of Eq. (30) into Eq. (15), and Eq. (31) into Eq. (16). These values are calculated as follows:

$$A_n^{\text{top}} = \frac{a_n(\beta_1 + N_n^2\beta_3) - b_n(\beta_2 N_n)}{(\beta_1 + N_n^2\beta_3) + (\beta_2 N_n)^2} \qquad 0 \le n \le n_{\max} \tag{32}$$

$$B_n^{\text{top}} = \frac{b_n(\beta_1 + N_n^2\beta_3) + a_n(\beta_2 N_n)}{(\beta_1 + N_n^2\beta_3) + (\beta_2 N_n)^2} \qquad 1 \le n \le n_{\max} \tag{33}$$

$$A_n^{\text{bottom}} = -\frac{a_n(\beta_1 + N_n^2\beta_3) + b_n(\beta_2 N_n)}{(\beta_1 + N_n^2\beta_3) + (\beta_2 N_n)^2} \qquad 0 \le n \le n_{\max} \tag{34}$$

$$B_n^{\text{bottom}} = -\frac{b_n(\beta_1 + N_n^2\beta_3) - a_n(\beta_2 N_n)}{(\beta_1 + N_n^2\beta_3) + (\beta_2 N_n)^2} \qquad 1 \le n \le n_{\max} \tag{35}$$

Having found the particular integral and complementary functions of Eqs. (15) and (16), the complete solution of these equations is given by the following equations:

$$\Delta y^{\text{top}} = q_1 e^{\lambda_1 s} + q_2 e^{\lambda_2 s} + \sum_{n=0}^{n_{\max}} A_n^{\text{top}} \cos(N_n s) + B_n^{\text{top}} \sin(N_n s) \tag{36}$$

$$\Delta y^{\text{bottom}} = q_3 e^{-\lambda_1 s} + q_4 e^{-\lambda_2 s} + \sum_{n=0}^{n_{\max}} A_n^{\text{bottom}} \cos(N_n s) + B_n^{\text{bottom}} \sin(N_n s) \tag{37}$$

Equations (36) and (37) contain four unknown constants: q_1, q_2, q_3, and q_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. For the airfoil shown in Fig. 3, this yields four equations with four unknowns, q_1, q_2, q_3, and q_4, which are found from the following matrix:

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & e^{L\lambda_1} & e^{L\lambda_2} \\ e^{-\lambda_1 L_E} & e^{-\lambda_2 L_E} & -e^{\lambda_1 L_E} & -e^{\lambda_2 L_E} \\ -\lambda_1 e^{-\lambda_1 L_E} & -\lambda_2 e^{-\lambda_2 L_E} & -\lambda_1 e^{\lambda_1 L_E} & -\lambda_2 e^{\lambda_2 L_E} \end{bmatrix}^{-1} \times \begin{bmatrix} -\sum_{n=1}^{n_{\max}} A_n^{\text{bottom}} \\ -\sum_{n=1}^{n_{\max}} A_n^{\text{top}} \\ \sum_{n=1}^{n_{\max}} \Delta A_n \cos(N_n L_E) + \Delta B_n \sin(N_n L_E) \\ \sum_{n=1}^{n_{\max}} -\Delta A_n N_n \sin(N_n L_E) + \Delta B_n N_n \cos(N_n L_E)] \end{bmatrix} \tag{38}$$

where

$$\Delta A_n = A_n^{\text{top}} - A_n^{\text{bottom}} \tag{39}$$

$$\Delta B_n = B_n^{\text{top}} - B_n^{\text{bottom}} \tag{40}$$

The new surface contours, α_{new}, based on these modifications will be

$$\alpha_{\text{new}}^+ = \alpha_{\text{old}}^+ + \Delta y_{\text{top}} \tag{41}$$

$$\alpha_{\text{new}}^- = \alpha_{\text{old}}^- + \Delta y_{\text{bottom}} \tag{42}$$

where α_{old} is the surface contour of the previous blade and is a function dependent on the x coordinate. However, in this method, the blade thickness is constant and the camber line is modified during design iterations. Thus, a relationship between the new and old camber lines is found by adding Eqs. (37) and (38):

$$\alpha_{\text{new}}^+ + \alpha_{\text{new}}^- = \alpha_{\text{old}}^+ + \alpha_{\text{old}}^- + \Delta y_{\text{top}} + \Delta y_{\text{bottom}} \tag{43}$$

