Adjoint Aerodynamic Design Optimization for Blades in Multistage Turbomachines—Part I: Methodology and Verification

D. X. Wang
School of Engineering,
Durham University,
Durham DH1 3LE, UK

L. He
Department of Engineering Science,
Oxford University,
Parks Road,
Oxford OX1 3PJ, UK

The adjoint method for blade design optimization will be described in this two-part paper. The main objective is to develop the capability of carrying out aerodynamic blading shape design optimization in a multistage turbomachinery environment. To this end, an adjoint mixing-plane treatment has been proposed. In the first part, the numerical elements pertinent to the present approach will be described. Attention is paid to the exactly opposite propagation of the adjoint characteristics against the physical flow characteristics, providing a simple and consistent guidance in the adjoint method development and applications. The adjoint mixing-plane treatment is formulated to have the two fundamental features of its counterpart in the physical flow domain: conservation and nonreflectiveness across the interface. The adjoint solver is verified by comparing gradient results with a direct finite difference method and through a 2D inverse design. The adjoint mixing-plane treatment is verified by comparing gradient results against those by the finite difference method for a 2D compressor stage. The redesign of the 2D compressor stage further demonstrates the validity of the adjoint mixing-plane treatment and the benefit of using it in a multi-bladerow environment. [DOI: 10.1115/1.3072498]

1 Introduction

With the advance in computational fluid dynamics (CFD) and computing power, modern turbomachinery aerodynamic design relies almost completely on CFD to develop three dimensional blade sections [1]. There are many different approaches that are used to explore three dimensional features of turbomachinery blades. All these approaches can be generally classified into two categories: inverse design methods and direct design methods.

The conventional inverse design offers better designs with low and affordable computational time cost [2,3]. The major disadvantage of this method is that it requires the specification of target flow field in terms of pressure or velocity distributions on a blade surface; hence its success depends largely on a designer's experience and insight. Apart from that, it is usually difficult to apply constraints in the inverse design [4].

The direct design methods can be further classified into gradient-based methods and stochastic methods. Stochastic methods, such as the genetic algorithm and the simulated annealing, do not need gradients of an objective function but values of an objective function only. In principle, these methods are able to find the global optimum in a design space and have long been researched in turbomachinery aerodynamic design optimization [4–6]. But their applications in routine designs are normally restricted due to their huge time costs. The response surface method [7] and the evolutionary method with approximate models [8] have been gaining a lot of attention over the past several years mainly due to their easy implementation and affordable time cost with a few design variables.

The gradient-based methods, according to the way in which gradients are calculated, include the finite difference method, the linearized method, and the adjoint method. The finite difference method suffers from step size dependence and is not recommended. The main drawback is that the time cost of evaluating objective function derivatives using both the finite difference method and the linearized method is usually proportional to the number of design variables; hence they are not preferred for situations where there are a large number of design variables. The adjoint method does not have the well-known disadvantage of the finite difference method and offers a means to calculate derivatives of an objective function with respect to design variables with low time cost independent of the number of design variables. However the adjoint method is complex and counterintuitive, which has been curbing its popularity.

The application of the adjoint method in aerodynamic design optimization based on CFD was pioneered by Jameson [9]. Based on the way in which a final discrete adjoint system is formed, there are two variations: the continuous adjoint method and the discrete adjoint method. In the continuous adjoint method, the nonlinear flow equations in a partial differential equation form are linearized first with respect to a design variable. Then an adjoint system will be derived from the linearized flow equations, followed by discretization. In the discrete adjoint method, the flow equations in a partial differential equation form are discretized first, followed by the linearization and adjoint formulation.

The discrete adjoint method can produce the exact gradient of an objective function with respect to the variables in a discretized flow system, which will ensure that a design process converges quickly and fully. However it is very difficult to develop discrete adjoint codes, particularly by hand and with high-order upwind schemes and sophisticated turbulence models implemented in a flow solver. It is worthwhile to note that automatic differentiation (AD) can be exploited to generate discrete adjoint codes, and data structures of a flow solver can be exploited to improve the runtime performance of AD generated adjoint codes. However the discrete adjoint codes usually require more memory and CPU time compared with its continuous counterpart [10]. In particular, the discrete adjoint codes obtained with the state of the art AD technologies would typically consume three times of CPU time compared with that by a flow solver [11,12] (the time cost of a continuous adjoint solver is usually the same as that of its flow solver). In the

Contributed by the International Gas Turbine Institute of ASME for publication in the Journal of Turbomachinery. Manuscript received August 31, 2008; final manuscript received October 7, 2008; published online January 13, 2010. Review conducted by David Wisler. Paper presented at the ASME Turbo Expo 2008: Land, Sea and Air (GT2008), Berlin, Germany, June 9–13, 2008.

continuous adjoint method, one is free to discretize the adjoint equations in any consistent way, although it is always better to consult the flow solver schemes. The gradients from a continuous adjoint solver are subject to two levels of discretization errors for both the flow equations and the adjoint equations, which vanish with mesh refinement. More details about the advantages and disadvantages of the two approaches can be found in Refs. [10,13].

In the context of a CFD-based design optimization, the adjoint method (both the continuous one and the discrete one) first gained attention in applications to external flows (e.g., for aerofoil designs). This might be due to the fact that the adjoint method was first introduced to the external flow community. Jameson and coworkers [9,10,14,15] investigated both the continuous adjoint method and the discrete one for designs ranging from aerofoils and wings to aircrafts. Kim and Nakahashi [16] and Nielsen and Anderson [17] developed hand-coded discrete adjoint solvers based on flow solvers using unstructured meshes for the ONERA M6 wing design optimization. Recently the adjoint method has been extended to aerofoil designs with an unsteady flow field involved using the unsteady time-domain flow equations [18] and frequency domain flow equations [12,19] to control the time-averaged aerodynamic performance, unsteady aerodynamics, or aeroelasticity.

The applications of the adjoint method in design optimization to internal flows have been lagging far behind. The status might be indicated by the fact that an adjoint solution with the 2D Euler flow equations for turbomachinery blading aerodynamics was still a research topic very recently (see, e.g., Refs. [20,21]). There have been increasing efforts made in the applications of the adjoint method in turbomachinery blading aerodynamics. Yang et al. [22] and Wu et al. [23] applied the continuous adjoint method to 2D turbomachinery blading aerodynamic design optimization. Wu et al. [24] and Papadimitriou and Giannakoglou [25] developed continuous adjoint solvers for 3D turbomachinery blading aerodynamic design optimization. Corral and Gisbert [26] developed a hand-coded discrete adjoint solver for turbine end-wall profiling to reduce the secondary flow losses. The AD tool TAPENADE was exploited by Duta et al. [11] to generate discrete adjoint solvers to save human effort in coding development for turbomachinery aerodynamic design optimization. Florea and Hall [27] developed a discrete adjoint solver based on the time-linearized method for sensitivity analysis of an unsteady inviscid flow through turbomachinery cascades. Thomas et al. [12] used the AD tool TAF to develop adjoint codes to model unsteady aerodynamic design sensitivities. The adjoint method was also used for an efficient blade forced response minimization by Duta et al. [28].

It is noted that all the reported design optimizations of turbomachinery blades by using the adjoint method have been carried out in a single blade row computational domain. For multistage turbomachines, the matching between adjacent blade rows has important effects. There are different levels of computational analysis tools for including these blade row matching/interaction effects ranging from a fully time-domain unsteady Reynolds averaged Navier-Stokes (RANS) to a relatively simple circumferentially mixing-out steady flow treatment (the mixing-plane treatment first proposed by Denton [29]). The mixing-plane multistage steady flow analysis has become a standard industrial tool used in a design environment. Given the importance of the multistage matching, it seems natural to argue that an aerodynamic design optimization should be carried out in a multistage environment in order to make an effective impact on practical design processes. Recognizing this need and the current status of the multistage analysis in practical blading designs, the present work is aimed at developing an adjoint mixing-plane method for aerodynamic design optimization of blades in a multistage environment using the adjoint method.

2 Flow Equations and Boundary Conditions

The flow solver was developed at Durham University [30]. It has been under extensive development for steady and unsteady flow field calculations and has been validated against experimental data, analytic solution or benchmark results for its capability of producing quality steady and unsteady flow field [31–33]. As we all know, turbomachinery involves rotary components, and the time-averaged flow field inside turbomachinery exhibits a periodicity in the circumferential direction. Therefore, it is convenient to adopt a cylindrical coordinate system, which will facilitate an easy implementation of the periodic boundary condition and the use of a single blade passage domain. The 3D Reynolds averaged steady flow Navier-Stokes equations in a cylindrical coordinate system are given by

$$\frac{\partial(F-V_x)}{\partial x}+\frac{\partial(G-Uv_g-V_\theta)}{r\,\partial\,\theta}+\frac{\partial r(H-V_r)}{r\,\partial\, r}=S \tag{1}$$

where

$$U=(\rho,\rho u,\rho v r,\rho w,\rho E)^T$$

$$F=(\rho u,\rho u^2+p,\rho v r u,\rho w u,\rho H u)^T$$

$$G=(\rho v,\rho u v,(\rho v^2+p)r,\rho w v,\rho H v)^T$$

$$H=(\rho w,\rho u w,\rho v r w,\rho w^2+p,\rho H w)^T$$

$$S=(0,0,0,(\rho v^2+p)/r,0)^T$$

$$V_x=(0,\tau_{xx},r\tau_{\theta x},\tau_{rx},u\tau_{xx}+v\tau_{\theta x}+w\tau_{rx}+q_x)^T$$

$$V_\theta=(0,\tau_{x\theta},r\tau_{\theta\theta},\tau_{r\theta},u\tau_{x\theta}+v\tau_{\theta\theta}+w\tau_{r\theta}+q_\theta)^T$$

$$V_r=(0,\tau_{xr},r\tau_{\theta r},\tau_{rr},u\tau_{xr}+v\tau_{\theta r}+w\tau_{rr}+q_r)^T$$

v_g is the grid velocity at the circumferential direction.

Four sets of boundary conditions are used in the computations. They are the subsonic inlet boundary condition, the subsonic exit boundary condition, the viscous solid wall boundary condition, and the periodic boundary condition. At a subsonic inlet boundary, total pressure, total temperature, and flow angles are specified. At a subsonic exit boundary, static pressure is prescribed at the hub or tip, with the pressure radial distribution subject to the simple radial equilibrium. Along a viscous solid wall boundary, a slip boundary condition together with a wall function (log-law), as used by Denton [29], is adopted to reduce the mesh density near the solid wall region to save computational time. The periodic boundary condition is the simplest one among the four to be implemented and requires flow variables in dummy cells along a periodic boundary to be equal to flow variables in corresponding repeating cells.

In a multistage computation, the circumferentially averaged and characteristic-based mixing-plane treatment [34] is used at an interface between adjacent rows to enable the communication in the flow solution between adjacent rows. The flow mixing-plane treatment will ensure the conservation in mass, momentum, and energy across an interface at each radial section. This circumferential mixing-out of the flow nonuniformity will inevitably generate the mixing loss across the interface. In addition, the mixing-out would consequently mean that the unsteady interaction between adjacent rows cannot be included in this treatment.

The flow governing equations are spatially discretized in a cell-centered finite volume method framework with inviscid fluxes calculated using the central difference scheme coupled with a blended second- and fourth-order numerical dissipation. Time integration is achieved by using the four-stage Runge–Kutta method. The turbulent viscosity (μ_t) is obtained by the one equation Spalart–Allmaras turbulence model. The multigrid and local time-stepping techniques are employed to speed up the convergence of the solution process.

