INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2010; **62**:1332–1356
Published online 11 May 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.2070

Aerodynamic shape optimization on overset grids using the adjoint method

Wei Liao[1,2,*,†] and Her Mann Tsai[1]

[1]*Temasek Laboratories, National University of Singapore, Singapore 117508, Singapore*
[2]*Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA 23529, U.S.A.*

SUMMARY

This paper deals with the use of the continuous adjoint equation for aerodynamic shape optimization of complex configurations with overset grids methods. While the use of overset grid eases the grid generation process, the non-trivial task of ensuring communication between overlapping grids needs careful attention. This need is effectively addressed by using a practically useful technique known as the implicit hole cutting (IHC) method. The method depends on a simple cell selection process based on the criterion of cell size, and all grid points including interior points and fringe points are treated indiscriminately in the computation of the flow field. This paper demonstrates the simplicity of the IHC method for the adjoint equation. Similar to the flow solver, the adjoint equations are solved on conventional point-matched and overlapped grids within a multi-block framework. Parallel computing with message passing interface is also used to improve the overall efficiency of the optimization process. The method is successfully demonstrated in several two- and a three-dimensional shape optimization cases for both external and internal flow problems. Copyright © 2009 John Wiley & Sons, Ltd.

Received 11 November 2008; Revised 19 February 2009; Accepted 24 March 2009

KEY WORDS: aerodynamics; optimization design; adjoint equation; overset grid; Implicit Hole Cutting

1. INTRODUCTION

For engineering optimization, the choice of the method used depends on the cost of evaluating the objective function, the availability of gradient information and robustness of the optimization technique. Among optimization techniques, there is the general distinction of gradient-based and

*Correspondence to: Wei Liao, Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA 23529, U.S.A.
†E-mail: wliao@odu.edu

non-gradient-based methods. In gradient-based methods, gradient information can be found using finite difference methods [1, 2] or adjoint solutions [3, 4]. Non-gradient-based methods include direct search method and stochastic method, among which evolutionary algorithms [5, 6] have gained considerable attention in recent years. Such methods typically involve hundreds or even thousands of analysis to locate a near optimal solution even for fairly simple cases. Even if the use of surrogate models leads to a considerable reduction of computational cost, gradient-based optimization algorithms usually outperform evolutionary algorithms in terms of convergence speed [7]. The primary concern is the computational expense of the function evaluations, especially those that involve large-scale computations.

For gradient-based optimization methods, the calculation of cost function gradients with respect to the design variables or parameters that defines the shape is the main challenge. The traditional method of finite difference [1, 2] achieves gradients by directly perturbing the design variables used to define the geometry deformation, in which the number of flow solutions is proportional to the number of design variables. This method can be extremely expensive in practical applications involving large number of design variables. In contrast, the adjoint method produces gradient information without the cost increasing with the number of design parameters. The spectacular success and efficiency of this approach have been well demonstrated by Jameson [3, 4] for complex aerodynamic shape design problems. Using techniques of control theory, the gradients of the cost function are indirectly determined by solving the adjoint equations, which have coefficients determined from the flow equations. Because of the similarity of the adjoint equations to the flow equations, the same numerical methods, which are efficient for the solution of the flow equations, can also be effectively used for the adjoint equations. The cost for solving the adjoint equations will be approximately the same as the cost for solving the flow equations. Thus, the true advantage is that the new design can be determined with roughly the computational cost of two flow solutions but independent of the number of design variables involved.

However, the issues involved in computing the aerodynamics around complex configurations must also be effectively handled if the adjoint method is to be used for realistic engineering applications. To handle topologically complex problems such as a full aircraft configuration, a multi-block approach is preferred over the single-block approach. The multi-block approach seeks to break a complex computational domain into smaller regions each of which can be represented by relatively simple grids. Such a multi-block approach also depends on whether zonal boundaries exactly align to one another or arbitrarily overlap each other. The former is conventionally termed matched multi-block grid and the latter the overset or Chimera grid. The overset grid approach allows for arbitrary boundary interface, in which the grids can be generated independently for different zones or components, and grid overlaps are allowed. This very significantly eases the grid generation process compared with the matched grid approach. Generally, the overset method provides great flexibility to handle topologically complex configurations [8–12].

To exploit the flexibility of overset methods, two additional issues need to be addressed. The first is the major hurdle to creating the data structure that specifies the interconnectivity among the overset grids. The other is how to implement the proper communication among overlapping multi-block grids to ensure that the computation does not suffer from any artifacts arising from the multi-block decomposition. While simple at first sight, the generation of the grid connectivity for a system of overlapping grids is an expensive and daunting task, although some existing connectivity codes for overset grid (such as PEGSUS [10], OVERTURE [11] and FASTRAN [12]) have

DOI: 10.1002/fld

attempted to ameliorate this. These codes are usually long and involve some logically complicated procedures, such as automatic hole cutting, ADT operations, overlap optimization, projection of overlapping viscous grid surfaces and other steps.