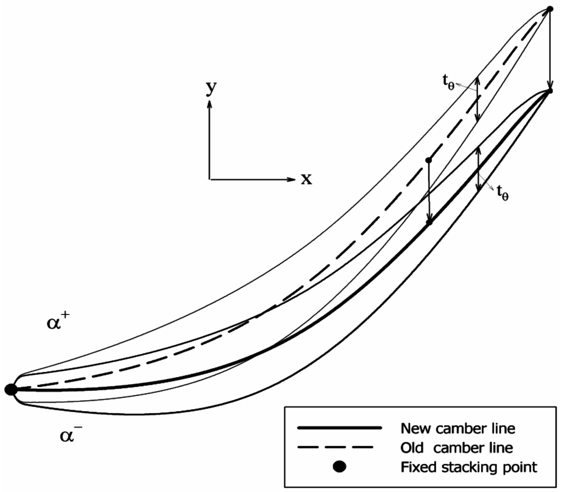

Fig. 4 New inverse design method; the stacking condition and the blade thickness are imposed.

By considering f_{old} and f_{new} as the old and new camber lines, respectively, and t_θ as the blade thickness, we have

$$\alpha^-_{\text{old}} = f_{\text{old}} - (t_\theta/2) \tag{44}$$

$$\alpha^+_{\text{old}} = f_{\text{old}} + (t_\theta/2) \tag{45}$$

$$\alpha^-_{\text{new}} = f_{\text{new}} - (t_\theta/2) \tag{46}$$

$$\alpha^+_{\text{new}} = f_{\text{new}} + (t_\theta/2) \tag{47}$$

The blade thickness, t_θ, is considered to be constant during design iterations (see Fig. 4). Substituting these four equations into Eq. (43) yields the following relation between the old and new camber lines:

$$f_{\text{new}} = f_{\text{old}} + \frac{\Delta y^{\text{top}} + \Delta y^{\text{bottom}}}{2} \tag{48}$$

Thus, by specifying the target pressure loading, the camber line instead of the blade surface deforms under the aerodynamics load. It deforms smoothly during design iterations until the calculated

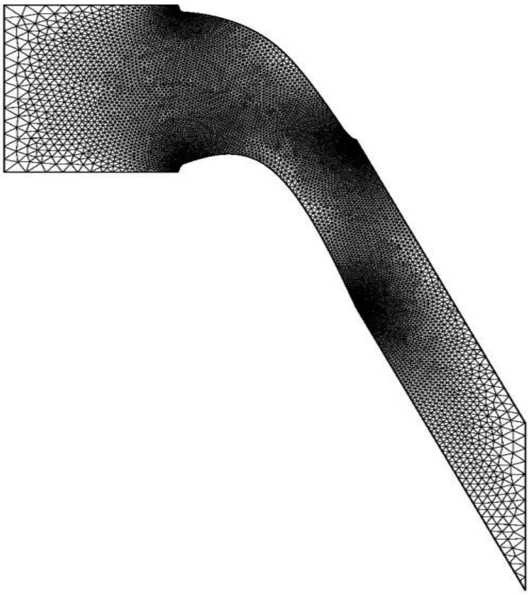

Fig. 5 The mesh with 15,776 cells around the initial stator blade.