3 Adjoint Equations and Boundary Conditions

The continuous adjoint method has been adopted in our research due to its lower memory requirement, lower CPU time cost, and easier implementation (compared with the discrete one). The derivation of the adjoint equations based on the above flow governing equations follows the same way given in Refs. [13,14]. Turbulent viscosity is frozen in our adjoint formulation for simplicity. Thus for each design cycle, the gradient calculations are conducted at a frozen turbulent viscosity, while it will be updated in the direct flow solution once in every design cycle.

3.1 Adjoint Principle. The principle of the adjoint formulation will be first illustrated concisely here. The objective function I (a scalar) in an aerodynamic design optimization is a function of the flow variable vector U and a design variable α expressed as follows:

$$I = I(U, \alpha) \tag{2}$$

The relation between the flow variable and the design variable is determined through the nonlinear flow equation (a vector equation),

$$R(U, \alpha) = 0 \tag{3}$$

In a gradient-based design optimization, the gradient of the objective function to design variables is an essential element. Once the gradients are obtained, the steepest descent method or the conjugate gradient method can be applied directly for an optimization. The gradient of the objective function to a design variable can be given by

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial \alpha} + \frac{\partial I}{\partial U}\frac{\partial U}{\partial \alpha} \tag{4}$$

In the above gradient expression, $\partial I/\partial \alpha$ and $\partial I/\partial U$ can be calculated analytically or by the complex variable method [35], and no iteration will be involved. The calculation of the flow variable sensitivity $\partial U/\partial \alpha$ will have to involve solving equations (the nonlinear flow equations or the linearized flow equations). This can be done either directly by taking a difference between a perturbed flow field solution and the original (i.e., the finite difference method) or by solving the linearized flow equation,

$$\frac{\partial R}{\partial \alpha} + \frac{\partial R}{\partial U}\frac{\partial U}{\partial \alpha} = 0 \tag{5}$$

The linearized flow Eq. (5) again will need to be solved in an iterative process, accounting for the main time cost in the gradient evaluation. This linearized equation also depends on a design variable, implying that each design variable requires the linearized flow equation to be solved once. Hence both the direct finite difference and the linearized methods prohibit the use of a large number of design variables. The situations with a large number of design variables are exactly what the adjoint method is devised to deal with. The key is to find a way to decouple the influence of different design variables on an objective function through the flow variable sensitivity, and this can be achieved by eliminating the explicit dependency of the objective function sensitivity on the flow variable sensitivity $\partial U/\partial \alpha$ in Eq. (4).

Multiplying the right hand side of the linearized flow equation (5) with the adjoint variable vector (also called Lagrange multiplier) λ of the same dimension of U and subtracting the product from the gradient expression in Eq. (4) yields

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial \alpha} + \frac{\partial I}{\partial U}\frac{\partial U}{\partial \alpha} - \lambda^T\left[\frac{\partial R}{\partial \alpha} + \frac{\partial R}{\partial U}\frac{\partial U}{\partial \alpha}\right] \tag{6a}$$

We regroup the above in the following way:

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial \alpha} - \lambda^T\frac{\partial R}{\partial \alpha} + \left[\frac{\partial I}{\partial U} - \lambda^T\frac{\partial R}{\partial U}\right]\frac{\partial U}{\partial \alpha} \tag{6b}$$

If the adjoint variable is chosen to satisfy the following equation:

$$\frac{\partial I}{\partial U} - \lambda^T\frac{\partial R}{\partial U} = 0 \tag{7}$$

then the gradient expression is reduced as follows:

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial \alpha} - \lambda^T\frac{\partial R}{\partial \alpha} \tag{8}$$

This new gradient expression (8) does not depend on the flow variable sensitivity anymore. Furthermore, the adjoint equation (7) does not depend on any design variable. This implies that the gradient of a scalar objective function to all the design variables can be obtained by solving only two sets of equations: the flow equation (Eq. (3)) and the adjoint equation (Eq. (7)). Once the flow and adjoint solution is obtained, it can be substituted into the new gradient expression (8) to calculate the gradients very efficiently.

3.2 Adjoint Equations. The illustration in the preceding section is presented in an algebraic way. As mentioned in Sec. 1, the continuous adjoint method works on the flow equations in a partial differential equation form. Instead of regrouping in expression (6*b*), integration by parts needs to be used to derive the corresponding adjoint equations in a partial differential equation form. This section presents the detailed derivation of the adjoint equations from the flow equations in a partial differential equation form.

The derivation of the adjoint equations based on the Reynolds averaged Navier–Stokes Eqs. (1) is given in Appendix A1. The following derivation is based on the Euler flow equations, aimed at a simpler illustration of the principal procedure of a continuous adjoint formulation.

In the present research work, the objective function is a weighted sum of mass flow rate, pressure ratio and entropy generation rate, which can be expressed as a boundary integral in the following general form:

$$I = \int_{\partial D} M ds \tag{9}$$

where M is a function of flow variables and design variables. The gradient of the objective function to a design variable takes the following formulation:

$$\frac{dI}{d\alpha} = \int_{\partial D}\left[\frac{\partial M}{\partial U}\tilde{U} + \frac{\partial M}{\partial \alpha}\right]ds + \int_{\partial D} M\widetilde{ds} \tag{10}$$

where $\tilde{U}=\partial U/\partial \alpha$ is the flow variable sensitivity. $\partial M/\partial \alpha$ can be obtained in an analytic way or by the complex variable method [35] with little effort. However, the flow variable sensitivity is determined through the linearized flow equations,

$$\frac{\partial(A\tilde{U})}{\partial x} + \frac{\partial(B\tilde{U} - \tilde{U}v_g)}{r\,\partial\,\theta} + \frac{\partial r(C\tilde{U})}{r\,\partial\,r} - D\tilde{U} = f \tag{11}$$

where $A=\partial F/\partial U$, $B=\partial G/\partial U$, $C=\partial H/\partial U$, and $D=\partial S/\partial U$ are Jacobian matrices. f contains the linearization of geometric terms in the flow equations to a design variable α corresponding to $-\partial R/\partial \alpha$ in Eq. (5).

In order to avoid solving Eq. (11) n times for n design variables, we resort to the adjoint method, which offers a means to calculate the objective function sensitivity at a low time cost independent of the number of design variables.

Multiplying two sides of Eq. (11) with the adjoint variable λ and rearranging yields

$$\lambda^T\left[\frac{\partial(A\tilde{U})}{\partial x} + \frac{\partial(B\tilde{U} - \tilde{U}v_g)}{r\,\partial\,\theta} + \frac{\partial r(C\tilde{U})}{r\,\partial\,r} - D\tilde{U} - f\right] = 0 \tag{12a}$$

where $\lambda=(\lambda_1,\lambda_2,\lambda_3,\lambda_4,\lambda_5)^T$. λ_1, λ_2, λ_3, λ_4, and λ_5 correspond to the continuity equation, the axial momentum equation, the moment of momentum equation, the radial momentum equation, and the energy equation, respectively. Equation (12*a*) is a scalar equation and is valid everywhere in a computational domain, no matter what value the adjoint variable will take. Integrating the left hand side of the above equation over the whole computational domain, one has

$$\int_D \lambda^T \left[\frac{\partial (A\tilde{U})}{\partial x} + \frac{\partial (B\tilde{U} - \tilde{U} v_g)}{r\,\partial \theta} + \frac{\partial r(C\tilde{U})}{r\,\partial r} - D\tilde{U} - f \right] dv \tag{12b}$$

Performing integration by parts once yields

$$\int_{\partial D} [\lambda^T A n_x + \lambda^T (B - v_g I) n_\theta + \lambda^T C n_r] \tilde{U} ds - \int_D \left[\frac{\partial \lambda^T}{\partial x} A + \frac{\partial \lambda^T}{r\,\partial \theta}(B - v_g I) + \frac{\partial \lambda^T}{\partial r} C + \lambda^T D \right] \tilde{U} dv - \int_D \lambda^T f dv \tag{12c}$$

where I is a 5×5 identity matrix. Expression (12*c*) is zero. Subtracting expression (12*c*) from Eq. (10) does not change the gradient in Eq. (10), namely,

$$\frac{dI}{d\alpha} = \int_{\partial D} \left[\frac{\partial M}{\partial U} \tilde{U} + \frac{\partial M}{\partial \alpha} \right] ds + \int_{\partial D} M\widetilde{ds} - \int_{\partial D} [\lambda^T A n_x + \lambda^T (B - v_g I) n_\theta + \lambda^T C n_r] \tilde{U} ds + \int_D \left[\frac{\partial \lambda^T}{\partial x} A + \frac{\partial \lambda^T}{r\,\partial \theta}(B - v_g I) + \frac{\partial \lambda^T}{\partial r} C + \lambda^T D \right] \tilde{U} dv + \int_D \lambda^T f dv \tag{13a}$$

Collecting the domain integral terms with $\tilde{U}$ and boundary integral terms with $\tilde{U}$ separately, one has

$$\frac{dI}{d\alpha} = \int_D \left[\frac{\partial \lambda^T}{\partial x} A + \frac{\partial \lambda^T}{r\,\partial \theta}(B - v_g I) + \frac{\partial \lambda^T}{\partial r} C + \lambda^T D \right] \tilde{U} dv - \int_{\partial D} \left[\lambda^T A n_x + \lambda^T (B - I) n_\theta + \lambda^T C n_r - \frac{\partial M}{\partial U} \right] \tilde{U} ds + \int_{\partial D} \frac{\partial M}{\partial \alpha} ds + \int_{\partial D} M\widetilde{ds} + \int_D \lambda^T f dv \tag{13b}$$

Returning to the original objective of developing the adjoint method, the gradient for *n* design variables can be efficiently evaluated if we do not have to solve the flow equations *n* times. This can be achieved if the explicit dependence of the objective function gradient on the flow variable sensitivity can be eliminated. In order to eliminate the dependence of the gradient on the flow variable sensitivity $\tilde{U}$, the first two integral terms on the right hand side of expression (13*b*) must vanish. Inside a computational domain, at every point, components of $\tilde{U}$ are independent and cannot always be zero; then the term multiplying $\tilde{U}$ in the domain integral term needs to vanish, leading to the adjoint Euler equations,

$$\frac{\partial \lambda^T}{\partial x} A + \frac{\partial \lambda^T}{r\,\partial \theta}(B - v_g I) + \frac{\partial \lambda^T}{\partial r} C + \lambda^T D = 0 \tag{14}$$

Along the boundary of a computational domain, components of $\tilde{U}$ are not always independent and can always be zero somewhere; e.g., on a viscous solid wall, the convective fluxes are always zero. Therefore the term multiplying $\tilde{U}$ in the boundary integral does not need to be zero all the time. The flow boundary conditions need to be incorporated to eliminate the dependence of the boundary integral term multiplying $\tilde{U}$ in expression (13*b*) on the flow variable sensitivity $\tilde{U}$, leading to the adjoint boundary conditions. These boundary conditions will be detailed in the following adjoint boundary condition section. The remaining terms including both the domain integral terms and boundary integral terms in Eq. (13*b*) are independent of the flow variable sensitivity $\tilde{U}$. These terms give the final expression of the gradient in an intended form, independent of the flow variable sensitivity,

$$\frac{dI}{d\alpha} = \int_{\partial D} \frac{\partial M}{\partial \alpha} ds + \int_{\partial D} M\widetilde{ds} + \int_D \lambda^T f dv \tag{15}$$

The values of the adjoint variables are determined through a solution to the adjoint equations. No matter how many design variables are there, there are now only two sets of equations to be solved: the nonlinear flow equations and the linear adjoint equations. The time cost of solving the adjoint equations is equivalent to that for the baseline flow equations. Once the adjoint solution is obtained, it can be substituted into the above gradient expression to calculate gradients very efficiently.

