Proceedings of ASME Turbo Expo 2011
GT2011
June 6-10, 2011, Vancouver, British Columbia, Canada

GT2011-45805

AERODYNAMIC SHAPE DESIGN OPTIMIZATION FOR TURBOMACHINERY CASCADE BASED ON DISCRETE ADJOINT METHOD

Chaolei ZHANG **Zhenping FENG***

Institute of Turbomachinery, Xi'an Jiaotong University, Xi'an 710049, P.R.CHINA
Tel/Fax: +86-29-82665062, Email: zhangchaolei@yahoo.cn, *zpfeng@mail.xjtu.edu.cn

ABSTRACT

Achieving higher aerodynamic performance in terms of efficiency, pressure ratio or stable operation range has been of interest to both researchers and engineers in the field of turbomachinery. The design of optimal shaped aerodynamic configurations based on Computational Fluid Dynamics (CFD) and predefined targets can be obtained by using deterministic search algorithms, which need to calculate the first and second order sensitivities of the objective function with respect to the design variables. With the characteristics of quick and exact sensitivity analysis, as well as less computational resource requirement, the adjoint method has become a research focus in aerodynamic shape design optimization over the past decades. In this paper, a discrete adjoint solver was developed and validated based on an in-house flow solver code. Moreover, a turbomachinery cascade optimization design system was established by coupling the flow solver, the discrete adjoint solver, the parameterization technology, the grid generation technology and the gradient-based optimization algorithms. During the development process of the discrete adjoint solver, the automatic differentiation tool was used in order to ease the construction of the discrete adjoint system based on the flow solver code. However, in order to save the memory requirement and to reduce the computational cost, the automatic differentiation tool was used selectively to build the fundamental subroutines. The top-most module of the discrete adjoint solver was established based on the discrete adjoint theory and the automatic differentiation technology manually. The treatments of the discontinuity in the flow field, such as strong shocks, and the imposition of strong boundary conditions which were implemented in the adjoint solver were discussed in detail. At the same time, several technologies were used to accelerate convergence. Based on the optimization system, a typical 2D transonic turbomachinery cascade was optimized under the viscous flow environment. The optimization results were analyzed in detail. The validity and efficiency of the present optimization design system were proved.

INTRODUCTION

The aerodynamic optimization design of turbomachinery cascades is a complex optimization problem because of its intrinsic characteristics of nonlinear, multi-variable, and multi-objective. Generally, such problems can be solved using some stochastic or deterministic search algorithms, or any hybridization of them. Stochastic optimization methods, such as Simulated Annealing (SA), Evolutionary Algorithm (EA) and so on, are well known for their ability to capture the global optimal solution without being trapped into local optima. The price to pay is the higher computational cost which can be reduced only through introducing some surrogate analysis tools and Design of Experiment (DOE) methods[1-3]. Deterministic optimization methods have characteristics of great local optima searching ability and low computational cost, but they require the sensitivity information of objective function with respect to design variables. However, the traditional sensitivity analysis methods, such as the finite differences method, the direct differentiation method, and the complex variables approach and so on, are quite costly, since their computational cost is proportional to the number of design variables. The aerodynamic optimization methods which based on the control theory implement the sensitivity analysis independent of the number of the design variables and reduce greatly the computational cost of the sensitivity analysis. Pironneau[4] first introduced the continuous adjoint method based on the control theory in fluid dynamics context but the application to aeronautical design optimization was pioneered by Jameson et al.[5-7]. Since then, the continuous adjoint method has been a major research area for its characteristics of low memory requirement and low computational cost per iteration. Moreover, it can use the mature commercial CFD solvers which reduce the development work greatly. Therefore, the continuous adjoint method has been applied widely in the aeronautical design optimization. In the area of turbomachinery aerodynamic design, the blade aerodynamic optimization based on the continuous adjoint method has been the research focus in recent years due to its complex internal flow characteristics, and several studies have been performed in the design of turbomachinery blades by using the continuous adjoint method [8-13].

In contrast with the continuous adjoint method, the discrete adjoint method starts with nonlinear discrete equations, linearizes the discrete equations, and then transposes the linear operator to form the corresponding discrete adjoint equations [14]. There are several significant advantages for the discrete adjoint method as compared with the continuous adjoint method. The construction of the discrete adjoint equations and the boundary conditions is a clear and straightforward process. The linearization of the nonlinear discrete equations can be performed either manually or by automatic differentiation software, and the linear code can be validated by directly comparing with the nonlinear code. With the great development of the automatic differentiation tools, several automatic differentiation tools can be used to substantially ease the development of the discrete adjoint code. Because the discrete adjoint code is obtained by transposing the linear operator, it yields exactly the same values for the linearized objective function and thus can be validated against the linear code[14].

The aerodynamic optimization design based on the discrete adjoint method has been paid much attention by researchers since 1990's. Elliott and Peraire[15] and Anderson and Bonhaus[16] applied the discrete adjoint method for the aerodynamic design optimization using the unstructured grid. Nadarajah[17] performed the aerodynamic design optimization of airfoil based on the discrete adjoint method and compared the calculation speed and the precision between the discrete adjoint method and the continuous adjoint method. Mohammadi et al.[18, 19] constructed the discrete adjoint system directly using automatic differentiation tool from a fluid solver and calculated the gradient information accurately. Marta et al.[20] proposed a methodology for the development of discrete adjoint solvers using the automatic differentiation tools. The aerodynamic optimization design of the turbomachinery cascade has been interested by researchers only in recent years. Florea and Hall[21] proposed a sensitivity analysis method based on the discrete adjoint method for the turbmachinery cascade unsteady inviscid flow. Giles et al.[22-24] studied the aerodynamic design optimization of the turbmachinery based on the discrete adjoint method and developed a series of design software. Papadimitriou and Giannakoglou[25] proposed an efficient calculation method of Hessian matrix by coupling direct differentiation and discrete adjoint method and applied it to the aerodynamic design optimization. Nielsen et al.[26] performed some works in the field of extending the flow to the turbulence based on the discrete adjoint method. Although some progress has been made in the area of the shape design optimization of the turbomachinery blades, the blade design optimization under the complicated constraints based on the discrete adjoint method is still worth studying because of the intrinsic complicated characteristics for the shape design optimization of the turbomachinery blades.