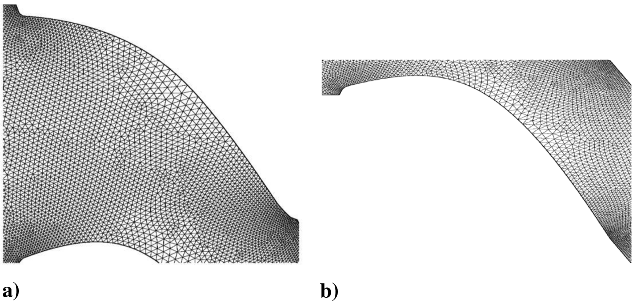

Fig. 6 Mesh: a) close to the pressure side of the stator blade, and b) close to the suction side of the blade.

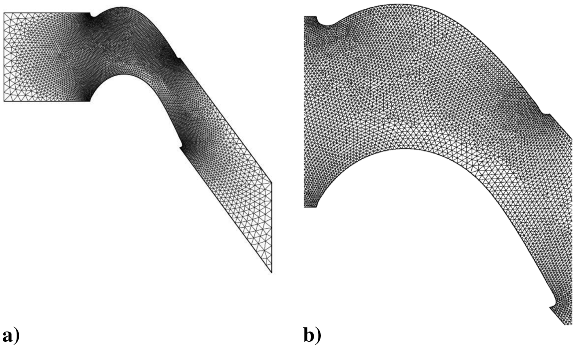

Fig. 7 Mesh with 13,436 triangle cells around the initial rotor blade: a) in the complete computational domain, and b) close to the blade.

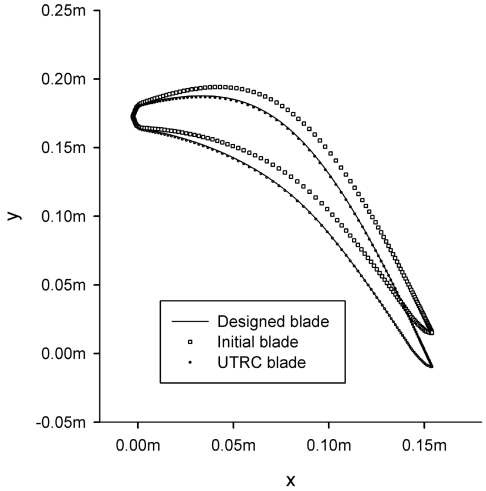

Fig. 8 Reproducing the UTRC stator blade based on prescribed pressure loading.

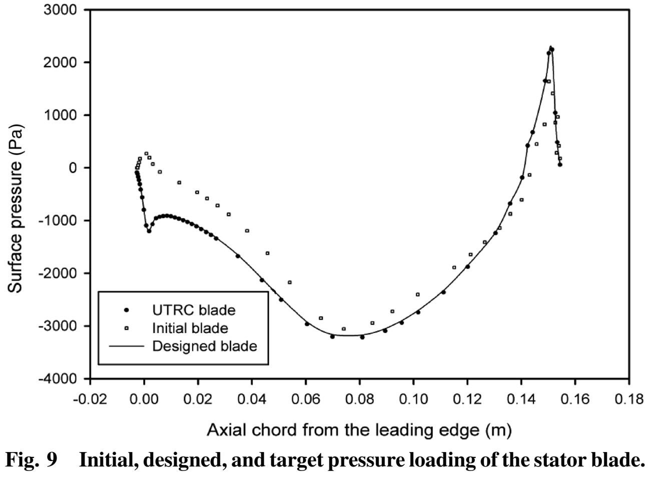

Fig. 9 Initial, designed, and target pressure loading of the stator blade.

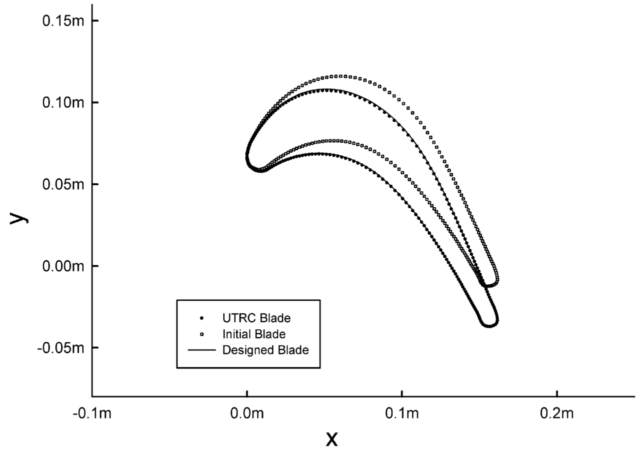

Fig. 10 Initial, designed, and UTRC rotor blade.