A recent approach that efficiently addressed such complications in overset methods is the implicit hole cutting (IHC) method as introduced by Lee and Baeder [8]. This approach is a cell selection process based on the main criterion of cell size, and all grid points including hole interior points and hole fringe points are treated indiscriminately in the flow computation. In contrast to the hole cutting method in traditional overset codes and methods, this approach chooses the grids based on cell size when comparing the grid overlapping regions and the hole cutting process around bodies is a by-product of this operation. The key advantages of this approach lie in the simplicity of the concept and its implementation, in contrast to the careful book keeping approach of the common methods used in traditional overset grid codes. Potentially, these features also ease the implementation of the multi-grid algorithms, which are rarely used in overset methods. The use of IHC method as an inter-grid communication method within a hybrid multi-block framework amenable for the implementation of a multi-grid method and parallel computation has previously been demonstrated by the authors [9]. The convergence, accuracy and efficiency particularly for large-scale computations of realistic aerodynamic configurations have been well discussed in [9]. As shown in [9], the oversetting process of the volume grids with 1.6 million cells for a three-dimensional case was efficiently performed in only 105 s on a single Intel Xeon CPU 2.80GHz. It is expected that the overset grid techniques may also show distinct advantages when applied to realistic complex design optimization problems. An important advantage of the overset grid technique from an aerodynamic shape design standpoint is that since the grid for each aerodynamic component is generated independently, the need for regenerating grids during the shape modifications process is thus significantly more straightforward.

In recent years, significant progress is made in aerodynamic design optimization using adjoint equation method based on unstructured grid [13, 14] and Cartesian grid [15, 16] in handling complex configuration challenges. Because of the above-mentioned advantages of overset methods, demonstration of the use of overset grid method for solving the adjoint equations to handle shape optimization problems is also reported in the literature [17–19]. Elliott [17] demonstrated the use of a discrete adjoint analysis by extending the direct sensitivity solver based on the OVERFLOW connectivity code [17] for two-dimensional problems. As it is based on the OVERFLOW concept in handling overset grid, the same challenges are faced for the adjoint solver. Lee and Kim [18] further demonstrated the discrete adjoint formulation to optimize three-dimensional configurations based on the traditional overset grid concept used in the PEGSUS code.

As for the choice of adjoint method used, Kim *et al.* [20] recently concluded that gradients from the continuous adjoint method are in good agreement with those computed by finite difference methods. Whether to use the continuous or discrete adjoint approach, there seems to be indications that the continuous adjoint formulation works best for the interior of the domain and the discrete adjoint for the boundary conditions [21].

For the present work, the benefits of the IHC overset grids method are demonstrated for the solution of the continuous adjoint equation. As noted above, unlike the other overset methods, the present scheme would greatly ease the organization and inclusion of the adjoint solver and thus facilitate the use of the adjoint method for engineering shape optimization. The remainder of the paper is organized as follows. The formulations of the adjoint equations used are outlined in the Section 2, where the IHC method is also included in the discussion. In Section 3 the concept is

DOI: 10.1002/fld

illustrated via four test cases covering both inverse design and drag minimization. The conclusion that includes an outlook is in Section 4.

2. NUMERICAL METHODS

2.1. Governing equations for fluid motion

The governing flow equations can be written in the usual integral form

$$\frac{\partial}{\partial t}\iiint_{\Omega} w\,\mathrm{d}V+\iint_{\partial\Omega} f\cdot\mathbf{n}\,\mathrm{d}B=0 \tag{1}$$

where Ω is an arbitrary control volume with the closed boundary surface B and $\mathbf{n}$ is the unit normal vector in outward direction. The vector of state variables w is defined as

$$w=(\rho,\rho u_1,\rho u_2,\rho u_3,\rho E)^{\mathrm{T}}$$

where ρ is the density, u_1, u_2 and u_3 are the three Cartesian velocity components, E is the total energy of the flow and T denotes the transpose. The flux tensor f consists of a convective part for an inviscid flow and an additive diffusive part for a viscous flow.

2.2. Summary of the design problem

The flow characteristics are functions of the flow variables w and the physical boundary of the body, which may be represented by a function F. Here, w is not independent of F. The design objective is measured by a cost function, which is defined by some aerodynamic properties such as the drag coefficient for a drag minimization problem or the deviation from a given pressure distribution for an inverse design problem. The adjoint equations can be derived following the general approach proposed by Jameson [22].

In general, the cost function I can be defined by

$$I=I(w,F) \tag{2}$$

A change in the geometry F results in a change for the cost function I

$$\delta I=\frac{\partial I^{\mathrm{T}}}{\partial w}\delta w+\frac{\partial I^{\mathrm{T}}}{\partial F}\delta F \tag{3}$$

where T denotes the transpose.