This paper focused on the direct design optimization using the discrete adjoint method for the turbomachinery cascade. Firstly, the principles of the discrete adjoint method and the automatic differentiation were introduced briefly. Then the adjoint field solution strategy and the sensitivity analysis method for the design optimization of the turbomahinery blades were discussed in detail. Based on the principle of the discrete adjoint method and the adjoint field solution strategy, a discrete adjoint solver was developed and validated with an in-house flow solver code by using selectively TAPENADE[27]. After that, an aerodynamic shape design optimization system based on the discrete adjoint method for the turbomachinery blades in two-dimensional (2D) viscous flows was proposed and established by coupling the parameterization technology based on the Non-uniform B-Spline, the structured multi-block grid generation technology, the in-house CFD solver and the discrete adjoint solver developed above. Using the design optimization system, a typical 2D transonic turbomachinery cascade was optimized under the viscous flow environment. The minimization of the entropy generation rate was taken as the objective function in the design optimization process and a mass flow rate constraint was added at the same time. The optimization results were analyzed in detail, and the validity and efficiency of the optimization design system were proved too.

NOMENCLATURE

A	Matrix
C	Coefficient
f	Scalar function
I	Augmented objective function
J	Objective function
L	Linear discrete operator
P	Non-singular square matrix
R	Nonlinear discrete residual operator
s	Entropy generation ratio
u	State variables
x	Coordinates
y	Coordinates
α	Design variables
ψ	Adjoint variables
ε	Step size

Subscripts

ref	Reference state

Superscripts

T	Transpose

DISCRETE ADJOINT METHOD

A typical aerodynamic shape design optimization problem for turbomachinery cascades minimizes an objective function J with respect to the parameterized cascade shape parameters α. It can be described as follows:

$$\begin{cases} J(\alpha,u) \to \min\limits_{\alpha\in\Lambda} \\ \text{s.t. } R(\alpha,u)=0, \text{ on } \Omega_\Gamma \end{cases} \tag{1}$$

where, J is the objective function and R is the state equations respectively. Both of which are function of the design variables α and the state variables u. In the equation, the “s.t.” indicates that design variables and state variables are constrained by the state equation R at the same time. Linearizing the above objective function, the variation of the objective function J can be formulated as：

$$\delta J = \frac{\partial J}{\partial \alpha}\delta\alpha + \frac{\partial J}{\partial u}\delta u = \left(\frac{\partial J}{\partial \alpha} + \frac{\partial J}{\partial u}\frac{\partial u}{\partial \alpha}\right)\delta\alpha \tag{2}$$

Linearizing the state equation gives sensitivity equation as follows：

$$\frac{\partial R}{\partial \alpha}\delta\alpha + \frac{\partial R}{\partial u}\delta u = 0 \tag{3}$$

Based on the discrete adjoint method, introducing adjoint variables $\psi = \{\psi_1,\psi_2,\cdots,\psi_n\}^T$, an augmented function $I(\alpha,u)$ can be defined as:

$$I(\alpha,u) = J(\alpha,u) + \psi^T R(\alpha,u) \tag{4}$$

The variation of the augmented objective function $I(\alpha,u)$ with respect to the design variables can be expressed as follows:

$$\begin{aligned} \delta I &= \frac{\partial J}{\partial \alpha}\delta\alpha + \frac{\partial J}{\partial u}\delta u + \psi^T\left(\frac{\partial R}{\partial \alpha}\delta\alpha + \frac{\partial R}{\partial u}\delta u\right) \\ &= \left(\frac{\partial J}{\partial \alpha} + \psi^T\frac{\partial R}{\partial \alpha}\right)\delta\alpha + \left(\frac{\partial J}{\partial u} + \psi^T\frac{\partial R}{\partial u}\right)\delta u \end{aligned} \tag{5}$$

Since the variation of the state equation is zero, the derivatives of the $I(\alpha,u)$ and of the $J(\alpha,u)$ are identical. The dependence of the augmented objective function on the state variable can be removed by setting the coefficient of δu to zero. Thus the adjoint equation can be obtained. Introducing a pseudo-time term, the adjoint equation can be formulated as:

$$\frac{\partial \psi}{\partial t} + \left(\frac{\partial R}{\partial u}\right)^T \psi = -\left(\frac{\partial J}{\partial u}\right)^T \tag{6}$$

The adjoint equation does not contain any derivatives with respect to the design variable. Therefore the computational cost of solving the adjoint equation is nearly independent of the number of the design variables. Once the adjoint solution has been obtained, the gradient of the augmented objective function with respect to the design variables can be computed from the following formula:

$$\frac{\delta I}{\delta \alpha} = \frac{\partial J}{\partial \alpha} + \psi^T\frac{\partial R}{\partial \alpha} \tag{7}$$

According to the above equations, it can be seen that $\partial J/\partial u$, $\partial J/\partial \alpha$, $\psi^T(\partial R/\partial \alpha)$ and $\psi^T(\partial R/\partial u)$ should be solved firstly in order to calculate the gradient of the objective function with respect to the design variables. The item $\psi^T(\partial R/\partial u)$ should be solved iteratively but other items are cheaper to evaluate because they do not require any iterative solution and involve only matrix-vector products.

AUTOMATIC DIFFERENTIATION

Automatic differentiation, also known as algorithmic differentiation, is a well known technology which is implemented by using systematically the chain rule of differentiation to computer programs. Then a program can be generated automatically which can be used to compute the derivatives specified by user. With the development of the automatic differentiation technology and associated automatic differentiation tools, several automatic differentiation tools for different program languages are available at present. Among the automatic differentiation tools for different programming languages, there are two main approaches to implement the automatic differentiation: source code transformation and operator overloading[28]. The source code transformation method adds new statements to the original source code in order to generate the program unit used for derivatives calculation. The operator overloading approach defines a new user defined data type instead of the real numbers. This new data type includes not only the value of the original variable, but the derivative as well. All the intrinsic operations and functions have to be redefined for the derivative to be calculated together with the original computations.