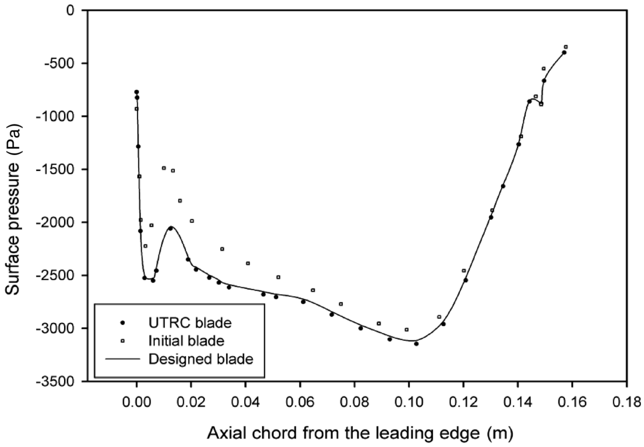

Fig. 11 Pressure loading of the initial, designed, and UTRC rotor blade.

surface pressure loading matches the target surface pressure loading. The new blade geometry is calculated by adding the constant blade thickness to the new camber line during design iterations. The process is repeated until the calculated blade pressure loading matches the target pressure loading.

Inverse blade design methods are now used in the industry to improve the performance of a known blade or to create a new design based on an earlier design that operates under the same conditions [10,11]. However, to assess the robustness and efficiency of inverse design methods, they are usually applied to the redesign of blades for which the experimental data are available.

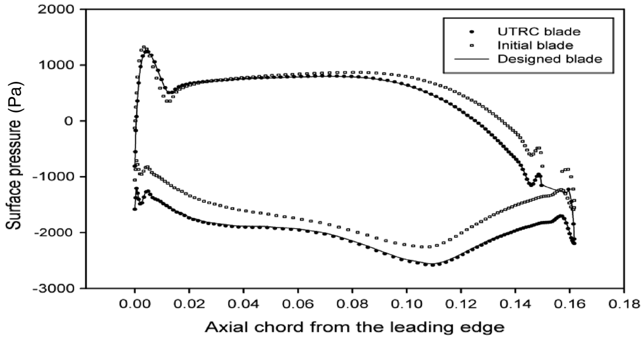

Fig. 13 Surface pressure of the initial, designed, and UTRC rotor blade.

VI. Inverse Design Applications

To assess the capability and robustness of the current method and its convergence to a unique solution, it is applied to the redesign of the midspan of a United Technologies Research Center (UTRC) rotor and stator blade. The data published in [20] on the experimental measurements of the rotor and stator at the UTRC are used as the target design in this study. The high-speed ratio of this blade isolates the end-wall effects and gives rise to a smooth and nearly two-dimensional flow near the midspan. Also, the average Mach numbers of flows at the outlet of the stator and rotors are approximately 0.2, and so the flow can be considered as an incompressible two-dimensional flow at the midspan of the blade. The axial chords of the stator and rotor blades are 15.06 and 0.161 m, respectively. The blade span for both blades is 0.1524 m. Figures 5–7 show the triangular mesh around the initial stator and rotor in the complete numerical domain and the close-up around the blade surface, respectively. To simulate the nominal operating rotational speed of the rotor, a constant tangential velocity of 29.5 m/s is specified in the flow computation. The axial flow velocity is 22.86 m/s, the same as the axial velocity in the experiment. The goal is to redesign the UTRC blade using the target pressure loading and an initial blade. To redesign the blade, an arbitrary blade that has the same blade thickness as the original UTRC blade but with a different camber line is taken to be the initial blade. Then the pressure loading of this initial blade is calculated. The discrepancies between the target and calculated pressure loadings determine the geometrical blade modification. Figures 8–11 show the geometries and pressure loadings of the initial, target, and redesigned blades. As shown in Figs. 12 and 13, the surface pressures of the designed blades and the original blade also match very closely. The results show a reasonable agreement between the UTRC blade and the designed blade. The global convergence criterion is set to 10^{-5} for all discretized equations of the flow analysis algorithm. The blade shape converges when the maximum discrepancy between the target and calculated