From control theory, the governing equations of the flow field can be introduced as a constraint so that the final expression for the gradient does not require reevaluation of the flow field. Generally, the governing equation R, which expresses the dependence of w and F within the flow field domain Ω, can be written as

$$R(w,F)=0 \tag{4}$$

Thus, δw can be determined from the equation

$$\delta R=\left[\frac{\partial R}{\partial w}\right]\delta w+\left[\frac{\partial R}{\partial F}\right]\delta F=0 \tag{5}$$

DOI: 10.1002/fld

By introducing a Lagrange multiplier ψ, we get

$$\delta I = \frac{\partial I^{\mathrm{T}}}{\partial w}\delta w + \frac{\partial I^{\mathrm{T}}}{\partial F}\delta F - \psi^{\mathrm{T}}\left(\left[\frac{\partial R}{\partial w}\right]\delta w + \left[\frac{\partial R}{\partial F}\right]\delta F\right)$$

$$= \left\{\frac{\partial I^{\mathrm{T}}}{\partial w} - \psi^{\mathrm{T}}\left[\frac{\partial R}{\partial w}\right]\right\}\delta w + \left\{\frac{\partial I^{\mathrm{T}}}{\partial F} - \psi^{\mathrm{T}}\left[\frac{\partial R}{\partial F}\right]\right\}\delta F \tag{6}$$

In order to eliminate the explicit dependence of δI on δw, ψ has to satisfy the following adjoint equation:

$$\frac{\partial I}{\partial w} = \left[\frac{\partial R}{\partial w}\right]^{\mathrm{T}}\psi \tag{7}$$

With the first term in δI eliminated, we have

$$\delta I = G\delta F \tag{8}$$

where

$$G = \frac{\partial I^{\mathrm{T}}}{\partial F} - \psi^{\mathrm{T}}\left[\frac{\partial R}{\partial F}\right] \tag{9}$$

Equation (8) relates the cost variation δI directly with shape variation δF. Since δI is independent of δw, the gradient of I with respect to an arbitrary number of design variables can be determined without the need for any additional flow field evaluations. Once the adjoint equation is solved, the gradient direction can be determined. The process of computing the gradient can be repeated to follow the direction provided by some searching method, such as the steepest descent direction, until a minimum is reached.

The above general procedure can be applied to the Euler equations and the specific cost functions. A detailed description of the derivation for the cases of inverse design and drag minimization with constraint conditions is presented in the Appendix. From the derivation, we can notice that if the cost function is different, the only changes for the derived formula are the boundary conditions for the adjoint equations and the formula for the computation of δI. The adjoint equation maintains the same form.

2.3. Space discretization and time integration for adjoint equation

The properties of the adjoint equations are very similar to the flow equations. Thus the same numerical methods, which have been proven to be efficient for the solution of the flow equations, can be applied to solve the adjoint equations efficiently as well. In the present study, only steady-state solutions are considered. A time marching method is used to solve both the flow equations and adjoint equations.

The governing equations for fluid motion and the adjoint equations are solved by the well-known Jameson–Schmidt–Turkel (JST) scheme [23, 24], which is a finite volume method with central differencing and artificial dissipation. In this method, the solution domain is divided into small hexagonal cells by joining the cell vertices by straight lines. In the JST scheme, a blend of second-order and fourth-order differences provides first-order dissipation around shocks and third-order dissipation in smooth flow region. A multi-stage Runge–Kutta scheme [23] is used

DOI: 10.1002/fld

for the time integration. In order to increase the stability range and relax the restriction on the Courant–Friedrichs–Lewy number so that a larger time step can be used, implicit residual smoothing is employed. In this study, the five-stage Runge–Kutta scheme is implemented with residual smoothing applied at stage 1, 3 and 5 [25].

More details for these numerical techniques can be found in the References [23–25]. As previously noted above, the same numerical methods are applied to solve the adjoint equations and flow equations.

2.4. Overset grid strategy for adjoint equation solver

The present study exploits the benefits and straightforwardness of the IHC method [8, 9] for the solution of the continuous adjoint equation on overset grids. For this work, the conventional point-matched grids and overset grids are combined within a single framework. The IHC method is used to efficiently get the grid connectivity among the overlapping blocks of the grid in the hybrid multi-block framework. Details are found in References [8, 9]. However, for completeness the method is outlined here.

The method is a cell selection process principally based on the criterion of cell size. In the traditional overset grid usage, the concept of 'donor' and 'receiver' is used to refer the points involved in the transfer of inter-grid information. Here, the donor cell for a receiver point in one grid is identified as the cell on another grid that contains the receiver point. The 'hole cutting effect' of the wall is felt using a simple but important observation that cell size or density is progressively smaller toward the wall. The fundamental hole cutting algorithms for the IHC method are based on three basic steps as follows:

(1) A *quick first-order inside/outside cell test*. The test that checks whether a point is inside or outside a cell is required in every step of the donor search. Generally, a quick cross and dot product test is used to identify the likely donor cell of the current test point.
(2) *High-order inside/outside cell test*. Once the cross product test has identified a cell as a likely donor, an unambiguous test is required for the determination of computational coordinates (η) of the test point within the cell. Usually, the linear or the higher-order Hermit cubic [26] coordinate transform is applied between the physical coordinate $\mathbf{x}$ and computational coordinate η. Here, Newton iteration is used to determine the value of η in the transformed space. Convergence of η to a value within the range 0 to 1 indicates that the test points are inside the cell. This cell becomes a candidate for the optimum donor of the current test point. In most cases, the linear transform provides satisfactory accuracy for reduced computing cost.
(3) *Donor cell selection based on the cell size*. In the case of multiple donor cells, the one with the smallest cell size (or other cell properties) may be selected as the optimum donor cell.