Automatic differentiation can be performed in two modes which are forward mode and reverse mode, respectively. Both modes employ the chain rule to accumulate contributions to derivatives in a different manner. The implementation details about the automatic differentiation can refer to Ref [29]. To sum up, the forward mode gives the tangential derivative $\nabla f \cdot \vec{a}$ for some given vector $\vec{a}$, while the reverse mode gives all the components of ∇f for a scalar function f. For a vector function F, the forward mode automatic different gives $\nabla F \cdot \vec{a}$ while the reverse mode gives $(\nabla F)^T \cdot \vec{a}$. According to the discussion in the previous section, the $\partial J/\partial u$, $\partial J/\partial \alpha$, $\psi^T(\partial R/\partial \alpha)$ can be calculated by the program unit generated by using the reverse mode of the automatic differentiation from the original flow solver with the adjoint field information. While the calculation of the $\psi^T(\partial R/\partial u)$ will be discussed in detail in the following section.

SENSITIVITY ANALYSIS

For a scalar function $I(\alpha,u)$ to be optimized, its calculation process can be formulated as follows

$$\alpha \;\to\; x \;\to\; u \;\to\; I \tag{8}$$

where α is a set of design parameters, x is the computational grid, and u is the discrete flow solution.

In order to use a gradient-based optimization method, one wishes to compute the derivative of I with respect to α. Adopting the notation used in the automatic differentiation community, let $\dot{\alpha}$, $\dot{x}$, $\dot{u}$, $\dot{I}$ denote the derivative with respect to one particular component of α, respectively.

If at each stage in the process the output is an explicit function of the input, then straight forward differentiation gives

$$\dot{x} = \frac{\partial x}{\partial \alpha}\dot{\alpha}\ ,\quad \dot{u} = \frac{\partial u}{\partial x}\dot{x}\ ,\quad \dot{I} = \frac{\partial I}{\partial u}\dot{u} \tag{9}$$

and according to the chain rule of differentiation,

$$\dot{I} = \frac{\partial I}{\partial u}\frac{\partial u}{\partial x}\frac{\partial x}{\partial \alpha}\dot{\alpha} \tag{10}$$

Again following the notation used in the automatic differentiation community, the adjoint quantities $\overline{\alpha}$, $\overline{x}$, $\overline{u}$, $\overline{I}$ denote the derivatives of I with respect to α, x, u, I, respectively, with $\overline{I}=1$ by definition. Differentiating again, and with a superscript "T" denoting a matrix or vector transposing, we obtains

$$\overline{\alpha} = \left(\frac{\partial I}{\partial \alpha}\right)^T = \left(\frac{\partial I}{\partial x}\frac{\partial x}{\partial \alpha}\right)^T = \left(\frac{\partial x}{\partial \alpha}\right)^T \overline{x} \tag{11}$$

and similarly

$$\overline{x} = \left(\frac{\partial u}{\partial x}\right)^T \overline{u}\ ,\quad \overline{u} = \left(\frac{\partial I}{\partial u}\right)^T \overline{I} \tag{12}$$

thus giving

$$\overline{\alpha} = \left(\frac{\partial x}{\partial \alpha}\right)^T \left(\frac{\partial u}{\partial x}\right)^T \left(\frac{\partial I}{\partial u}\right)^T \overline{I} \tag{13}$$

Note that the linear sensitivity analysis proceeds forwards through the process

$$\dot{\alpha} \ \rightarrow\ \dot{x} \ \rightarrow\ \dot{u} \ \rightarrow\ \dot{I} \tag{14}$$

While the adjoint analysis proceeds backwards

$$\overline{\alpha} \ \leftarrow\ \overline{x} \ \leftarrow\ \overline{u} \ \leftarrow\ \overline{I} \tag{15}$$

Given these definitions, the sensitivity of the output I with respect to the inputs α can be evaluated in a number of ways as follows

$$\dot{I} = \overline{u}^T\dot{u} = \overline{x}^T\dot{x} = \overline{\alpha}^T\dot{\alpha} \tag{16}$$

Therefore, it is possible to proceed forwards through part of the process and combine this with going backwards through the other part of the process. This is useful in applications in which part of the process is a black-box which cannot be touched. For example, if the step $\alpha \rightarrow x$ involves a proprietary CAD system or a grid generator, then the only option may be to approximate the forward mode linear sensitivity through a central finite difference using $x(\alpha \pm \Delta\alpha)$.

SOLUTION STRATEGY OF ADJOINT SYSTEM

The construction of the discrete adjoint system for the aerodynamic optimization design of turbomachinery cascades has its special characteristics. The construction method and solution strategy of the discrete adjoint system will be discussed in detail in the following section. According to the discussion in the automatic differentiation section, the items of $\partial J/\partial u$, $\partial J/\partial \alpha$, $\psi^T(\partial R/\partial \alpha)$ can be calculated by using program units which are generated by using the reverse mode of the automatic differentiation from the original flow solver if the adjoint field have been obtained. This section shows how to calculate the $\psi^T(\partial R/\partial u)$ in detail.

The flow solution u is not an explicit function of the grid coordinates x, but instead it is defined implicitly through the solution of a set of non-linear discrete flow equation

$$R(u,x) = 0 \tag{17}$$

To solve these equations, many CFD algorithms use iterative methods which can be written as

$$u^{n+1} = u^n - P(u^n, x)R(u^n, x) \tag{18}$$

where P is a non-singular square matrix which is a differentiable function of its arguments. If P is defined to be L^{-1}

$$L = \frac{\partial R}{\partial u} \tag{19}$$

where L is the non-singular Jacobian matrix.

Linearising Eq.(17) gives

$$L\dot{U} + \dot{R} = 0 \tag{20}$$

where $\dot{R}$ is defined as

$$\dot{R} = \frac{\partial R}{\partial x}\dot{x} \tag{21}$$

with both derivatives being evaluated based on the implicitly-defined baseline solution $u(x)$.

Differentiating Eq.(18) around a fully-converged baseline solution in which $u^n = u$ gives

$$\dot{u}^{n+1} = \dot{u}^n - P\left(L\dot{u} + \dot{R}\right) \tag{22}$$

with P based on $u(x)$. This will converge to the solution of Eq.(20) with exactly the same terminal rate of convergence as the non-linear iteration.