3.3 Adjoint Boundary Conditions. It can be seen from the previous section (e.g., Eqs. (13*b*) and (14)) that the adjoint field equation is obtained by eliminating the dependence on the flow variable sensitivity in the *field* integration. Accordingly, the starting point of the adjoint boundary condition treatment is to eliminate the dependence on the flow variable sensitivity in the *boundary* integration, i.e., to ensure at a boundary,

$$\left[\lambda^T A n_x + \lambda^T (B - I) n_\theta + \lambda^T C n_r - \frac{\partial M}{\partial U} \right] \tilde{U} = 0 \tag{16}$$

Corresponding to the flow boundary conditions, there are also four sets of adjoint boundary conditions: the subsonic inlet boundary condition, the subsonic exit boundary condition, the viscous solid wall boundary condition, and the periodic boundary condition.

Another important element in the boundary condition treatment is the equation and its characteristics. Equation (14) is usually solved in a time-marching manner with a pseudo-time-derivative term added to the equation to form a hyperbolic system (Eq. (21)). For a hyperbolic system, the specification of boundary conditions is based on the propagation of characteristic information. The basic compatibility requirement is that the number of boundary conditions specified must be equal to the number of the characteristics running into the computational domain. Hence the directions of the characteristics matter. As Giles and Pierce [36] pointed out, the sign of each adjoint characteristic velocity is opposite to that of each corresponding flow characteristic velocity. Overall it is useful to note that in a simple but consistent sense, the adjoint information propagates in exactly the opposite way to the physical information. This "antiphysics" path of information propagation in fact governs all the boundary condition treatments for the adjoint equations, described in this section and in Sec. 4. This will also be illustrated further in the numerical examples, to be presented later.

At a subsonic inlet boundary for the flow equations, there are four incoming characteristics and one outgoing characteristic. Typically the static pressure at the boundary is extrapolated from the interior domain, and other flow variables are determined through the isentropic relation and specified quantities. Thus at the inlet boundary, pressure is the only independent flow variable. The flux perturbation can be expressed as the derivative of the flux to the pressure times the pressure perturbation. The corresponding adjoint boundary condition is applied to eliminate the dependence of the boundary integral term along an inlet boundary on the flow variable sensitivity,

$$\lambda^T \cdot \partial F/\partial p \cdot n_x = \partial M/\partial p \tag{17}$$

in which an inlet plane is assumed to be normal to the x axis, resulting in the x-directional flux only through the boundary. Equation (17) is a scalar equation. This is consistent with one incoming and four outgoing characteristics for the adjoint equations at a subsonic inlet boundary. The adjoint variables $\lambda_{1,2,3,4}$ are extrapolated from the interior domain, while λ_5 is worked out through Eq. (17). The derivative of the flux vector to the pressure $(\partial F/\partial p)$ can be obtained efficiently via the complex variable method [35].

At a subsonic exit boundary, there is one incoming and four outgoing characteristics for the flow equations. In the present flow solver, the density and three velocity components are extrapolated from the interior domain, and static pressure is determined through the value specified at the hub or tip and the simple radial equilibrium distribution. This means that at a flow exit, density and velocity components are independent flow variables. The flux perturbation at the boundary can be expressed as follows:

$$\widetilde{F} = \partial F/\partial \rho \cdot \widetilde{\rho} + \partial F/\partial u \cdot \widetilde{u} + \partial F/\partial v \cdot \widetilde{v} + \partial F/\partial w \cdot \widetilde{w} \tag{18}$$

The corresponding adjoint boundary condition takes the following form:

$$\lambda^T \partial F/\partial q \cdot n_x = \partial M/\partial q, \quad q = \rho, u, v, w \tag{19}$$

where again an exit plane is assumed to be normal to the x axis. Equation (19) represents four equations. This is in line with the four incoming and one outgoing adjoint characteristics at a subsonic exit boundary. The adjoint variable λ_5 is extrapolated from the interior domain; other four adjoint variables are determined through Eq. (19).

Corresponding to a slip boundary condition with the log-law at a viscous solid wall boundary, the adjoint boundary condition is formulated as follows:

$$\lambda_2 \cdot n_x + \lambda_3 \cdot n_\theta \cdot r + \lambda_4 \cdot n_r + \lambda_5 \cdot n_\theta \cdot v_g = \partial M/\partial p$$

$$\sigma_x = \tau_w/W \cdot \lambda_2$$

$$\sigma_\theta = \tau_w/W \cdot (\lambda_3 + \Omega \cdot \lambda_5)$$

$$\sigma_r = \tau_w/W \cdot \lambda_4$$

$$\frac{\partial \lambda_5}{\partial n} = 0 \quad \text{for an adiabatic wall}$$

$$\lambda_5 = 0 \quad \text{for a nonadiabatic wall (temperature specified)} \tag{20}$$

Ω is the angular speed of the rotating mesh associated with a blade. τ_w is the wall shear stress calculated by the log-law. W is the magnitude of the velocity along the slip wall.

The implementation of the adjoint periodic boundary condition is identical to the flow periodic boundary and simply equals the adjoint variables at corresponding periodic boundaries. The adjoint mixing-plane treatment will be illustrated in the next section in detail.

In order to solve the adjoint Eq. (14) in a finite volume framework, it is first written in a strong conservative form with extra terms taken as source terms. Then a pseudo-time-derivative term is added to the adjoint equations as follows to make use of the Runge–Kutta time integration method:

$$\frac{\partial \lambda}{\partial \tau} - \frac{\partial A^T \lambda}{\partial x} - \frac{\partial (B - v_g I)^T \lambda}{r\,\partial \theta} - \frac{\partial C^T \lambda}{\partial r} = \left[\frac{\partial A^T}{\partial x} + \frac{\partial (B - v_g I)^T}{r\,\partial \theta} + \frac{\partial C^T}{\partial r} - D^T \right] \lambda \tag{21}$$

As can be seen from the above, compared with Eq. (14), all terms change their signs into their opposite ones when the pseudo-time-derivative term is added. This is again in line with the property of adjoint characteristics (Giles and Pierce [36]) propagating in exactly the opposite way to the flow characteristics.

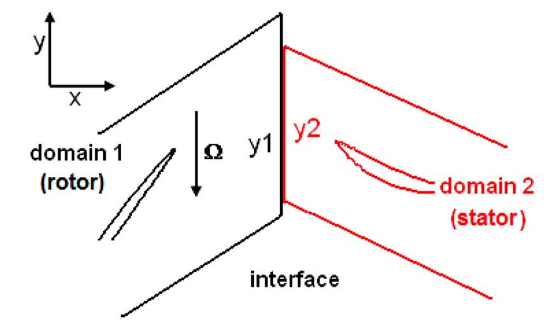

Fig. 1 An interface between a rotor and a stator configuration

The spatial discretization and time integration of the adjoint equations are implemented in the same way as for the flow governing equations. The multigrid and local time-stepping techniques are also used to speed up solving the adjoint equations. Subroutines in the flow solver, such as those for time-marching, multigrid, and metrics calculation, can be reused in developing the adjoint solver without any major modification. As for the initialization of the adjoint solution, our practice shows that the adjoint solution can be initialized with zeros or real numbers of small magnitudes.

4 Adjoint Mixing Plane

The main objective of the present work is to enable the adjoint-method-based design optimization to be carried out in a multistage environment. The basis of the present development is the mixing-plane treatment for flow solutions after Denton [29]. There are two basic features in the mixing-plane approach. The first is to satisfy the conservation of the total mass, momentum, and energy, which is fundamentally required. The second is to be nonreflective, which is highly desirable. The present adjoint mixing-plane treatment is also aimed at having these two features.

In the flow mixing-plane treatment, the circumferentially averaged fluxes of mass, momentum, and energy are conserved across an interface, as schematically shown in Fig. 1. In the adjoint mixing-plane treatment, the adjoint variable weighted and circumferentially averaged fluxes are conserved across an interface (see Appendix A2 for details),

$$\frac{1}{Y_1}\int_{y1} \lambda_1 \rho u dy = \frac{1}{Y_2}\int_{y2} \lambda_1 \rho u dy \tag{22a}$$

$$\frac{1}{Y_1}\int_{y1} \lambda_2(\rho u^2 + p) dy = \frac{1}{Y_2}\int_{y2} \lambda_2(\rho u^2 + p) dy \tag{22b}$$

$$\frac{1}{Y_1}\int_{y1} \lambda_3(\rho v u) dy = \frac{1}{Y_2}\int_{y2} \lambda_3(\rho v u) dy \tag{22c}$$

$$\frac{1}{Y_1}\int_{y1} \lambda_4(\rho w u) dy = \frac{1}{Y_2}\int_{y2} \lambda_4(\rho w u) dy \tag{22d}$$

$$\frac{1}{Y_1}\int_{y1} \lambda_5(\rho H u) dy = \frac{1}{Y_2}\int_{y2} \lambda_5(\rho H u) dy \tag{22e}$$

where Y_1 and Y_2 are the pitch lengths of two adjacent domains. $y1$ and $y2$ are the exit of domain 1 and the inlet of domain 2, respectively. This conservation ensures that the two boundary integral terms in Eq. (A9) cancel out each other, leading to the independence of Eq. (A9) on the flow variable sensitivity.

The conservation of these adjoint variable weighted fluxes can be achieved through the nonreflective implementation following the nonreflective boundary conditions for solving the flow equations [34]. Applying the mean value theorem for integration to either side of the equalities (22*a*)–(22*e*), the following mixed-out variables can be obtained:

$$\widehat{\lambda_{1,i}} = \frac{\int_{yi} \lambda_1 \rho u dy}{\int_{yi} \rho u dy} \tag{23a}$$

$$\widehat{\lambda_{2,i}} = \frac{\int_{yi} \lambda_2 (\rho u^2 + p) dy}{\int_{yi} (\rho u^2 + p) dy} \tag{23b}$$

$$\widehat{(\lambda_3 v)_i} = \frac{\int_{yi} \lambda_3 (\rho v u) dy}{\int_{yi} (\rho u) dy} \tag{23c}$$

$$\widehat{(\lambda_4 w)_i} = \frac{\int_{yi} \lambda_4 (\rho w u) dy}{\int_{yi} (\rho u) dy} \tag{23d}$$

$$\widehat{\lambda_{5,i}} = \frac{\int_{yi} \lambda_5 \rho H u dy}{\int_{yi} \rho H u dy} \tag{23e}$$

In expressions (23*c*) and (23*d*), the mass flux is used in the denominator instead of corresponding momentum fluxes. This is because the circumferential and radial momentum fluxes may change their signs along an interface, which violates the premise of the mean value theorem for integration. However the mass flux, axial momentum flux, and energy flux usually do not change their signs along an interface (no reverse flow). According to the conservation in equalities (22*a*)–(22*e*), the mixed-out variables in Eqs. (23*a*)–(23*e*) should also be equal across an interface. Once there is a difference in these mixed-out variables across the interface, this difference is taken as a characteristic jump disturbance. The present implementation will pass the disturbance to the two sides of the interface. The local adjoint variables will be corrected according to the adjoint characteristic propagation to drive the difference to zero.

The adjoint characteristic variables in term of the variable perturbation are expressed as follows:

$$\tilde{\lambda}^c = S^T \tilde{\lambda} \tag{24}$$

where S is the matrix diagonalizing the Jacobian matrix $A = \partial F/\partial U = S \Lambda S^{-1}$, with Λ being a diagonal matrix of which diagonal elements are eigenvalues of the matrix A. Expanding the vector form of Eq. (24) gives

$$\tilde{\lambda}_1^c = \tilde{\lambda}_1 + (u-c)\tilde{\lambda}_2 + vr\tilde{\lambda}_3 + w\tilde{\lambda}_4 + (H - uc)\tilde{\lambda}_5 \tag{25a}$$

$$\tilde{\lambda}_2^c = \tilde{\lambda}_1 + (u+c)\tilde{\lambda}_2 + vr\tilde{\lambda}_3 + w\tilde{\lambda}_4 + (H + uc)\tilde{\lambda}_5 \tag{25b}$$

$$\tilde{\lambda}_3^c = \tilde{\lambda}_3 + \frac{v}{r}\tilde{\lambda}_5 \tag{25c}$$

$$\tilde{\lambda}_4^c = \tilde{\lambda}_4 + w\tilde{\lambda}_5 \tag{25d}$$

$$\tilde{\lambda}_5^c = \frac{\tilde{\lambda}_1}{u} + \tilde{\lambda}_2 + \left(u - \frac{W^2}{2u}\right)\tilde{\lambda}_5 \tag{25e}$$

where c is the speed of sound, $W^2 = u^2 + v^2 + w^2$, and r is the radial coordinate.