In order to efficiently implement the IHC method in the current hybrid multi-block framework, a few important concepts are introduced. First, we introduce the concept of cluster. A cluster is usually generated around one component, such as a wing or a body. In practice, each cluster can be generated for one geometric component independent of other components. Certainly, a cluster can also be an arbitrary collection of matched grid blocks that cover a certain region of the flow field. A cluster is usually a matched multi-block grid, which consists of one or more blocks with matched boundaries between neighboring blocks. When implementing the IHC method, only the grid blocks in different clusters are needed to check the overlapping relationship. The value at

DOI: 10.1002/fld

a receiver point in one cluster will be interpolated from the donor cell in another cluster. The next concept is that in order to efficiently get the grid connectivity among the grid blocks in different clusters, all the boundaries of a block are classified as the physical boundary, 'inner boundary' and 'outer boundary'. The physical boundary will impose the physical boundary conditions for the flow solver, such as wall boundary, far-field boundary and inlet or outlet boundary. The 'inner boundary' is the boundary that exactly connected the matched grid blocks and also where the boundary points will exchange the information through two layer of halo cells. The outer boundary is different from the above two types of boundaries. In the current hybrid multi-block framework, the 'outer boundary' is just the overset boundary. The boundaries of a whole cluster only consist of the physical boundary and 'outer boundary'. In this study, along each 'outer boundary', two-layer halo cells are introduced to join the grid connectivity and become the hole fringe points in order to ensure the proper transfer among the grid blocks.

In this work, the IHC method is developed to accommodate the cell-centered finite volume scheme, which is used in the adjoint equation solver and flow solver. In the present method used, the receiver points are allocated at cell center and the stencil points consist of the grid vertices in the corresponding donor cells. All points, whether they fall inside or outside a solid body, will be treated indiscriminately.

Generally, the information transfer from a donor cell to a receiver cell is completed by an interpolation algorithm. A simple interpolation method is to directly transfer the flow variables from the center points of donor cells to the receiver points. However, the errors introduced by the simple interpolation may lead to a poor convergence for the adjoint equation. In the current method, a first-order trilinear interpolation scheme is used with little additional computational cost, in which four (for two-dimension) or eight (for three-dimension) vertices of the donor cell form the interpolation stencil points for the receiver point. The values at the vertices can be obtained by averaging the values at the surrounding cell centers. The variables at the receiver point can then be easily obtained by using a trilinear interpolation over this set of stencil points. Results of the test cases shown in the present work are obtained by using the trilinear interpolation scheme as follows:

$$P_{\eta_1,\eta_2,\eta_3} = \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} \eta_1^i (1-\eta_1^{1-i})\eta_2^j (1-\eta_2^{1-j})\eta_3^k (1-\eta_3^{1-k})\bar{P}_{i,j,k} \tag{10}$$

which is consistent with a globally second-order scheme used in the current solver. Here, P_{η_1,η_2,η_3} is the function value at interpolated point and $\bar{P}_{i,j,k}$ are the function values at the stencil points. (η_1, η_2, η_3) are the values of the computational coordinate η in three dimensions.

In this study, with the use of the IHC method, the computational coordinates η in Equation (10) have been determined by Newton's method during the oversetting process. Then η are stored and used for the interpolation of the variables among the overlapping grid blocks. Therefore, the determination of η not only addresses the problems of searching donor cell but also simplifies the interpolation of the variables among the overlapping grid blocks in the flow solver and the adjoint equation solver. More details and demonstrations of the IHC overset grid strategy can be found in Reference [9].

2.5. Numerical implementation for design optimization

In this work, the adjoint equations and the flow governing equations are coupled with each other and solved based on the conventional point-matched grids and overlapped grids within a multi-block

DOI: 10.1002/fld

framework. Parallel computation with message passing interface (MPI) is implemented to speed up the computation.

The performance of the optimum design strongly depends on how well the geometry representation method can approach the optimum shape. In Reference [27], Wu *et al.* compared three geometric representation methods for various transonic inverse design cases and suggested that the mesh-point method can reach higher accuracy although it may need more design cycles than the shape function method does to the optimum point. Therefore, in the present study, we choose the mesh-point method to adjust the shape of the body surface to be designed, in which the coordinates of all the mesh points defining the surfaces to be designed are considered as design variables. Hence, the numbers of design variables can be very large.