Since

$$\dot{u} = -L^{-1}\dot{R} \tag{23}$$

the adjoint sensitivities satisfy the equation

$$\overline{R} = -(L^T)^{-1}\overline{u} \tag{24}$$

which implies that

$$L^T\overline{R} + \overline{u} = 0 \tag{25}$$

This equation can be solved iteratively using the adjoint iteration

$$\overline{R}^{n+1} = \overline{R}^n - P^T\left(L^T\overline{R}^n + \overline{u}\right) \tag{26}$$

where $\overline{u}$ is the $\partial I/\partial u$ and $L^T\overline{R}^n$ is the transposed results of the $\psi^T(\partial R/\partial u)$. From the above discussion we can learn that the physical significance of the adjoint variables is the derivative of the objective function respect to the residual of the state equations[14].

It can be seen from Eq.(26), the derivative of the objective function with respect to the numerical solution keeps constant for a converged flow field. Therefore it needs to calculate only once. In order to solve the adjoint field, following procedure can be applied: initializes the adjoint variables $\overline{R}=0$, establishes the discrete adjoint module to calculate the $L^T\overline{R}^n$, combines the above $\partial I/\partial u$ and $L^T\overline{R}^n$ to form the final residual, and updates the adjoint field by using the iterative method until reaching the final convergence.

In order to get the gradient information of the objective function with respect to the design variables after getting the adjoint field, we can calculate the gradient as following

$$\dot{I}=\overline{x}^T\dot{x} \tag{27}$$

In Eq.(27), the variation of the coordinates is obtained by using a proprietary grid generator based on the central finite difference method in the present work. The step size used to calculate the variation of the coordinates is 1E-6 m. The gradient of the objective function with respect to the design variables can be obtained by iterating through the design variables.

IMPLEMENTATION AND VALIDATION

Development of discrete adjoint code

The discrete adjoint solver was established based on the in-house flow solver code. In order to reduce the requirements of the computational cost and memory, the top-most module of the discrete adjoint was constructed according to the principles of the automatic differentiation and the solution strategy of the discrete adjoint system manually. The fundamental adjoint subroutines were constructed from the corresponding ones in the flow solver by using the automatic differentiation tool with the reverse mode. The fundamental subroutines included those which were used to construct state information at the interface of the grid, and to calculate the inviscid flux, viscous flux, gradient information of the state variables and so on. Because the limiter subroutines in the flow solver were also used to generate the corresponding adjoint limiter subroutine, the discontinuity in the adjoint field could be dealt with the adjoint solver. The adjoint boundary conditions subroutines were directly generated from the corresponding boundary conditions subroutine in the flow solver without any special treatment. However, the calling order of the fundamental adjoint subroutines generated by the automatic differentiation tool was arranged carefully according to the principle of the automatic differentiation with the reverse mode.

As a result, the requirement of the memory of the adjoint solver was increased by 120 percent as compared with that of the original CFD solver. The computational cost of the discrete adjoint solver was about 2.5 time of that associated with the original flow solver.

Verification of adjoint solver

The verification of the adjoint solver was performed based on the following method. Firstly, suppose the adjoint code evaluate

$$(\partial R/\partial u)^T\psi=A^T\psi\ ,\quad A(u)\equiv\frac{\partial R(u)}{\partial u} \tag{28}$$

According to the follow identity

$$x^TAy=y^TA^Tx \tag{29}$$

for any two vector x and y, an approximation to Ay can be obtained by using finite differences for some small $\varepsilon>0$.

$$Ay\approx\frac{1}{\varepsilon}\left[R(u+\varepsilon y)-R(u)\right] \tag{30}$$

The correctness of the discrete adjoint code generated using the automatic differentiation tools can be verified using the dot-product test method as following:

(a) Choose a random vector y whose elements lie in (-1, +1) and a small step size $\varepsilon>0$, compute

$$t=\frac{1}{\varepsilon}\left[R(u+\varepsilon y)-R(u)\right] \tag{31}$$

(b) Set $x=1$ and compute $s=A^{\mathrm{T}}x$ using the discrete adjoint code;

(c) Verify the $x^{\mathrm{T}}t\approx y^{\mathrm{T}}s$.

Since t is a finite difference approximation, we have to try out different values of ε to get a good agreement between the two values in the last step. In present work, several step sizes from 1E-10 to 1E-6 were used to validate the adjoint solver code.

OPTIMIZATION SYSTEM

Parameterization

The parameterization of the turbomachinery cascade was implemented based on the Non-uniform B-Spline library. The pressure side and the suction side of the profile for a 2D turbomachinery cascade were splitted firstly, and then the parameterization of the pressure side and the suction side were performed respectively. The distribution of the control points along the parameterization curve was controlled through setting the several stretch factors.

Grid Generation

A multi-block grid generator for 2D turbomachinery cascade was implemented based on the Transfinite Interpolation (TFI) method. The H-Grid topology was used for the leading and the trailing regions, while the O-Grid topology was used for the blade region. In order to improve the quality of the grid, a Laplace smoothing functionality was appended. The output formats supported by the grid generator included the PLOT3D, TECPLOT and so on.

Flow Solver

An in-house flow field solver was used to perform the numerical simulation of the turbomachinery cascade. In order to evaluate the objective function and to satisfy the requirement of generating the discrete adjoint solver based on the previous theory by using the automatic differentiation tool, the flow solver was adjusted properly, and validated against several commercial CFD solvers firstly. The solver was implemented based on the finite volume method with a multi-block structured grid. The non-dimensional compressible Reynolds Averaged N-S (RANS) equations were solved. The state variables were stored at the cell center. The information exchange between the blocks was implemented by defining interface boundary conditions in advance. The discretisation of the space and the time adopted the method of lines[30]. The inviscid convective fluxes were calculated based on Roe scheme of the flux-difference splitting. The reconstruction of the left and right state variables at the face of cell was implemented based on the Van Leer's MUSCL (Monotone Upstream-Centered Schemes for Conservation Laws) approach[30]. Moreover, the MUSCL interpolation was enhanced by introducing several limiters. The viscous fluxes in the governing equations were evaluated from variables averaged at the faces of the control volume. The first derivatives of the velocity components and of the temperature were accomplished based on the Green's theorem by constructing of an auxiliary control volume. The temporal discretisation of the governing equations contained the multi-stage Runge-Kutta explicit scheme and the three factor approximate factorization implicit scheme. Some turbulence models, such as $k-\varepsilon$ and $k-\omega$ two equation turbulence models, were implemented in the CFD solver. In order to accelerate convergence, several technologies, such as local time-stepping, residual smoothing and multi-grid, were implemented.