The information represented by $\tilde{\lambda}_1^c$ propagates at the speed of $-u+c$. The information represented by $\tilde{\lambda}_2^c$ propagates at the speed of $-u-c$. And the information represented by the other three adjoint characteristic variables all propagates at the same speed of $-u$. For both incoming and outgoing adjoint characteristic variables, flow variables in the adjoint characteristic expressions take their local values. For incoming adjoint characteristic variables, the primitive adjoint variable perturbations take values through the difference in the mixed-out variables across the interface, e.g.,

$$\tilde{\lambda}_1 = \widehat{\lambda_{1,2}} - \widehat{\lambda_{1,1}} \tag{26a}$$

$$\tilde{\lambda}_2 = \widehat{\lambda_{2,2}} - \widehat{\lambda_{2,1}} \tag{26b}$$

$$\tilde{\lambda}_3 = \frac{\widehat{(\lambda_3 v)_2} - \widehat{(\lambda_3 v)_1}}{v} \tag{26c}$$

$$\tilde{\lambda}_4 = \frac{\widehat{(\lambda_4 v)_2} - \widehat{(\lambda_4 v)_1}}{w} \tag{26d}$$

$$\tilde{\lambda}_5 = \widehat{\lambda_{5,2}} - \widehat{\lambda_{5,1}} \tag{26e}$$

For outgoing adjoint characteristic variables, the primitive adjoint variable perturbations take the local values.

In Fig. 1, the interface separates two boundaries, an exit for domain 1 and an inlet for domain 2. The exit of domain 1 has four incoming adjoint characteristics $(\tilde{\lambda}_2^c, \tilde{\lambda}_3^c, \tilde{\lambda}_4^c, \tilde{\lambda}_5^c)$, which will be calculated through expressions (25) and (26). The inlet of domain 2 has one incoming adjoint characteristic $\tilde{\lambda}_1^c$, which will be also obtained through expressions (25) and (26).

Once all the adjoint characteristic variables are obtained for each side of an interface, they can be converted to primitive adjoint variable perturbations via the following formulation. The primitive adjoint variable perturbations can then be directly used to update the local adjoint solution,

$$\tilde{\lambda} = (S^T)^{-1}\tilde{\lambda}^c \tag{27}$$

The scalar form of the above is given by

$$\tilde{\lambda}_5 = \frac{k-1}{c^2}\left(\frac{\tilde{\lambda}_1^c}{2} + \frac{\tilde{\lambda}_2^c}{2} - u\tilde{\lambda}_3^c - vr\tilde{\lambda}_4^c - w\tilde{\lambda}_5^c\right) \tag{28a}$$

$$\tilde{\lambda}_4 = -w\tilde{\lambda}_5 + \tilde{\lambda}_5^c \tag{28b}$$

$$\tilde{\lambda}_3 = -\frac{v}{r}\tilde{\lambda}_5 + \tilde{\lambda}_4^c \tag{28c}$$

$$\tilde{\lambda}_2 = -u\tilde{\lambda}_5 + \frac{1}{2c}(-\tilde{\lambda}_1^c + \tilde{\lambda}_2^c) \tag{28d}$$

$$\tilde{\lambda}_1 = \frac{W^2}{2}\tilde{\lambda}_5 + \frac{u}{2c}(\tilde{\lambda}_1^c - \tilde{\lambda}_2^c + 2c\tilde{\lambda}_3^c) \tag{28e}$$

5 Gradient Calculation

The gradient expression (15) is in terms of the adjoint variable and geometric gradient. The term *f* is related to the perturbation of the steady flow residual *R* to a design variable, expressed in the following way:

$$f = -\frac{\partial R}{\partial \alpha}$$

The calculation of *f* requires perturbing the mesh followed by calculating the flow residual perturbation due to this mesh perturbation.

For *n* design variables, *n* times of mesh perturbation are required. For thousands of design variables, the time cost for all the mesh perturbation is not negligible, but the time cost does not usually exceed that of a steady flow/adjoint solution. For example, the mesh needs to be perturbed once for each design variable to calculate the corresponding flow residual perturbation. Such a residual perturbation calculation has a cost that is more or less one-fourth of one full four-stage Runge–Kutta iteration. With a complex variable implementation, the cost of flow residual perturbation is doubled. Therefore for 100 design variables, the time cost will be 50 full four-stage Runge–Kutta iterations, which is only a fraction of thousands of iterations required to solve either the flow equations or the adjoint equations.

Though there are ways to avoid perturbing the mesh by converting the domain integral in Eq. (15) into a boundary integral [25,14], it usually involves calculation of spatial derivatives of flow variables along the boundary of a computational domain, which is not easy to obtain with desirable accuracy. Therefore, the gradient calculation here adopts the mesh perturbing approach.

In addition, in our computations, this residual sensitivity is obtained using the complex variable method [35]. The complex variable method is an alternative to the finite difference method but does not share the well-known disadvantage of subtraction cancellation of the finite difference method when a perturbation size becomes very small. According to the above gradient expression, boundary conditions applied when solving the flow equations should be applied in a gradient calculation.

6 Results and Discussion

6.1 Lift Coefficient Gradient. The adjoint solution provides information with which gradients of an objective function to a large number of design variables can be obtained efficiently. Thus the adjoint solution can be validated by comparing gradients using the adjoint solution with those calculated by other well established methods.

A simple means to obtain gradients of an objective function to design variables is the finite difference method. However it should be recalled that this approach suffers from step size dependence. A big step size may lead to unreliable gradient results if the relation between an objective function and design variables is highly nonlinear. The complex variable method or linearized method is a better alternative to the finite difference method. However its implementation in a large-scale nonlinear flow solver requires a lot of extra work.

As we know, within a certain range of angle of attack, the lift coefficient of an aerofoil has a linear relationship with its angle of attack. This provides a good case to calculate the gradient of the lift coefficient of an aerofoil to its angle of attack by the finite difference method by choosing a step size that is big enough to avoid significant subtraction cancellation errors.

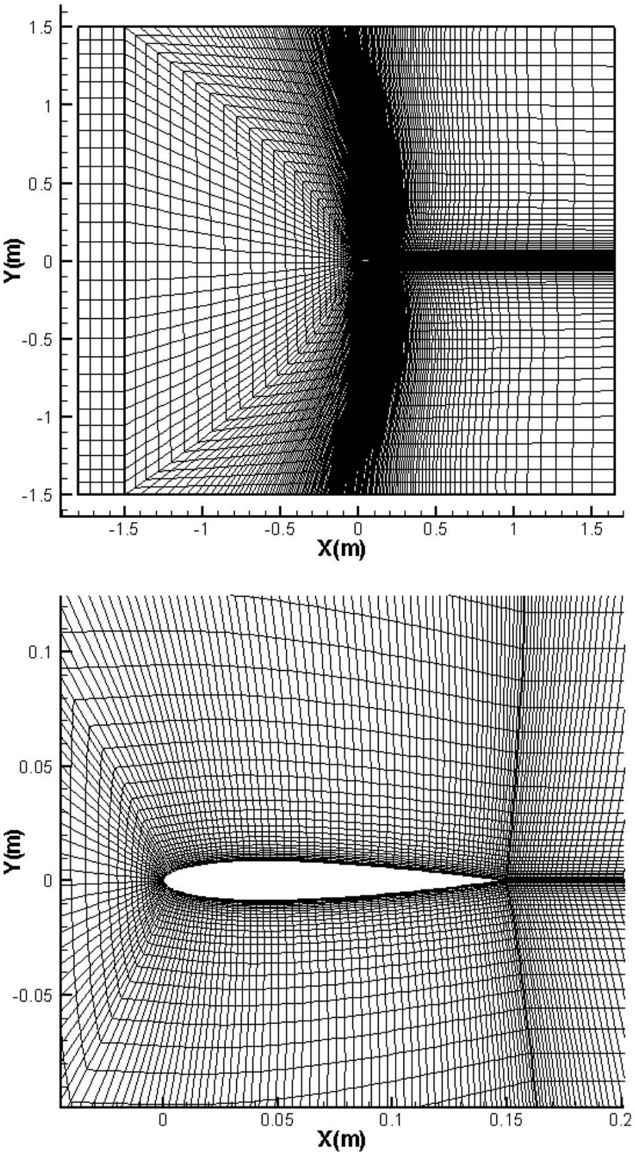

Fig. 2 Overview (upper) and close-up (lower) of the mesh around a NACA0012 aerofoil

The lift coefficient of NACA0012 with a chord length of 0.15 m is calculated for angles of attack between 0 deg and 2 deg at a Mach number of 0.62 in the far field. The computational domain is extended about ten times chord away from the aerofoil surface and is divided into four blocks with about 15,000 mesh points. Figure 2 shows the mesh around NACA0012, Fig. 3 shows the Mach number distribution, and Fig. 4 shows the adjoint solution (the adjoint variable corresponding to the continuity equation) distribution at zero angle of attack. Comparing the adjoint solution contours and those of Mach number, one can see that the upstream of the flow field corresponds to the downstream of the adjoint field, and vice versa. This is consistent with the property of the adjoint characteristics, which have an opposite direction to the flow characteristics.

The linear relationship between the lift coefficient of NACA0012 and its angle of attack is presented in Fig. 5. The gradient by the finite difference method and that by the adjoint method are in very good agreement at different angles of attack (Fig. 5). The biggest relative difference between the gradients by the two methods is within 5%.

6.2 Inverse Design. Inverse design is a popular case for adjoint method researchers to demonstrate their proper implementation of the adjoint method [22]. As a conventional inverse design, it may require the specification of a target pressure distribution on a blade surface. The pressure distribution on the surface of NACA0015 with a chord length of 0.15 m is obtained by the present flow solver at a flow condition, which is identical to that

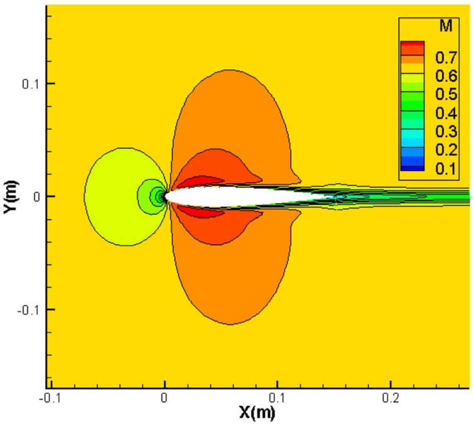

Fig. 3 Mach number contours around NACA0012 at zero angle of attack

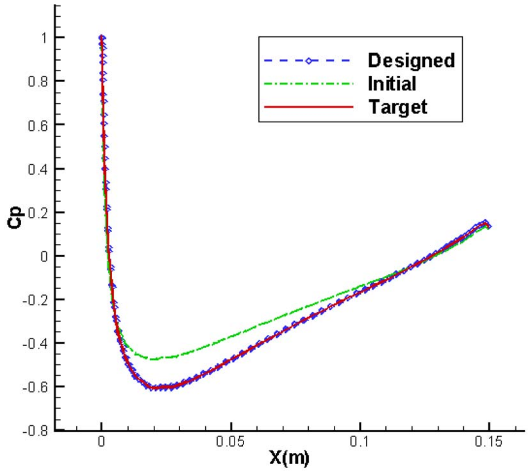

Fig. 6 Pressure coefficient distributions over the initial, target, and designed aerofoils' surface for the inverse design

for the previous NACA0012 case. This is taken as the target pressure distribution. The NACA0012 profile is taken as the initial aerofoil by the design system to approach the target pressure distribution. The objective function for this inverse design is given by

$$I = \frac{\int_{\mathrm{BS}} (p - p_t)^2 ds}{\int_{\mathrm{BS}} (p_0 - p_t)^2 ds}$$

where "BS" denotes the blade surface, p_t is the target pressure distribution, and p_0 is the pressure distribution on the NACA0012 blade surface. During the design process, the chord length of the NACA0012 aerofoil is fixed by fixing the leading and trailing points. The Hicks–Henne function as used by Yang et al. [22] is used to parametrize perturbations of y coordinates of all mesh points (except for the leading and trailing edge points) on the blade surface, resulting in 155 design variables. With the adjoint method, only two sets of equations (the flow equation and the adjoint equation) are solved with some postprocessing at each design cycle to enable the gradients of the objective function to all design variables to be calculated. However the finite different method requires the flow equation to be solved at least 156 times at each design cycle. The great time saving by the adjoint method over the finite difference method is apparent.