Generally, in the design process, the design variables are updated based on the gradient information. In the current study, the adjoint equation method is used to obtain the gradient information for the optimization design by combining with overset grid strategy to facilitate the treatment of complex aerodynamic configurations. The design procedure in one design cycle used in this work can be summarized as follows:

(1) Obtain the grid connectivity among the overlapping grid blocks using IHC method.
(2) Solve the flow governing equations (Equation (1)) to get the values for flow variables based on appropriate boundary conditions, including physical boundary, inner boundary and outer boundary.
(3) Solve the adjoint equations (Equation (A10) in Appendix) for ψ subject to appropriate boundary conditions similar to those for the flow governing equations.
(4) Perturb each design variable and calculate the resulting changes of the cost function δI. The gradient can be computed by evaluating the geometric variation integrals for each design variable based on the corresponding formula (Equation (A12) or Equation (A17)).
(5) Make the implicit smoothing for the gradient to maintain the smoothness of the designed surface.
(6) Introduce the gradient into a one-dimensional optimization search procedure.
(7) Update the shape based on the gradient information and then generate the new mesh by perturbing the old mesh.
(8) Repeat steps 1–7 until convergence is reached.

As shown in the above procedure, in each design cycle, the grid connectivity information among the overlapping grid blocks using the IHC method should be achieved first. Thereafter, the flow governing equations and adjoint equations can be solved subject to appropriate boundary conditions. As a consequence, the gradient is obtained in the following way. Small perturbations are imposed to an initial shape with each design variable being perturbed. Then the resulting variation of the cost function δI for each perturbed design variable can be calculated directly from the corresponding formula (see such as Equation (A12) or Equation (A17) in the Appendix), in which all the variations of flow variables have been eliminated and only the geometric variation integrals need to be evaluated. Therefore, the corresponding gradient component can be achieved without additional flow evaluation. Since the coordinates of all the mesh points defining the body surfaces to be designed are chosen as design variables, all of these surface points need to be perturbed by an appropriate amount point by point. In this work, the amount of perturbation is usually set to one thousandth of the characteristic length, for instance, the chord length of an airfoil. When such a mesh point on the body surface is perturbed, the mesh has to be adjusted once correspondingly. Generally, the new mesh can be reproduced by using a mesh generator or by perturbing the old

DOI: 10.1002/fld

mesh. In this work, an efficient trans-finite interpolation method [28] is adopted to perturb the old mesh by interpolating grid displacements, given at the boundary, to yield displacements of all grid points in the computational domain.

The gradient distribution on the designed surface is usually less smooth than the original shape. When the gradient information is achieved by evaluating the geometric variation integrals, a smoothing procedure has to be applied to the gradient in order to maintain the smoothness of the modified body surface. In this study, the Sobolev implicit smoothing approach [22] is applied. The smoothing technique effectively enhances the stability and efficiency of design process, which greatly reduces the number of required design cycles. Then the conjugate gradients [29] are calculated from the achieved gradient information and are used for optimization instead of the gradients themselves. In this study, for the first cycle in every five design cycles, the search direction is reset to the gradient direction in order to avoid error accumulation.

In each design cycle, after the gradient is determined, a one-dimensional search method [30] is applied to find an optimal step size along the gradient direction so as to accelerate the descent of the cost function. In this search method, with the initial design set to A, we can find two new designs B and C along the gradient direction with a chosen step size s and $2s$, respectively. The value of s is adjusted to make the cost function of B less than that of A and the cost function of C is larger than that of B. Then a parabolic fit can be used to find an approximate local minimum point D. The step size between A and D is just the optimal one that we are looking for, which is also used for the initial step size for the next design cycle. The details about this method can also be found in Reference [30].

Accordingly, the coordinates of the mesh points on the body surface being designed can be modified based on the gradient information achieved above. Then the mesh should be altered to accommodate the modified surfaces. With the use of the overset grid strategy in this study, only the blocks that contain the surfaces being designed are regenerated. The solutions of the previous flow evaluation and adjoint equation evaluation are used as the initial conditions for the next evaluation in order to reduce the number of iterations needed.

With the use of an overset grid system, the mesh perturbation should be treated with care when the changes of the cost function are computed as made in Step 4. The computation of the partial derivative $\partial R/\partial F$ can be obtained in a consistent way without being affected by the change of the connectivity due to the following reasons. First, the mesh points in most of the mesh blocks are not perturbed during the design process except the blocks that contain the surfaces that need to be redesigned. This greatly simplifies the computation of δI. Second, in the IHC method, all grid points including hole interior points and hole fringe points are treated indiscriminately in the flow computation. Therefore, all points except those inside the bodies of the other components can have valid values. In the present work, the blocks containing the surfaces to be designed are suggested not to overlap with the other bodies since the invalid values introduced by the IHC method for the points inside the bodies may contaminate the computational accuracy of δI.