Discrete Adjoint Field Solver

The discrete adjoint field solver was implemented based on the above in-house flow field solver by using selectively the automatic differentiation tool Tapenade. Currently, the multi-stage Runge-Kutta explicit scheme and the three factor approximate factorization implicit scheme were implemented to calculate the adjoint field. Several acceleration technologies, such as local time step, multi-grid method, were implemented in the discrete adjoint solver in order to accelerate convergence. In the present work, the three factor approximate factorization implicit scheme was used because of its faster computation speed. Currently there wasn't turbulence model included in the adjoint field solver. Further work will introduce turbulence model to improve the solver farther.

Optimization Flow Chart

The whole optimization flow chart is shown in Fig. 1. The optimization algorithm used in the present work was the conjugate gradient optimization algorithm. After obtaining the flow field solution, the discrete adjoint solver read the solution and started to evaluate the adjoint field iteratively. The sensitivity information was obtained by coupling the adjoint field solution and the variation of the coordinates calculated with the center-difference method.

NUMERICAL EXAMPLES

In order to verify the validity and efficiency of the optimization design system established in this paper, a typical 2D transonic turbine cascade was optimized under the viscous flow environment based on the proposed optimization system. The objective function was the entropy generation ratio through the cascade passage. At the same time, a mass flow ratio constraint was added in order to keep a constant mass flow ratio. The working condition of the cascade is shown in Table 1.

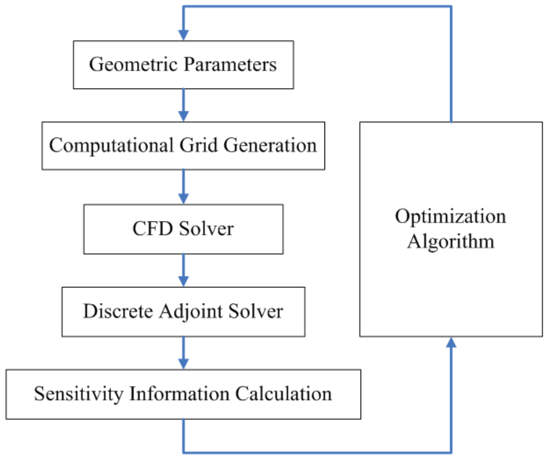


Fig. 1 Optimization flow chart

Tab. 1 Working condition of the turbine cascade

Fluid	Perfect air
Blade number	30
Middle diameter (m)	0.24
Inlet total temperature (K)	709.0
Inlet total pressure (Pa)	344,000.0
Inlet flow angle (deg)	0
Outlet static pressure (Pa)	172,370.0

The 2D cascade profile was divided into suction side and pressure side, and each side was parameterized by using Non-uniform B-Spline. The number of the control points used to parameterize the suction side and the pressure side was 39 and 34, respectively. In order to simplify the optimization process, we chose to optimize the suction side and selected 10 control points of the Non-uniform B-Spline as the design control points, with the leading and trailing edge geometry fixed. The distribution of the control points on both pressure side and suction side is shown in Fig. 2. The control points marked with filled red square were the adjustable control points during the optimization process, which were labeled from the leading edge to the trailing edge with 1 to 10 respectively. The coordinates of the adjustable control points were taken as the design variables. The total number of the design variables was 20.

A 2D H-O-H multi-block structured grid of 9,467 nodes was generated by a multi-block grid generator based on the TFI method and Laplace smoothing technology. The computational grid generated is shown in Fig. 3. It can be seen from Fig. 8 that the wake and shock wave are clearly visible, which indicates that the grid quality satisfies the requirement of the flow solver.

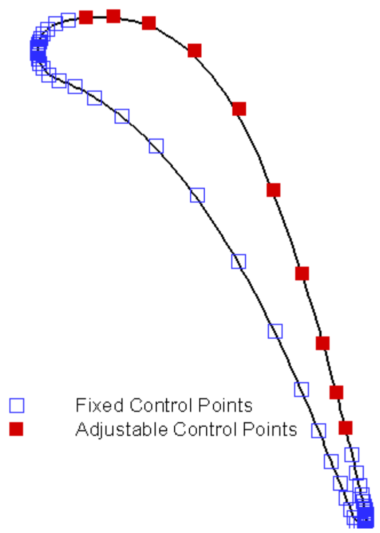


Fig. 2 Blade parameterization method

The numerical simulation of the flow field was performed by solving the steady non-dimensional compressible N-S equation with the in-house CFD solver. The boundary conditions were given as follows: the total pressure, the total temperature and the axial flow direction were given at the inlet; the static pressure was given at the outlet. The desired convergent target of each simulation was that the root mean square residuals of the momentum and mass equations, energy equation reached 1E-6. The adjoint field was analyzed based on the discrete adjoint solver developed in the present work. The convergence target of each simulation was the root mean square residuals of the adjoint equations reached 1E-5. The gradient of the objective function was calculated using the discrete adjoint method described previously.

The objective function was defined as follows:

$$J = \frac{\left(\dfrac{\int_{out} -\frac{1}{\gamma}\log(p_t)\,\rho\vec{u}\cdot d\vec{A}}{\int_{out}\rho\vec{u}\cdot d\vec{A}} - \dfrac{\int_{in}\left(-\frac{1}{\gamma}\log(p_t)\,\rho\vec{u}\cdot d\vec{A}\right)}{\int_{in}\rho\vec{u}\cdot d\vec{A}} \right)}{\Delta s_{ref}} + C\left| \frac{\int_{out}\rho\vec{u}\cdot d\vec{A}}{m_{ref}} - 1 \right| \tag{32}$$

where, C is the weight coefficient of the mass flux constraint, and here $C = 300$, Δs_{ref} is the reference entropy generation ratio from the inlet to the outlet of the cascade, which was set as 0.0002 during the optimization process. After the initial iterative step, the gradient components of the objective function J with respect to the 20 design variables is shown in Fig. 4. The gradient components of the objective function with respect to the 20 design variables calculated by using the center finite difference method with step size of 1E-6 m is given in the figure in order to verify the validity of the adjoint solver. The comparison result indicates almost the same trend of the gradients computed by using the adjoint method and the center finite difference method, respectively. It indicates the validity of the current discrete adjoint solver. It can be seen from the figure, the largest partial gradient occurs at the adjustable control points 6, 7 and 8, which just locates at the throat of the cascade passage. It indicates that the most losses locate at the cascade throat at the initial step.