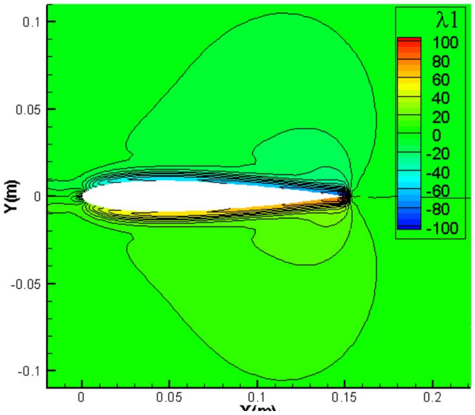

Fig. 4 Adjoint field (λ_1) around NACA0012 at zero angle of attack

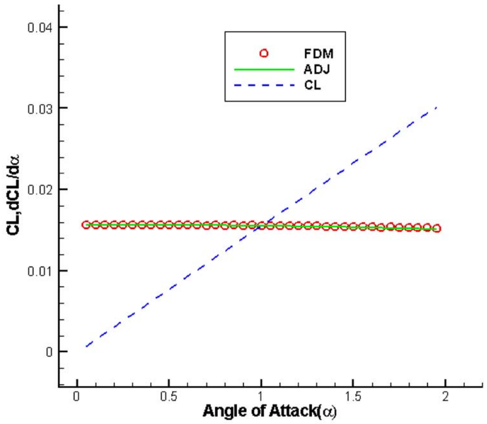

Fig. 5 Lift coefficient/gradient versus angle of attack for NACA0012 (FDM: finite difference method, ADJ: adjoint method, and CL: calculated lift coefficient)

The initial, target, and designed pressure coefficient distributions are compared in Fig. 6. The target pressure is achieved with good accuracy in the inverse design. As Fig. 7 reveals, the target aerofoil profile is recovered. Figure 8 depicts the change in the objective function with design cycles. The objective function in terms of the pressure difference is reduced by three orders of magnitude from its initial value.

6.3 Validation of Adjoint Mixing-Plane Treatment. For the verification of the adjoint mixing-plane treatment, the Euler equations are solved and the adjoint equations are based on the Euler flow solution to avoid complexity introduced by the viscous effects and freezing the turbulent viscosity. The verification case uses a span section of a 3D compressor stage—a transonic DLR (German Aerospace Center) compressor stage [37]. The section is put on a cylindrical surface at a constant radius. Figure 9 shows the mesh with a mesh density of 110×37 for the rotor domain and 90×37 for the stator domain.

The Mach number distribution is shown in Fig. 10. Figure 11 shows the contour of the adjoint variable corresponding to the continuity equation when the objective function is the tangential force on the stator blade. Figure 12 shows the contour of the adjoint variable corresponding to the moment of the momentum equation when the objective function is the mass flow rate calculated at the exit of the stator. The two adjoint solution contours

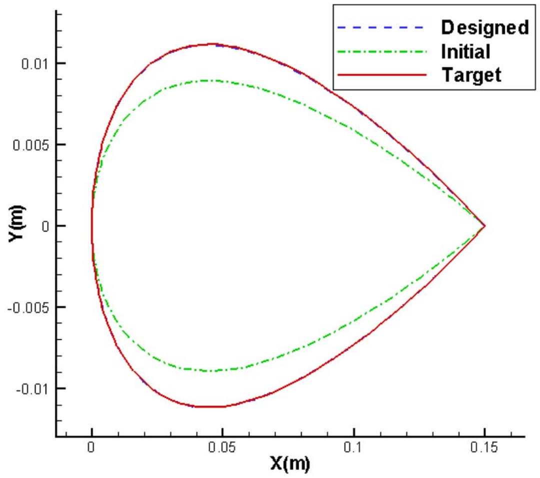

Fig. 7 Comparison between initial, target, and designed aerofoil profiles for the inverse design

consistently depict that the adjoint solution information propagates in a direction opposite to that of the physical flow disturbances. The flow upstream region becomes the downstream region of the adjoint solution, and the flow downstream region becomes the upstream region of the adjoint solution. Figure 13 compares the gradient results of the tangential force on the stator blade to design variables parametrizing the stator blade shape. The gradients by the adjoint method are very close to those by the finite difference method. The appropriate implementation of the adjoint mixing-plane treatment is further demonstrated in the good comparison of the gradient results of the mass flow rate to design variables parametrizing the rotor blade shape (Fig. 14). The mass flow rate is calculated at the exit of the stator exit (stage exit). According to the subsonic inlet boundary condition (Eq. (17)), $\partial M / \partial p$ is zero at the inlet of the stage. This means that the objective function information is applied at the stage exit only. Apart from that, the mass flow rate gradient calculation requires the rotor computational domain alone to be perturbed. Therefore the adjoint mixing-plane treatment plays a vitally important role in passing the objective function information correctly through the interface of the 2D rotor-stator configuration. Figures 11 and 12 also show that along the interface, there is a clear cut of contour lines right on both sides of the interface, which indicates clearly that the adjoint mixing-plane treatment is nonreflective.

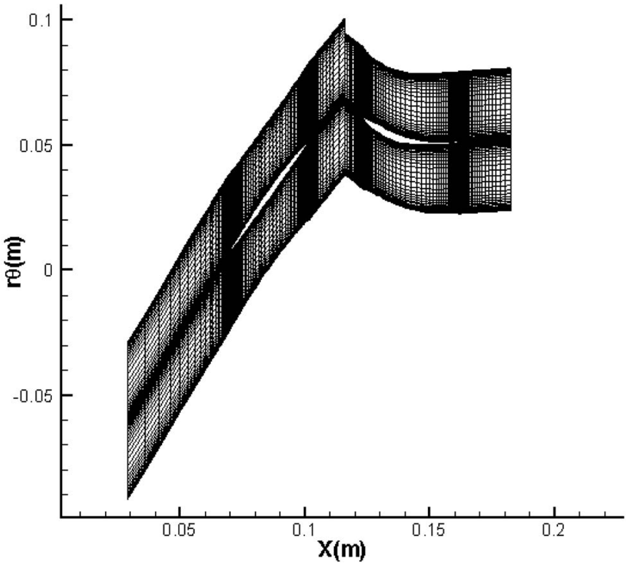

Fig. 9 Computational mesh of the 2D compressor stage

6.4 Redesign of Compressor Stage. The 2D section of the compressor stage used in Sec. 6.3 is optimized to increase the isentropic efficiency while maintaining its mass flow rate and pressure ratio. The objective function is a weighted sum of the entropy generation rate and the two constraints of mass flow rate and stagnation pressure ratio (for further details see Ref. [38]).

In this case, the RANS equations are solved. Ten design variables are used to parametrize perturbation to the rotor blade shape and eight design variables to the stator blade shape. The isentropic efficiency of the stage is increased from 84.91% to 88.00% with negligible changes in mass flow rate and pressure ratio. It should be pointed out the baseline flow state is not a very efficient one and is chosen here mainly for demonstrating the adjoint mixing-plane method. Figure 15 shows the difference of the optimized rotor blade from the original one. The optimized blade is curved toward its suction side between the leading edge and the mid-chord, leading to a weaker shock in the passage (Fig. 16). The trailing edge region is curved in an opposite direction toward the pressure side, better matched with the stator. Figure 17 shows the geometry change of the stator blade; the stator camber is increased to reduce the flow separation around the stator trailing edge

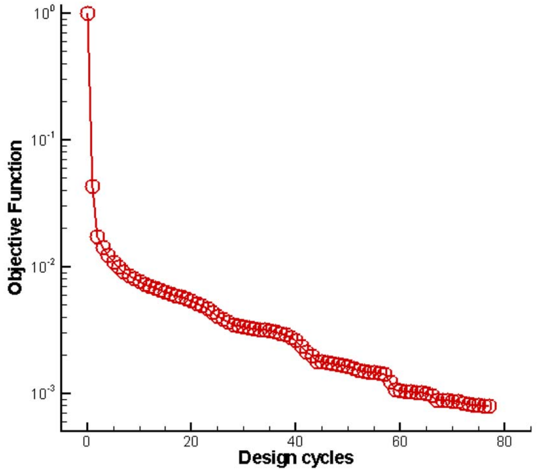

Fig. 8 History of the objective function for the inverse design

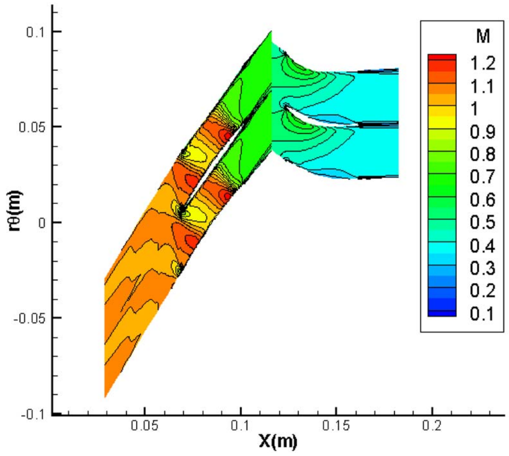

Fig. 10 Relative Mach number contours (2D configuration of the compressor stage)

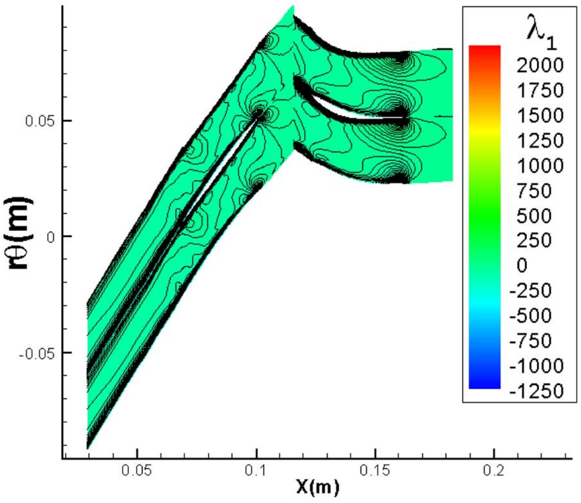

Fig. 11 Adjoint field (λ_1) with the tangential force on the stator blade as the objective function