To improve turn-around time and increase computational efficiency of the solvers on the hybrid multi-block grid system, parallel computing using MPI is used in solving both flow and adjoint equations. The level of parallelism used here is based on coarse grain data decomposition. The different blocks are automatically distributed by the load-balancing algorithms over a number of processors available on a parallel computer or networked commodity machines. Where the blocks are interfacing each other in a matched grid fashion, two layers of halo cells are used beyond the boundaries of each block to facilitate the implementation of boundary conditions and the

DOI: 10.1002/fld

communication between processors. For the overlapping grids, the fringe points obtained by the IHC methods are used for the communication. Connectivity information of the blocks and processors is stored in preprocessed pointer arrays. For each processor, the arrays store the information about the block numbers and cell numbers that have to be exchanged with another processor. Instead of exchanging information block by block, all values that need to be transferred to another node are stored in one array and then sent as one big message to minimize communication overhead. In order to further alleviate the load of the communication in parallel computing, the interpolated value is always computed on the processor where the donor cell lies and then sent to the corresponding receiver cell. This avoids sending all the stencil points from processors where the donor cells lie to other processors where the corresponding receiver points lie. The numerical tests in our previous study [9] have demonstrated that the current strategy for parallel computing works well in conjunction with the IHC method with good efficiency.

3. RESULTS AND DISCUSSIONS

The adjoint method has been used as an optimizer coupled with the flow solver in combination with the overset method. In this study, the flow governing equations considered are the Euler equations. As stated previously, the coordinates of every mesh point in the surfaces to be designed are considered as the design variables. Some two- and three-dimensional design cases are presented here to demonstrate the accuracy and efficiency of the current method. As the same code is used for both two- and three-dimensional test problems, even for the two-dimensional cases, a three-dimensional mesh is needed in the tests.

3.1. Inverse design for a desired pressure distribution

In order to check the accuracy of the current method based on the IHC method, a test was made for the redesign of an NACA0015 airfoil starting from the NACA0012 based on an overset grid system. The design objective is to reproduce a desired target pressure distribution, which was previously obtained from the computation of flow over an NACA0015 airfoil. The cost function for this problem is defined as given by Equation (A1). Here, the gradient information is calculated based on the coordinates on y direction for each vertex points. The overset grid system has been generated for a single airfoil as shown in Figure 1(a) (before hole cutting) and Figure 1(b) (after hole cutting). The mesh with totally around 5000 grid points is composed of 4 blocks, which are allowed to overlap each other and distributed over 3 CPUs for parallel computing.

Both subsonic and transonic flow conditions are considered. For the subsonic case, the Mach number is 0.4 and the angle of attack is 0°, whereas the Mach number case is set to 0.7 and the angle of attack is given as 2° for the transonic case. Figure 2 shows the comparison of the initial shape, the target shape and the designed shape for both cases. It can be seen that the designed shape matches the target shape very well in both subsonic and transonic flow conditions. Figure 3 shows the comparison of the corresponding pressure distribution on the airfoil surface. The distribution of Cp for the designed and target shape is almost identical. Figure 4 illustrates the history of the cost function in successive design cycles. It is apparent that the transonic case needs much more design cycles than the subsonic case to achieve the same drop in the cost function, which in this case is the difference in the computed pressure as compared with the target pressure.