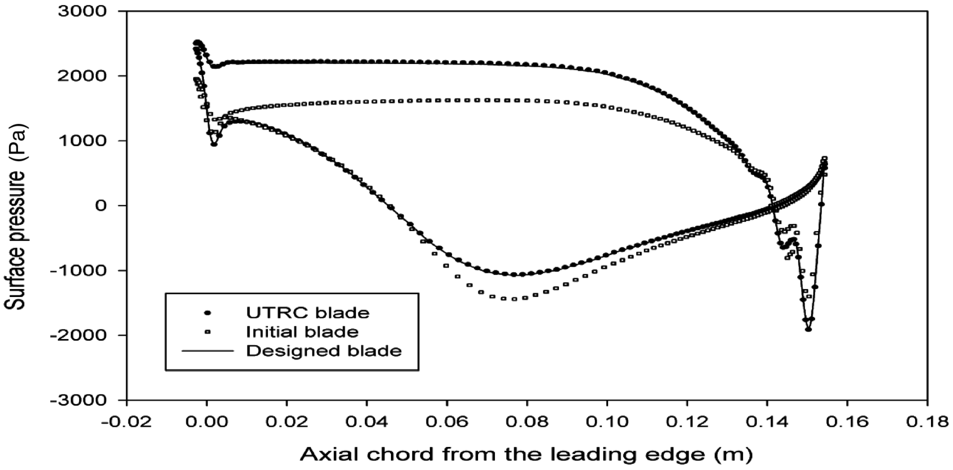

Fig. 12 Surface pressure of the initial, designed, and UTRC stator blade.

pressure loading is reduced to 1%. The design solution converges after only 65 and 80 calls to the analysis algorithm for the stator and rotor blades, respectively. Thus, the total numerical computational cost of the design is about 65–80 times the required convergence time of the flow analysis algorithm.

VII. Conclusions

A Navier–Stokes inverse design method based on the specification of pressure loading on the blade and the blade thickness is developed. In this method, the required blade modification is computed based on the discrepancies between the initial and target pressure loading distribution. Inviscid inverse design approaches are incapable of predicting viscous effects on the flow through the turbomachinery blades, which is the main cause for losses through skin friction on the walls and through separation. One of the improvements of the proposed method over previous inviscid methods is that the flowfields are treated as viscous turbulent flows using a Reynolds-averaged Navier–Stokes model. Also, as the proposed inverse design method is based on the specification of pressure loading distribution on the blade and the blade thickness, the method does not have the difficulties associated with a design based on the specification of surface pressure. Indeed, by specifying the pressure loading instead of the surface pressure, it is possible to control the pressure loading while maintaining structurally sound blades. The capabilities of this design methodology have been verified by reproducing the midsection of a UTRC rotor and stator blade operating in a subsonic flow region. By specifying a target pressure loading with blade thickness, the final blades were successfully reproduced. The numerical results show the efficiency of the method for blade inverse design.

Acknowledgment

The author would like to thank the many colleagues who helped undertake this research at the Department of Mechanical Engineering and Department of Engineering, Lancaster University.

References

[1] Denton, J. D., "The Calculation of Three-Dimensional Viscous Flow Through Multistage Turbomachines," *Journal of Turbomachinery*, Vol. 114, Jan. 1992, pp. 18–26. doi:10.1115/1.2927983

[2] Jennions, I. K., and Turner, M. G., "Three-Dimensional Navier Stokes Computations of Transonic Fan Flow Using an Explicit Flow Solver and an Implicit Solver," *Journal of Turbomachinery*, Vol. 115, No. 2, 1993, pp. 261–272. doi:10.1115/1.2929232

[3] Lakshminarayana, B., *Fluid Dynamics and Heat Transfer of Turbomachinery*, Wiley, New York, 1996, Chap. 2.