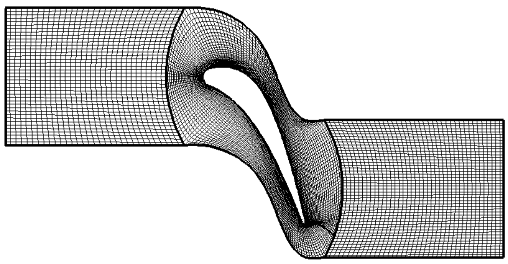


Fig. 3 Computational grids

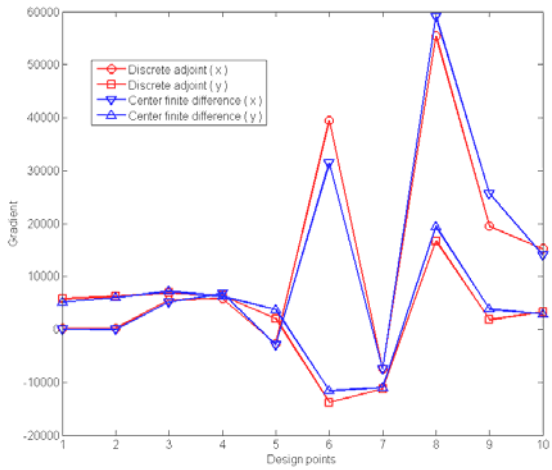
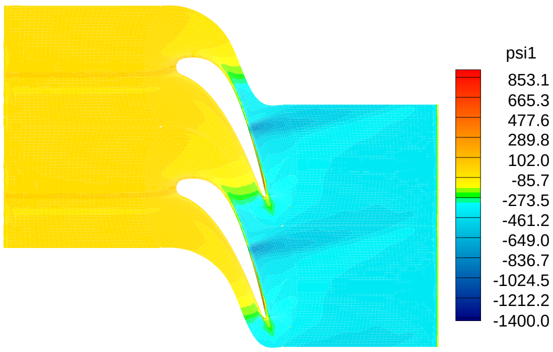
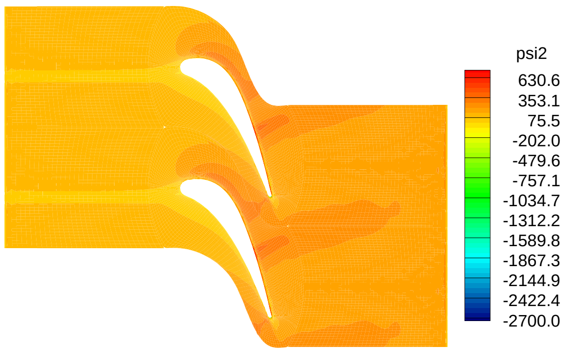


Fig. 4 Gradient components at the first iterative step

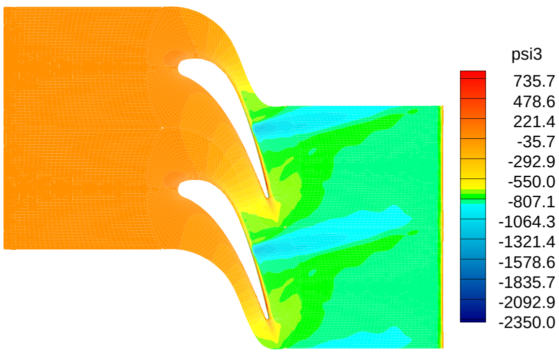
The adjoint field at the initial iterative step is shown in Fig. 5. The variables of ψ_1 and ψ_4 represent the adjoint density and density pressure, respectively. While the variables ψ_2 and ψ_3 represent the adjoint velocities of velocities u and v, respectively. The absolute value of the adjoint variable indicates level of the sensitivity according to the meaning of the adjoint variable. From the distribution of the four adjoint variables, we can see there is a high value region near the passage throat. The sensitivity of the design variables near the throat region should be higher as compared with design variables in other region, which has been verified from the gradient information in Fig. 4.



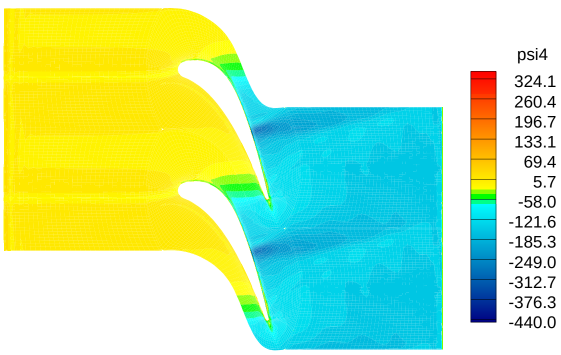
a) Distribution of ψ_1



b) Distribution of ψ_2



c) Distribution of ψ_3



d) Distribution of ψ_4

Fig. 5 Adjoint fields at the initial iterative step

The objective function value and the mass flow ratio were non-dimensional with respect to the reference values which were given in advance. Figure 6 shows the convergence history of the normalized objective function and the change history of the normalized mass flow ratio through the passage with relative to the initial values of the cascade for the sake of succinctness. It indicates that the optimization process gets convergence by 29 iterative steps and takes 88 times flow field computation with 20 design variables.