DOI: 10.1002/fld

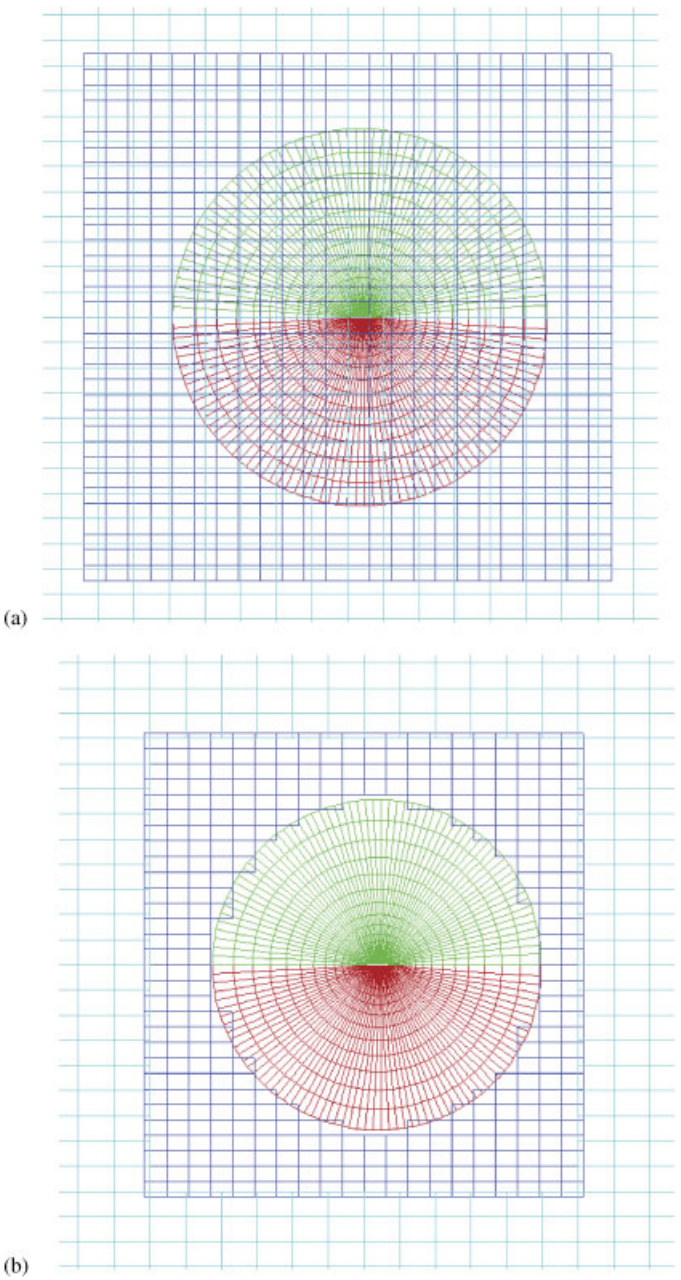

Figure 1. The overset grid for a NACA0012 airfoil: (a) before implicit hole cutting and (b) after implicit hole cutting.

DOI: 10.1002/fld

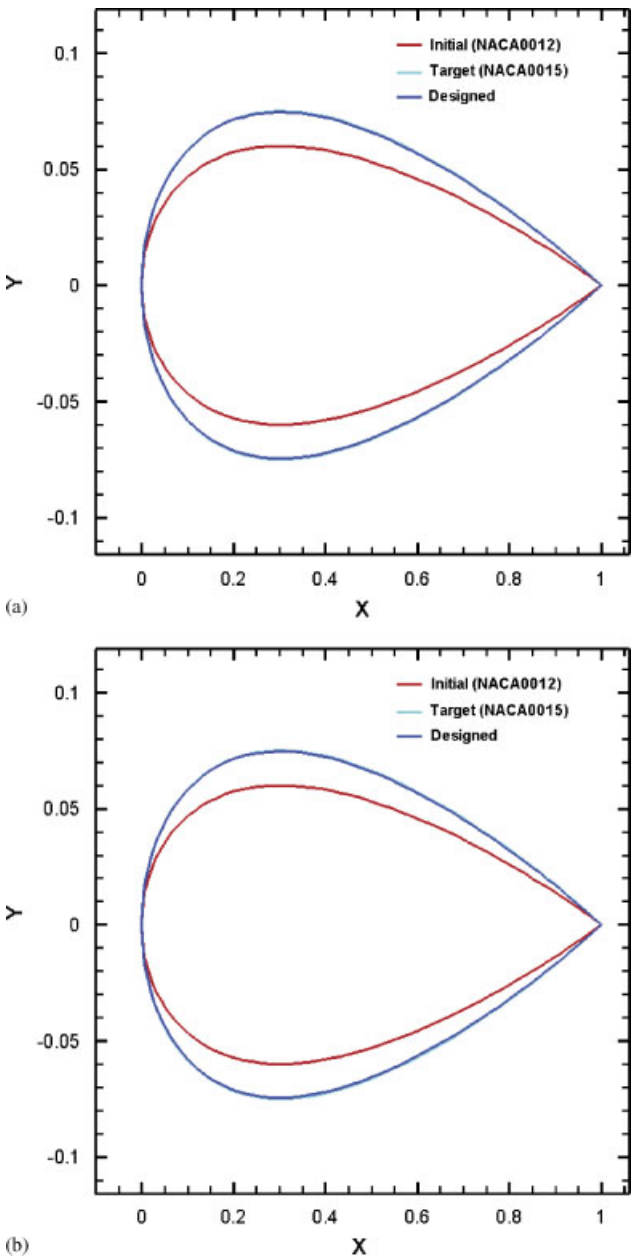

Figure 2. The designed shape, initial shape (NACA0012) and target shape (NACA0015) after the inverse design for two cases: (a) $Ma=0.4$, $\alpha=0$ and (b) $Ma=0.7$, $\alpha=2$.