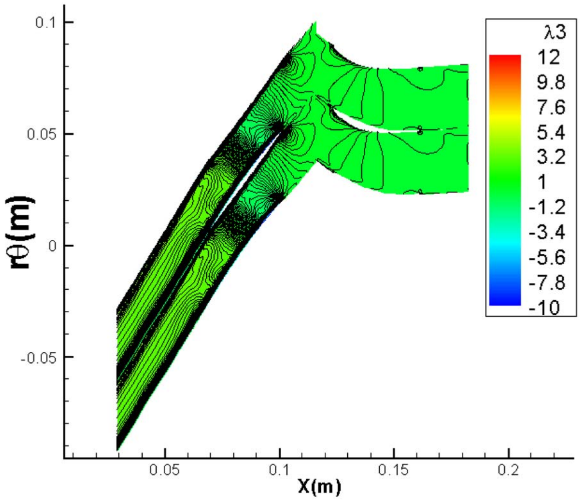

Fig. 12 Adjoint field (λ_3) with mass flow rate as the objective function

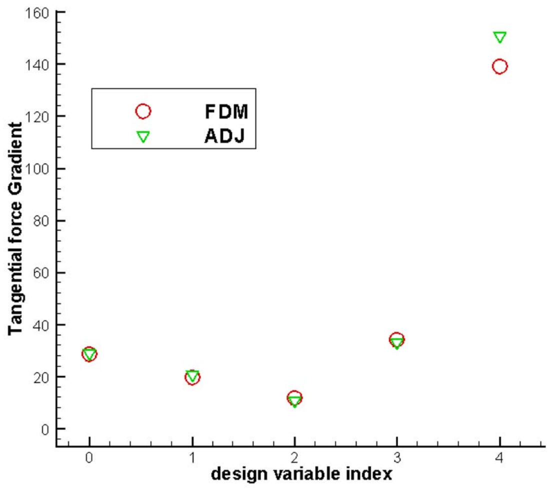

Fig. 13 Gradients of the tangential force on the stator blade (FDM: finite difference method and ADJ: adjoint method)

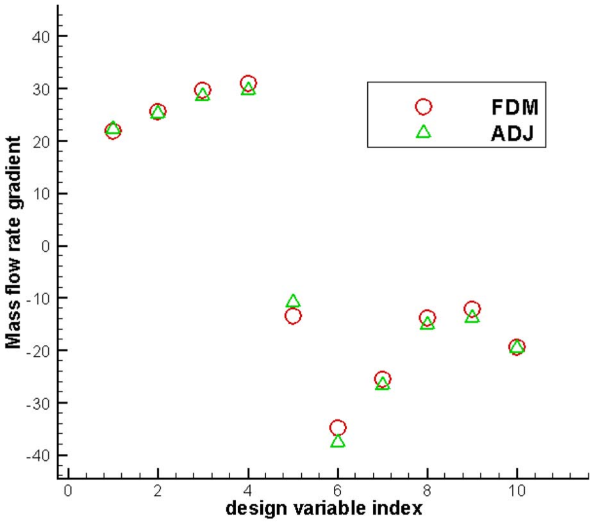

Fig. 14 Gradients of stage mass flow rate (FDM: finite difference method and ADJ: adjoint method)

(Fig. 16).

This case provides an illustration of the validity and advantage of the present adjoint mixing-plane treatment. Clearly it is far more effective to optimize both the rotor and stator at the same time than to deal with them separately and iteratively.

7 Concluding Remarks

The adjoint method for turbomachinery blading aerodynamic design optimization with the emphasis on the adjoint mixing-plane treatment is presented. It is believed that development and application of an adjoint method can benefit considerably from a clear appreciation of the antiphysics adjoint information propagation, as demonstrated in the computational results. The good agreement of the lift coefficient gradient by the adjoint method

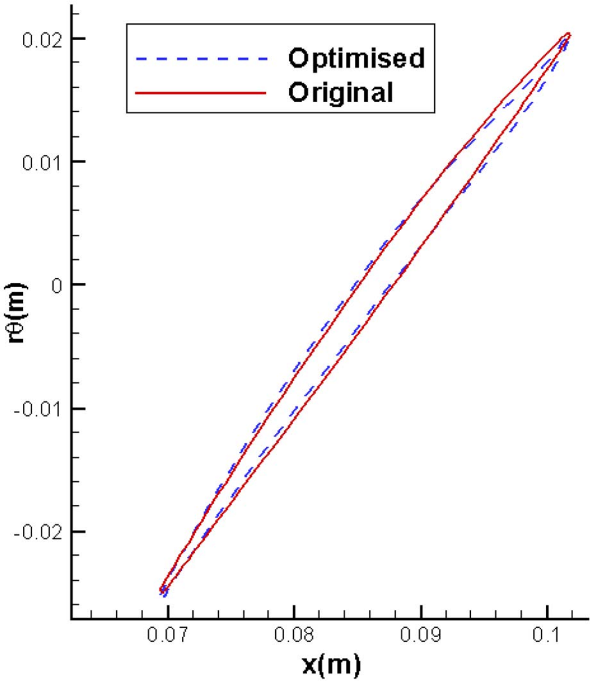

Fig. 15 Comparison of the original rotor blade and the optimized rotor blade

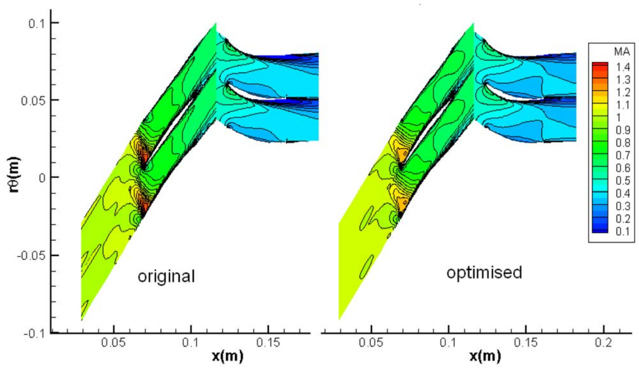

Fig. 16 Relative Mach contours of the original blade passage and the optimized blade passage

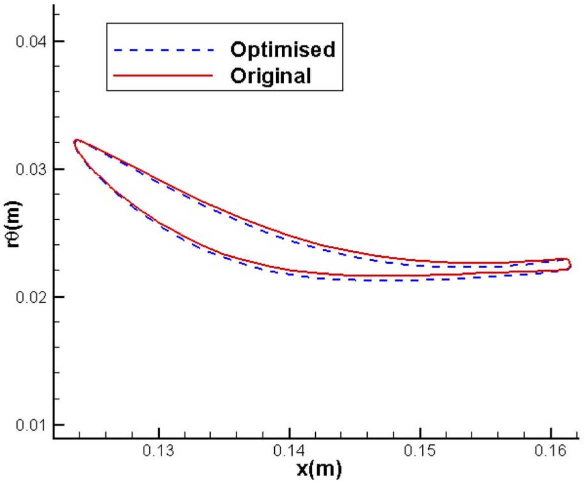

Fig. 17 Comparison of the original stator blade and the optimized stator blade

with that by the finite difference method demonstrates the validity of the adjoint method in the gradient calculation and its appropriate implementation. These are further confirmed by the inverse design well recovering the target pressure coefficient distribution and the target aerofoil profile. The close agreement of the gradient results by the adjoint method with those by the finite difference method for a 2D rotor-stator configuration case demonstrates the appropriate formulation and implementation of the adjoint mixing-plane treatment. The characteristic-based adjoint mixing-plane treatment is also shown to be nonreflective. The validity and advantage of the adjoint mixing-plane treatment is further confirmed by the optimization of the 2D compressor stage. Further validations and applications of the present approach to realistic 3D configurations in a multirow environment are presented in the companion paper [38].

Acknowledgment

The authors would like to thank Siemens Industrial Turbomachinery Ltd. for funding this work and allowing its publication. Discussions with Dr. Y.S. Li, R.G. Wells, and Dr. T. Chen at Siemens are much appreciated. The authors also would like to thank Dr. H.D. Li for his help during the early stage of the work.

Nomenclature

F,G,H = axial, circumferential, and radial flux vectors
u,v,w = axial, circumferential, and radial velocity components
S = source term of the flow equations
U = conservative flow variable vector
U_v = viscous flow variable vector
$V_{x,\theta,r}$ = axial, circumferential, and radial viscous flux vectors
ρ = fluid density
Ω = angular velocity
E = specific total energy
H = specific total enthalpy
p = pressure
τ_{ij} = $i,j{=}x,\theta,r$ shear stress
q_i = $i,j{=}x,\theta,r$ heat flux
λ = adjoint variable vector
A,B,C,D = Jacobian matrices of inviscid flux vectors and the source term to the conservative flow variable vector
A_v,B_v,C_v = Jacobian matrices of viscous flux vectors to the conservative flow variable vector
$D_{x,\theta,r}$ = adjoint viscous terms
σ_{ij} = adjoint viscous stress terms

Appendix

1 Derivation of Viscous Adjoint Equations. Linearizing the flow equations (1) with respect to a design variable, one has

$$\frac{\partial(A\tilde{U}-A_v\tilde{U}-D_{xx}\tilde{U}_x-D_{x\theta}\tilde{U}_\theta-D_{xr}\tilde{U}_r)}{\partial x}+\frac{\partial(B\tilde{U}-\tilde{U}v_g-B_v\tilde{U}-D_{\theta x}\tilde{U}_x-D_{\theta\theta}\tilde{U}_\theta-D_{\theta r}\tilde{U}_r)}{r\,\partial\,\theta}+\frac{\partial r(C\tilde{U}-C_v\tilde{U}-D_{rx}\tilde{U}_x-D_{r\theta}\tilde{U}_\theta-D_{rr}\tilde{U}_r)}{r\,\partial\,r}-D\tilde{U}=f \tag{A1}$$

where A, B, C, and D are as defined before in Eq. (11), and

$$A_v=\partial V_x/\partial U,\quad B_v=\partial V_\theta/\partial U,\quad C_v=\partial V_r/\partial U$$

$$\tilde{U}_x=\partial\tilde{U}/\partial x,\quad \tilde{U}_\theta=\partial\tilde{U}/(r\,\partial\,\theta),\quad \tilde{U}_r=\partial(r\tilde{U})/(r\,\partial\,r)$$

$$D_{ij}=\partial V_i/\partial U_j,\quad i,j=x,\theta,r$$

Multiplying Eq. (A1) with the adjoint variable and rearranging yields

$$\lambda^T\left[\frac{\partial(A\tilde{U}-A_v\tilde{U}-D_{xx}\tilde{U}_x-D_{x\theta}\tilde{U}_\theta-D_{xr}\tilde{U}_r)}{\partial x}+\frac{\partial(B\tilde{U}-\tilde{U}v_g-B_v\tilde{U}-D_{\theta x}\tilde{U}_x-D_{\theta\theta}\tilde{U}_\theta-D_{\theta r}\tilde{U}_r)}{r\,\partial\,\theta}+\frac{\partial r(C\tilde{U}-C_v\tilde{U}-D_{rx}\tilde{U}_x-D_{r\theta}\tilde{U}_\theta-D_{rr}\tilde{U}_r)}{r\,\partial\,r}-D\tilde{U}-f\right]=0 \tag{A2}$$

The linearized flow equations (Eq. (A1)) hold no matter what value the adjoint variable will take. Integrating the left hand side of the above equation over the whole computational domain, one has

$$\int_D \lambda^T\left[\frac{\partial(A\tilde{U} - A_v\tilde{U} - D_{xx}\tilde{U}_x - D_{x\theta}\tilde{U}_\theta - D_{xr}\tilde{U}_r)}{\partial x} + \frac{\partial(B\tilde{U} - \tilde{U}v_g - B_v\tilde{U} - D_{\theta x}\tilde{U}_x - D_{\theta\theta}\tilde{U}_\theta - D_{\theta r}\tilde{U}_r)}{r\,\partial\,\theta} + \frac{\partial r(C\tilde{U} - C_v\tilde{U} - D_{rx}\tilde{U}_x - D_{r\theta}\tilde{U}_\theta - D_{rr}\tilde{U}_r)}{r\,\partial\,r} - D\tilde{U} - f\right]dv = 0 \tag{A3}$$