[4] Obayashi, S., and Takanashi, S., "Genetic Optimization of Target Pressure Distributions for Inverse Design Methods," *AIAA Journal*, Vol. 34, No. 5, 1996, pp. 881–886. doi:10.2514/3.13163

[5] Rahmati, M. T., "Incompressible Navier–Stokes Inverse Design Method Based on Unstructured Meshes," Ph.D. Thesis, Dept. of Mechanical Engineering, University College London, London, 2006.

[6] Hu, P., Choo, B., Zangeneh, M., and Rahmati, M., "On Design of Transonic Fan Rotors by 3D Inverse Design Method," American Society of Mechanical Engineers Paper GT2006-91173, 2006.

[7] Lighthill, M. J., "A New Method of Two Dimensional Aerodynamics Design," Aeronautical Research Council Rept. 2112, London, 1945.

[8] Dang, T., "A Fully Three-Dimensional Inverse Method For Turbomachinery Blading in Transonic Flows Design," *Journal of Turbomachinery*, Vol. 115, 1993, pp. 354–361. doi:10.1115/1.2929241

[9] Demeulenaere, A., and Braembussche, R. V. D., "Three-Dimensional Inverse Method for Turbomachinery Blading Design," *Journal of Turbomachinery*, Vol. 120, No. 2, April 1998, pp. 247–255.

[10] Dang, T., and Jiang, J., "Design Method for Turbomachine Blades with Finite Thickness by the Circulation Method," *Journal of Turbomachinery*, Vol. 119, No. 3, 1997, pp. 539–543.

[11] Dang, T., Damle, S., and Qiu, X., "Euler-Based Inverse Method for Turbo-Machine Blade, Part 2: Three-Dimensional Flows," *AIAA Journal*, Vol. 38, No. 11, 2000, pp. 207–213.

[12] Garabedian, P., and McFadden, G., "Computational Fluid Dynamics of Airfoils and Wings in Transonic, Shock, and Multi-Dimensional Flows," *Advances in Scientific Computing*, Academic Press, 1982, pp. 1–16.

[13] Dulikravich, G. S., and Baker, D. P., "Fourier Series Solution for Inverse Design of Aerodynamic Shape," *37th Aerospace Sciences Meeting and Exhibit*, AIAA, Reston, VA, Jan. 1999, pp. 11–14.

[14] Launder, B. E., and Spalding, D. B., "The Numerical Computation of Turbulent Flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 3, 1974, pp. 269–289. doi:10.1016/0045-7825(74)90029-2

[15] Charlesworth, D., "Solutions of the Incompressible Navier-Stokes Equations on Unstructured Meshes," Ph.D. Thesis, Dept. of Mechanical Engineering, Univ. College London, London, 2003.

[16] Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Taylor and Francis, Washington, D.C., 1980.

[17] Mathur, S. R., and Murthy, Y. T., "A Pressure-Based Method for Unstructured Meshes," *Numerical Heat Transfer*, Vol. 31, Part B, 1997, pp. 195–215.

[18] Rahmati, M. T., "A RANS Code for Flow Simulation of Marine Propellers," *The International Conference on Computer Applications in Shipbuilding*, Royal Institution of Naval Architects, London, Sept. 2007.

[19] Batina, J. T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," *AIAA Journal*, Vol. 28, No. 8, 1990, pp. 1381–1388. doi:10.2514/3.25229

[20] Dring, R. P., Joslyn, H. D., Hardin, L. W., and Wagner, J. H., "Turbine Rotor-Stator Interaction," American Society of Mechanical Engineers, Paper 82-GT-3, 1982.

J. Wei
Associate Editor