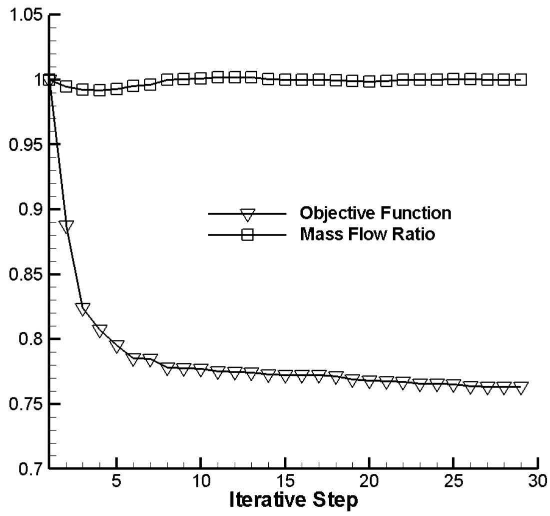


Fig. 6 Convergence history

The comparison between the profiles corresponding to the initial cascade and the optimized one is shown in Fig. 7. The thickness of the optimized cascade profile in the regions after the leading edge and after the throat is reduced a little as compared with that of the initial cascade profile.

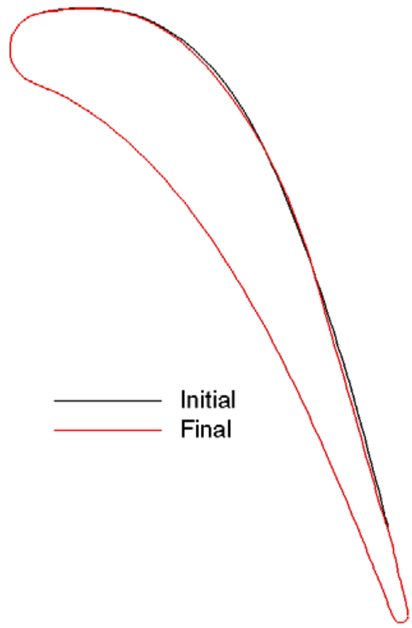
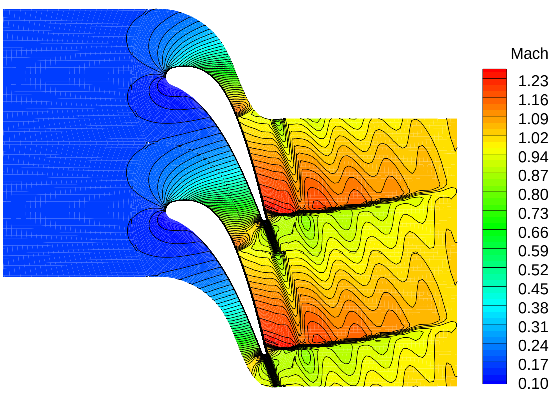


Fig. 7 Comparison of cascade profiles

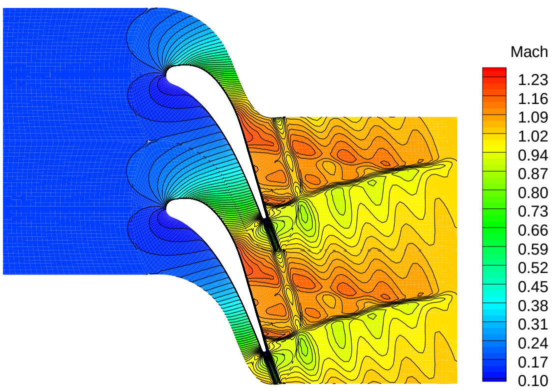
The comparison of the entropic generation ratio, the mass flow ratio of the cascades and the objective function value are listed in Table 2. The entropic generation ratio was reduced by 23.6%, while the mass flow ratio through the final cascade passage was only reduced by 0.001% as compared with the values correlated to the initial cascade. Figure 8 gives a comparison of the Mach number contours in blade passage. From the figure we can see that the optimization system is able to weaken the shock wave strength in the trailing edge. The shock wave strength in the region after the throat is enhanced as compared with that of the initial cascade.

Tab. 2 Comparison of objective function and mass flow ratio

Items	Initial	Final	Changed
Mass flow ratio	6.23236	6.23230	-0.001%
Entropy generation ratio	3.067E-2	2.341E-2	-23.66%
Objective function	153.47	117.16	-23.66%



a) initial cascade



b) final cascade

Fig. 8 Comparison of the Mach number contours

The entropy generation ratio along the pitch at the outlet of the computational domain is shown in Fig. 9. It can be seen from the figure, the entropy generation ratio at the outlet of the final cascade in the region between the 0-25% of the pitch and the 50%-100% of the pitch is reduced as compared with that of the initial cascade. The area covered by the curve and the y-axis indicates the total entropy generation ratio from the inlet to the outlet of the cascade. It can also be seen from the figure, the area is reduced a lot as compared with that of the initial cascade.

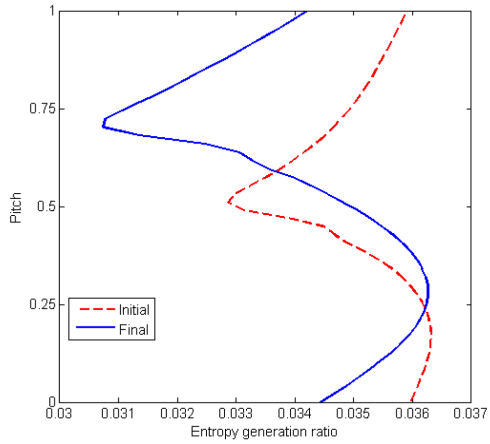


Fig. 9 Comparison of entropy generation ratio

CONCLUSION

This paper focused on the study of direct design optimization by using the discrete adjoint method for the turbomachinery cascade. An aerodynamic shape design optimization system based on the discrete adjoint method for the turbomachinery blades in 2D viscous flows environment was established and validated, by coupling up the parameterization technology based on Non-uniform B-Spline technology, the structured multi-block grid generation technology, the in-house CFD solver and the discrete adjoint solver. A typical 2D transonic turbine cascade was optimized under the viscous flow environment. The minimization of the entropy generation rate was taken as the objective function and a mass flow rate constraint was added.

According to the optimization result, the thickness of the optimized cascade profile in the regions after the leading edge and after the throat is reduced a little with unchanged throat diameter. The shock wave strength at the trailing edge is weakened as compared with that of the initial cascade profile. The entropic generation ratio was reduced by 23.6%, while the mass flow ratio through the final cascade passage was only reduced by 0.001% as compared with the initial cascade. The validity and efficiency of the optimization design system were proved.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China, Project No.51076121.