Performing integration by parts once gives

$$\int_{\partial D}[\lambda^T(\tilde{F} - \tilde{V}_x)n_x + \lambda^T(\tilde{G} - \tilde{U}v_g - \tilde{V}_\theta)n_\theta + \lambda^T(\tilde{H} - \tilde{V}_r)n_r]ds - \int_D\left[\frac{\partial\lambda^T}{\partial x}(A - A_v) + \frac{\partial\lambda^T}{r\,\partial\,\theta}(B - v_gI - B_v) + \frac{\partial\lambda^T}{\partial r}(C - C_v) + \lambda^TD\right]\tilde{U}dv + \int_D\left[\frac{\partial\lambda^T}{\partial x}(D_{xx}\tilde{U}_x + D_{x\theta}\tilde{U}_\theta + D_{xr}\tilde{U}_r) + \frac{\partial\lambda^T}{r\,\partial\,\theta}(D_{\theta x}\tilde{U}_x + D_{\theta\theta}\tilde{U}_\theta + D_{\theta r}\tilde{U}_r) + \frac{\partial\lambda^T}{\partial r}(D_{rx}\tilde{U}_x + D_{r\theta}\tilde{U}_\theta + D_{rr}\tilde{U}_r)\right]dv - \int_D \lambda^Tfdv = 0 \tag{A4}$$

Rearranging the last second domain integral in the above by collecting like terms with $\tilde{U}_x$, $\tilde{U}_\theta$, and $\tilde{U}_r$ yields

$$\int_D\left[\left(\frac{\partial\lambda^T}{\partial x}D_{xx} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta x} + \frac{\partial\lambda^T}{\partial r}D_{rx}\right)\tilde{U}_x + \left(\frac{\partial\lambda^T}{\partial x}D_{x\theta} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta\theta} + \frac{\partial\lambda^T}{\partial r}D_{r\theta}\right)\tilde{U}_\theta + \left(\frac{\partial\lambda^T}{\partial x}D_{xr} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta r} + \frac{\partial\lambda^T}{\partial r}D_{rr}\right)\tilde{U}_r\right]dv \tag{A5}$$

Performing integration by parts to the above once gives

$$\int_{\partial D}\left[\left(\frac{\partial\lambda^T}{\partial x}D_{xx} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta x} + \frac{\partial\lambda^T}{\partial r}D_{rx}\right)\tilde{U}\cdot n_x + \left(\frac{\partial\lambda^T}{\partial x}D_{x\theta} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta\theta} + \frac{\partial\lambda^T}{\partial r}D_{r\theta}\right)\tilde{U}\cdot n_\theta + \left(\frac{\partial\lambda^T}{\partial x}D_{xr} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta r} + \frac{\partial\lambda^T}{\partial r}D_{rr}\right)\tilde{U}\cdot n_r\right]ds - \int_D\left[\frac{\partial}{\partial x}\left(\frac{\partial\lambda^T}{\partial x}D_{xx} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta x} + \frac{\partial\lambda^T}{\partial r}D_{rx}\right) + \frac{\partial}{r\,\partial\,\theta}\left(\frac{\partial\lambda^T}{\partial x}D_{x\theta} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta\theta} + \frac{\partial\lambda^T}{\partial r}D_{r\theta}\right) + \frac{\partial}{\partial r}\left(\frac{\partial\lambda^T}{\partial x}D_{xr} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta r} + \frac{\partial\lambda^T}{\partial r}D_{rr}\right)\right]\tilde{U}dv \tag{A6}$$

Substituting Eq. (A6) into Eq. (A4) and collecting like terms yields

$$\int_{\partial D}\left[\lambda^T(\tilde{F} - \tilde{V}_x)n_x + \lambda^T(\tilde{G} - \tilde{U}v_g - \tilde{V}_\theta)n_\theta + \lambda^T(\tilde{H} - \tilde{V}_r)n_r + \left(\frac{\partial\lambda^T}{\partial x}D_{xx} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta x} + \frac{\partial\lambda^T}{\partial r}D_{rx}\right)\tilde{U}\cdot n_x + \left(\frac{\partial\lambda^T}{\partial x}D_{x\theta} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta\theta} + \frac{\partial\lambda^T}{\partial r}D_{r\theta}\right)\tilde{U}\cdot n_\theta + \left(\frac{\partial\lambda^T}{\partial x}D_{xr} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta r} + \frac{\partial\lambda^T}{\partial r}D_{rr}\right)\tilde{U}\cdot n_r\right]ds - \int_D\left[\frac{\partial\lambda^T}{\partial x}(A - A_v) + \frac{\partial\lambda^T}{r\,\partial\,\theta}(B - v_gI - B_v) + \frac{\partial\lambda^T}{\partial r}(C - C_v) + \lambda^TD + \frac{\partial}{\partial x}\left(\frac{\partial\lambda^T}{\partial x}D_{xx} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta x} + \frac{\partial\lambda^T}{\partial r}D_{rx}\right) + \frac{\partial}{r\,\partial\,\theta}\left(\frac{\partial\lambda^T}{\partial x}D_{x\theta} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta\theta} + \frac{\partial\lambda^T}{\partial r}D_{r\theta}\right) + \frac{\partial}{\partial r}\left(\frac{\partial\lambda^T}{\partial x}D_{xr} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta r} + \frac{\partial\lambda^T}{\partial r}D_{rr}\right)\right]\tilde{U}dv - \int_D \lambda^Tfdv \tag{A7}$$

Adding Eq. (A7) to the gradient expression in Eq. (10) and collecting the domain integral terms with $\tilde{U}$, one has

$$-\int_D\left[\frac{\partial\lambda^T}{\partial x}(A - A_v) + \frac{\partial\lambda^T}{r\,\partial\,\theta}(B - v_gI - B_v) + \frac{\partial\lambda^T}{\partial r}(C - C_v) + \lambda^TD + \frac{\partial}{\partial x}\left(\frac{\partial\lambda^T}{\partial x}D_{xx} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta x} + \frac{\partial\lambda^T}{\partial r}D_{rx}\right) + \frac{\partial}{r\,\partial\,\theta}\left(\frac{\partial\lambda^T}{\partial x}D_{x\theta} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta\theta} + \frac{\partial\lambda^T}{\partial r}D_{r\theta}\right) + \frac{\partial}{\partial r}\left(\frac{\partial\lambda^T}{\partial x}D_{xr} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta r} + \frac{\partial\lambda^T}{\partial r}D_{rr}\right)\right]\tilde{U}dv$$

Choose the value of the adjoint variable in such a way so that the above domain integral vanishes, leading to the viscous adjoint equations,

$$\frac{\partial\lambda^T}{\partial x}(A - A_v) + \frac{\partial\lambda^T}{r\,\partial\,\theta}(B - v_gI - B_v) + \frac{\partial\lambda^T}{\partial r}(C - C_v) + \lambda^TD + \frac{\partial}{\partial x}\left(\frac{\partial\lambda^T}{\partial x}D_{xx} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta x} + \frac{\partial\lambda^T}{\partial r}D_{rx}\right) + \frac{\partial}{r\,\partial\,\theta}\left(\frac{\partial\lambda^T}{\partial x}D_{x\theta} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta\theta} + \frac{\partial\lambda^T}{\partial r}D_{r\theta}\right) + \frac{\partial}{\partial r}\left(\frac{\partial\lambda^T}{\partial x}D_{xr} + \frac{\partial\lambda^T}{r\,\partial\,\theta}D_{\theta r} + \frac{\partial\lambda^T}{\partial r}D_{rr}\right) = 0 \tag{A8a}$$

The above equation can be rearranged as follows:

$$(A - A_v)^T\frac{\partial\lambda}{\partial x} + (B - v_gI - B_v)^T\frac{\partial\lambda}{r\,\partial\,\theta} + (C - C_v)^T\frac{\partial\lambda}{\partial r} + P^T\left(\frac{\partial D_x}{\partial x} + \frac{\partial D_\theta}{r\,\partial\,\theta} + \frac{\partial D_r}{\partial r}\right) + D^T\lambda = 0$$

$$P = \frac{\partial U}{\partial U_v}, \quad U_v = (p,u,v,w,T)^T$$

$$D_x = (0,\sigma_{xx},\sigma_{\theta x}/r,\sigma_{rx},\phi_x)^T$$

$$D_\theta = (0,\sigma_{x\theta},\sigma_{\theta\theta}/r,\sigma_{r\theta},\phi_\theta)^T$$

$$D_r = (0,\sigma_{xr},\sigma_{\theta r}/r,\sigma_{rr},\phi_r)^T$$

$$\sigma_{xx} = \frac{2}{3}\mu\left[2\left(\frac{\partial\lambda_2}{\partial x} + u\frac{\partial\lambda_5}{\partial x}\right) - \left(\frac{\partial\lambda_3}{\partial\theta} + v\frac{\partial\lambda_5}{r\,\partial\,\theta}\right) - \left(\frac{\partial\lambda_4}{\partial r} + w\frac{\partial\lambda_5}{\partial r}\right)\right]$$

$$\sigma_{\theta\theta} = \frac{2}{3}\mu\left[2\left(\frac{\partial\lambda_3}{\partial\theta} + v\frac{\partial\lambda_5}{r\,\partial\,\theta}\right) - \left(\frac{\partial\lambda_2}{\partial x} + u\frac{\partial\lambda_5}{\partial x}\right) - \left(\frac{\partial\lambda_4}{\partial r} + w\frac{\partial\lambda_5}{\partial r}\right)\right]$$

$$\sigma_{rr} = \frac{2}{3}\mu\left[2\left(\frac{\partial\lambda_4}{\partial r} + w\frac{\partial\lambda_5}{\partial r}\right) - \left(\frac{\partial\lambda_3}{\partial\theta} + v\frac{\partial\lambda_5}{\partial\theta}\right) - \left(\frac{\partial\lambda_2}{\partial x} + u\frac{\partial\lambda_5}{\partial x}\right)\right]$$

$$\sigma_{x\theta} = \sigma_{\theta x} = \mu\left[\left(\frac{\partial\lambda_2}{r\,\partial\,\theta} + u\frac{\partial\lambda_5}{r\,\partial\,\theta}\right) + \left(r\frac{\partial\lambda_3}{\partial x} + v\frac{\partial\lambda_5}{\partial x}\right)\right]$$

$$\sigma_{xr} = \sigma_{rx} = \mu\left[\left(\frac{\partial\lambda_2}{\partial r} + u\frac{\partial\lambda_5}{\partial r}\right) + \left(\frac{\partial\lambda_4}{\partial x} + w\frac{\partial\lambda_5}{\partial x}\right)\right]$$

$$\sigma_{r\theta} = \sigma_{\theta r} = \mu\left[\left(\frac{\partial\lambda_4}{r\,\partial\,\theta} + w\frac{\partial\lambda_5}{r\,\partial\,\theta}\right) + \left(r\frac{\partial\lambda_3}{\partial r} + v\frac{\partial\lambda_5}{\partial r}\right)\right]$$

$$\phi_x = k\frac{\partial \lambda_5}{\partial x}, \quad \phi_\theta = k\frac{\partial \lambda_5}{r\,\partial \theta}, \quad \phi_r = k\frac{\partial \lambda_5}{\partial r} \tag{A8b}$$

2 Conservation of Adjoint Variable Weighted Fluxes. According to expression (13*b*), for two adjacent computational domains separated by an interface, each has a boundary integral term depending on the flow variable sensitivity $\tilde{U}$ along the interface. As Fig. 1 shows, if an interface is perpendicular to the *x* direction, then the boundary integral terms can be reduced as follows:

$$\frac{1}{Y_1}\int_{y1} \lambda^T A\tilde{U}dy - \frac{1}{Y_2}\int_{y2} \lambda^T A\tilde{U}dy \tag{A9}$$

The appearance of the pitch length accounts for the whole annulus effect. Expression (A9) depends on the flow variable sensitivity $\tilde{U}$. The adjoint method requires these two terms to vanish, namely,

$$\frac{1}{Y_1}\int_{y1} \lambda^T A\tilde{U}dy - \frac{1}{Y_2}\int_{y2} \lambda^T A\tilde{U}dy = 0 \tag{A10a}$$

This can be achieved in two ways. One is to make each of them vanish. This requires setting $\lambda^T A = 0$ at every point along an interface, as the flow variable sensitivity $\tilde{U}$ is not always zero along an interface. Because the Jacobian matrix *A* usually has a full rank, setting $\lambda^T A = 0$ is equivalent to setting $\lambda = 0$, which is sufficient but too much to satisfy Eq. (A10*a*). The other way is to satisfy Eq. (A10*a*) as a whole across the interface. This means that each of the two terms from the two sides of the interface may not be zero, but they equal each other and hence cancel out.

Equation (A10*a*) can be written in the following equivalent form:

$$\frac{1}{Y_1}\int_{y1} \lambda^T \tilde{F}dy - \frac{1}{Y_2}\int_{y2} \lambda^T \tilde{F}dy = 0 \tag{A10b}$$

The adjoint variable vector is independent of a design variable; therefore, one has

$$\lambda^T \tilde{F} = \widetilde{\lambda^T F}$$

Substituting the above into Eq. (A10*b*) yields

$$\frac{1}{Y_1}\int_{y1} \widetilde{\lambda^T F}dy - \frac{1}{Y_2}\int_{y2} \widetilde{\lambda^T F}dy = 0 \tag{A10c}$$

The interface geometry can be kept intact when a design variable is perturbed. Hence the differentiation operation and integration operation can exchange their sequences, resulting in the following equivalent equation:

$$\widetilde{\frac{1}{Y_1}\int_{y1} \lambda^T F dy} - \widetilde{\frac{1}{Y_2}\int_{y2} \lambda^T F dy} = 0 \tag{A10d}$$

The above equation can be further rearranged as follows by removing the differentiation operation:

$$\frac{1}{Y_1}\int_{y1} \lambda^T F dy - \frac{1}{Y_2}\int_{y2} \lambda^T F dy = 0 \tag{A11}$$

This is a scalar equation. It is satisfied by balancing all the corresponding terms across the interface, leading to the following five equations, with each one corresponding to one adjoint variable:

$$\frac{1}{Y_1}\int_{y1} \lambda_1(\rho u)dy - \frac{1}{Y_2}\int_{y2} \lambda_1(\rho u)dy = 0$$

$$\frac{1}{Y_1}\int_{y1} \lambda_2(\rho u^2 + p)dy - \frac{1}{Y_2}\int_{y2} \lambda_2(\rho u^2 + p)dy = 0$$

$$\frac{1}{Y_1}\int_{y1} \lambda_3(\rho uv)dy - \frac{1}{Y_2}\int_{y2} \lambda_3(\rho uv)dy = 0$$

$$\frac{1}{Y_1}\int_{y1} \lambda_4(\rho uw)dy - \frac{1}{Y_2}\int_{y2} \lambda_4(\rho uw)dy = 0$$

$$\frac{1}{Y_1}\int_{y1} \lambda_5(\rho Hu)dy - \frac{1}{Y_2}\int_{y2} \lambda_5(\rho Hu)dy = 0 \tag{A12}$$

The five equations in Eq. (A12) state that the adjoint variable weighted and circumferentially averaged fluxes of mass, momentum, and energy are conserved across an interface.

References

[1] Horlock, J. H., and Denton, J. D., 2005, "A Review of Some Early Design Practice Using Computational Fluid Dynamics and a Current Perspective," ASME J. Turbomach., **127**, pp. 5–13.

[2] de Vito, L., Van den Braembussche, R. A., and Deconinck, H., 2003, "A Novel Two-Dimensional Viscous Inverse Design Method for Turbomachinery Blading," ASME J. Turbomach., **125**, pp. 310–316.

[3] van Rooij, M. P. C., Dang, T. Q., and Larosiliere, L. M., 2007, "Improving Aerodynamic Matching of Axial Compressor Blading Using a Three-Dimensional Multistage Inverse Design Method," ASME J. Turbomach., **129**, pp. 108–118.

[4] Büche, D., Guidati, G., and Stoll, P., 2003, "Automated Design Optimization of Compressor Blades for Stationary, Large Scale Turbomachinery," ASME Paper No. GT2003-38421.

[5] Oyama, A., and Liou, M. S., 2004, "Transonic Axial-Flow Blade Optimization: Evolutionary Algorithms/Three-Dimensional Navier-Stokes Solver," J. Propul. Power, **20**(4), pp. 612–619.

[6] Keskin, A., Dutta, A. K., and Bestle, D., 2006, "Modern Compressor Aerodynamic Blading Process Using Multi-Objective Optimization," ASME Paper No. GT2006-90206.

[7] Yi, W., Huang, H., and Han, W., 2006, "Design Optimization of Transonic Compressor Rotor Using CFD and Genetic Algorithm," ASME Paper No. GT2006-90155.

[8] Lian, Y., and Liou, M. S., 2005, "Multi-Objective Optimization of Transonic Compressor Blade Using Evolutionary Algorithm," J. Propul. Power, **21**(6), pp. 979–987.

[9] Jameson, A., 1988, "Aerodynamic Design Via Control Theory," J. Sci. Comput., **3**(3), pp. 233–260.

[10] Nadarajah, S. K., and Jameson, A., 2000, "A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization," AIAA Paper No. 00-0667.

[11] Duta, M. C., Shahpar, S., and Giles, M. B., 2007, "Turbomachinery Design Optimization Using Automatic Differentiated Adjoint Code," ASME Paper No. GT2007-28329.

[12] Thomas, J. P., Hall, K. C., and Dowell, E. H., 2003, "A Discrete Adjoint Approach for Modelling Unsteady Aerodynamic Design Sensitivities," AIAA Paper No. 03-0041.

[13] Giles, M. B., and Pierce, N. A., 2000, "An Introduction to the Adjoint Approach to Design," Flow, Turbul. Combust., **65**(3/4), pp. 393–415.

[14] Jameson, A., 2003, "Aerodynamic Shape Optimization Using the Adjoint Method," Lectures at the Von Karman Institute, Brussels.

[15] Reuther, J., Jameson, A., Farmer, J., Martinelli, L., and Saunders, D., 1996, "Aerodynamic Shape Optimization of Complex Aircraft Configurations via an Adjoint Formulation," AIAA Paper No. 96-0094.

[16] Kim, H. J., and Nakahashi, K., 2005, "Discrete Adjoint Method for Unstructured Navier-Stokes Solver," AIAA Paper No. 05-449.

[17] Nielsen, E. J., and Anderson, W. K., 1999, "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier-Stokes Equations," AIAA J., **37**(11), pp. 1411–1419.

[18] Nadarajah, S. K., and Jameson, A., 2002, "Optimal Control of Unsteady Flows Using a Time Accurate Method," AIAA Paper No. 02-5436.

[19] Nadarajah, S. K., and Jameson, A., 2006, "Optimum Shape Design for Unsteady Three Dimensional Viscous Flows Using a Nonlinear Frequency Domain Method," AIAA Paper No. 06-3455.

[20] Arens, K., Rentrop, P., Stoll, S. O., and Wever, U., 2005, "An Adjoint Approach to Optimal Design of Turbine Blades," Appl. Numer. Math., **53**, pp. 93–105.

[21] Li, Y., Yang, D., and Feng, Z., 2006, "Inverse Problem in Aerodynamic Shape Design of Turbomachinery Blades," ASME Paper No. GT2006-91135.

[22] Yang, S., Wu, H., Liu, F., and Tsai, H., 2003, "Aerodynamic Design of Cascades by Using an Adjoint Equation Method," AIAA Paper No. 03-1068.

[23] Wu, H., Yang, S., and Liu, F., 2003, "Comparison of Three Geometric Repre-

sentation of Airfoils for Aerodynamic Optimization," AIAA Paper No. 2003-4095.

[24] Wu, H., Liu, F., and Tsai, H., 2005, "Aerodynamic Design of Turbine Blades Using an Adjoint Equation Method," AIAA Paper No. 05-1006.

[25] Papadimitriou, D. I., and Giannakoglou, K. C., 2006, "Compressor Blade Optimization Using a Continuous Adjoint Formulation," ASME Paper No. GT2006-90466.

[26] Corral, R., and Gisbert, F., 2006, "Profiled End-Wall Design Using an Adjoint Navier-Stokes Solver," ASME Paper No. GT2006-90650.

[27] Florea, R., and Hall, K. C., 2001, "Sensitivity Analysis of Unsteady Inviscid Flow Through Turbomachinery Cascades," AIAA J., **39**(6), pp. 1047–1056.

[28] Duta, M. C., Giles, M. B., and Campobasso, M. S., 2002, "The Harmonic Adjoint Approach to Unsteady Turbomachinery Design," Int. J. Numer. Methods Fluids, **40**, pp. 323–332.

[29] Denton, J. D., 1992, "The Calculation of Three-Dimensional Viscous Flow Through Multistage Turbomachines," ASME J. Turbomach., **114**, pp. 18–26.

[30] He, L., and Denton, J. D., 1994, "Three-Dimensional Time-Marching Inviscid and Viscous Solutions for Unsteady Flows Around Vibrating Blades," ASME J. Turbomach., **116**, pp. 469–476.

[31] He, L., Chen, T., Wells, R. G., Li, Y. S., and Ning, W., 2002, "Analysis of Rotor-Rotor and Stator-Stator Interferences in Multi-Stage Turbomachines," ASME J. Turbomach., **124**, pp. 564–571.

[32] He, L., and Ning, W., 1998, "An Efficient Approach for Analysis of Unsteady Viscous Flows in Turbomachines," AIAA J., **36**(11), pp. 2005–2012.

[33] Moffatt, S., Ning, W., Li, Y. S., Wells, R. G., and He, L., 2005, "Blade Forced Response Prediction for Industrial Gas Turbines," J. Propul. Power, **21**(4), pp. 707–714.

[34] Saxer, A. P., 1992, "A Numerical Analysis of 3D Inviscid Stator/Rotor Interactions Using Non-Reflecting Boundary Conditions," Gas Turbine Laboratory, Massachusetts Institute of Technology, Technical Report.

[35] Vatsa, V. N., 2000, "Computation of Sensitivity Derivatives of Navier-Stokes Equations Using Complex Variables," Adv. Eng. Software, **31**(8–9), pp. 655–659.

[36] Giles, M. B., and Pierce, N. A., 1998, "On the Properties of Solutions of the Adjoint Euler Equations," Sixth ICFD Conference on Numerical Methods for Fluid Dynamics, Oxford, UK.

[37] Dunker, R., Rechter, H., Starken, H., and Weyer, H., 1984, "Redesign and Performance Analysis of a Transonic Axial Compressor Stator and Equivalent Plane Cascades With Subsonic Controlled Diffusion Blades," ASME J. Eng. Gas Turbines Power, **106**, pp. 279–287.

[38] Wang, D. X., He, L., Li, Y. S., and Wells, R. G., 2010, "Adjoint Aerodynamic Design Optimization for Blades in Multistage Turbomachines—Part II: Validation and Application," ASME J. Turbomach. **132**(2), p. 021012.