REFERENCES

[1] Jin, Y. C., 2003, “A Comprehensive Survey of Fitness Approximation in Evolutionary Computation,” Soft

Computing, 9(1), pp: 3-12.
[2] Wang, G. G., Shan, S., 2006, "Review of Meta modeling Techniques in Support of Engineering Design Optimization," ASME J. Mechanical Design, 129(4), pp: 370-380.
[3] Giannakoglou, K. C., 2002, "Design of Optimal Aerodynamic Shapes Using Stochastic Optimization Method and Computational Intelligence," Progress in Aerospace Science, 38(1), pp: 43-76.
[4] Pironneau, O., 1973, "On Optimum Shapes in Stokes Flow," Journal of Fluid Mechanics, 59(2), pp: 117-128.
[5] Jameson, A., 1988, "Aerodynamic Design via Control Theory," ICASE Technical Report 88-64.
[6] Jameson, A., Pierce, N. A., Martinelli, L., 1997, "Optimum Aerodynamic Design Using the Navier-Stokes Equations," AIAA Aerospace Sciences Meeting & Exhibit, 35th, Reno, NV, USA.
[7] Reuther J, Jameson A, Farmer J, Martinelli L, Saunders D., 1996, "Aerodynamic Shape Optimization of Complex Aircraft Configurations via an Adjoint Formulation," AIAA Paper, 96-0094.
[8] Yang, S., Wu, H., Liu, F., 2003, "Aerodynamic Design of Cascades by Using an Adjoint Equation Method," AIAA Paper, AIAA-2003-1068.
[9] Arens, K., Rentrop, P., Stoll, S. O., et al., 2005, "An Adjoint Approach to Optimal Design of Turbine Blades," Applied Numerical Mathematics, 53, pp.93-105.
[10] Li, Yingchen, Feng, Zhenping, 2008, "Three-Dimensional Aerodynamic Design of Turbine Blades Using the Adjoint Method," ASME Paper GT2008-51225.
[11] Li, Haitao, Song, Liming, Li, Yingchen, Feng, Zhenping, 2009, "2D Viscous Aerodynamic Shape Design Optimization for Turbine Blades Based on Adjoint Method," ASME Paper GT2009-59999, also ASME J. Turbomachinery, 2011, 133: 031014
[12] Wang, D. X., He, L., 2010, "Adjoint Aerodynamic Design Optimization for Blades in Multi-Stage Turbomachines: Part I," ASME J. Turbomachinery, 132: 021011.
[13] Wang, D. X., He, L., Li, Y. S., et al., 2010, "Adjoint Aerodynamic Design Optimization for Blades in Multi-Stage Turbomachines: Part II," ASME J. Turbomachinery, 132: 021012.
[14] Giles, M. B., Duta, M. C., 2003, "Algorithm Developments for Discrete Adjoint Methods," AIAA Journal, 41(2), pp: 198-205.
[15] Elliott, J., Peraire, J., 1997, "Practical 3D Aerodynamic Design and Optimization Using Unstructured Meshes," AIAA Journal, 35(9), pp: 1479-1485.
[16] Anderson, W. K., Bonhaus, D. L., 1999, "Airfoil Design on Unstructured Grids for Turbulent Flows," AIAA Journal, 37(2), pp: 185-191.
[17] Nadarajah, S. K., 2003, "The Discrete Adjoint Approach to Aerodynamic Shape Optimization," Department of Aeronautics and Astronautics of Standford University.
[18] Hafez, M., Mohammadi, B., Pironneau, O., 1996, "Optimum Shape Design Using Automatic Differentiation in Reverse Mode," Int. Conf. Numerical Meth. Fluid Dynamics, Monterey.
[19] Mohammadi, B., 1997, "Optimal Shape Design, Reverse Mode of Automatic Differentiation and Turbulence," AIAA Paper, 97-0099.
[20] Marta, A. C., Mader, C. A., Martins, J. R. R. A., et al, 2007, "A Methodology for the Development of Discrete Adjoint Solvers Using Automatic Differentiation Tools," International Journal of Computational Fluid Dynamics, 21(9-10), pp: 307-327.
[21] Florea, R., Hall, K. C., 2001, "Sensitivity Analysis of Unsteady Inviscid Flow through Turbomachinery Cascades," AIAA Journal, 39(6), pp: 1047-1056.
[22] Giles, M. B, Pierce, N. A., 1997, "Adjoint Equations in CFD: Duality, Boundary Conditions and Solution Behaviour," AIAA Paper, 97-1850.
[23] Duta, M. C., Giles, M. B., Campobasso, M. S., 2002, "The Harmonic Adjoint Approach to Unsteady Turbomachinery Design," International Journal for Numerical Methods in Fluids, 3(3-4), pp: 323-332.
[24] Campobasso, M. S., Duta, M. C., Giles, M. B., 2003, "Adjoint Calculation of Sensitivities of Turbomachinery Objective Functions," AIAA Journal of Propulsion and Power, 19(4), pp: 693-703.
[25] Papadimitriou, D. I., Giannakoglou, K. C., 2008, "Direct, Adjoint and Mixed Approaches for the Computation of Hessian in Airfoil Design Problems," International Journal for Numerical Methods in Fluids, 56(10), pp: 1929-1943.
[26] Nielsen, E. J., Lu, J., Park, M. A., Darmofal, D. L., 2004, "An Implicit Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids," Computers and Fluids, 33(9), pp: 1131-1155.
[27] Hascoët L, Pascual V., 2004, "Tapenade 2.1 user's guide," INRIA, http://www.inria.fr/rrrt/rt-0300.html.
[28] Martines, J. R. R. A., 2006, "An Automated Approach for Developing Discrete Adjoint Solvers," AIAA Paper, AIAA 2006-1608.
[29] Papadimitriou, D. I., Giannakoglou, K. C., 2008, "Aerodynamic Shape Optimization Using First and Second Order Adjoint and Direct Approaches," Arch Comput Methods Eng, 15, pp: 447-488.
[30] Blazek, J. 2001, "Computational Fluid Dynamics: Principles and Applications," Elsevier Science Ltd.