DOI: 10.1002/fld

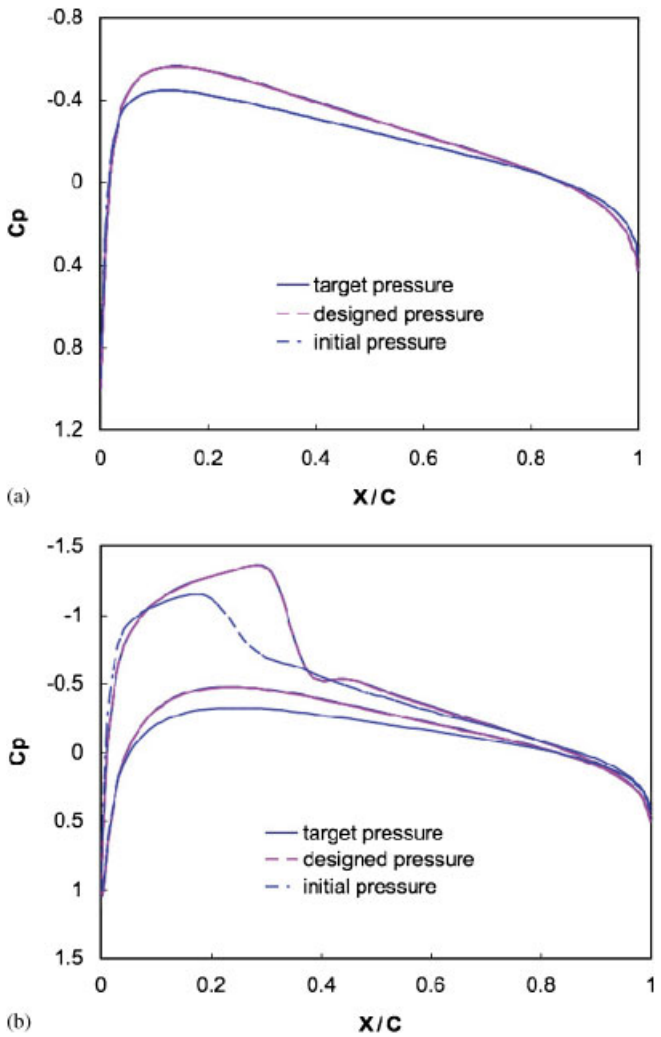

Figure 3. Comparison of the initial pressure distribution and designed pressure distribution for two cases: (a) $Ma=0.4$, $\alpha=0$ and (b) $Ma=0.7$, $\alpha=2$.

3.2. Constraint design for NACA0015 airfoil

In this case, the design optimization problem for the NACA0015 airfoil with a constraint condition is considered. The design objective is drag minimization without lift reduction. The computation is performed at a Mach number of 0.8 and an angle of attack of 1°. Under the transonic condition, the drag is the pressure drag of the airfoil mainly caused by the shock wave present. The objective function for this problem can be written as

$$I' = C_d - \Lambda \min(C_l - C_{l_0}, 0) \tag{11}$$

DOI: 10.1002/fld

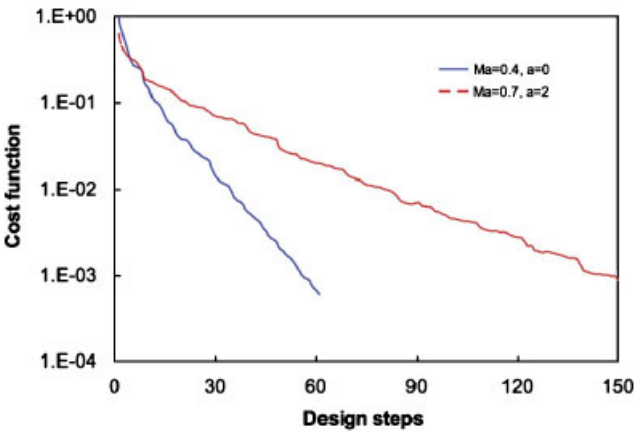

Figure 4. Convergence history of the cost functions for two design cases, i.e. Case 1: $Ma=0.4$, $\alpha=0$ and Case 2: $Ma=0.7$, $\alpha=2$.

where Λ is a Lagrange multiplier. Based on the theorems in optimization [31], the first-order method is used to solve the constrained minimization problem with an iterative process:

$$X^{k+1} = X^k - \lambda \nabla_X I' \qquad \Lambda^{k+1} = \Lambda^k + \varepsilon \nabla_\Lambda I' \tag{12}$$

Here, X are the design variables. If $C_l \geqslant C_{l_0}$

$$\frac{\partial I'}{\partial X} = \frac{\partial C_d}{\partial X}, \quad \frac{\partial I'}{\partial \Lambda} = 0 \tag{13}$$

If $C_l < C_{l_0}$

$$\frac{\partial I'}{\partial X} = \frac{\partial C_d}{\partial X} - \Lambda \frac{\partial C_l}{\partial X}, \quad \frac{\partial I'}{\partial \Lambda} = -(C_l - C_{l_0}) \tag{14}$$

For this case, we can use the same cost function as defined by Equation (A14). $I = C_d$ can be obtained by setting $\omega_1 = 1,\ \omega_2 = 0$ while $I = C_l$ can be got by setting $\omega_1 = 0,\ \omega_2 = 1$. Then $\partial C_d/\partial X$ and $\partial C_l/\partial X$ can be solved individually.

The overset grid system used here is similar to that in Case 1. Figure 5 shows the comparison of the initial shape and designed shape. Figure 6 indicates the Mach number contours for the initial solution and the designed shape. It can be seen that the strong shock in the initial solution is significantly weakened in the redesigned airfoil after 30 design cycles. Figure 7 gives the history of the cost function, drag efficient and lift efficient. The drag coefficient has dropped to approximately 20% of the initial value while the lift coefficient is not reduced.

3.3. Drag minimization for dual airfoils

The problem of testing two airfoils side by side brings out the issues of overset grids more poignantly. The problem is a drag minimization of one of the airfoil in the presence of the other under the transonic flow conditions at a Mach number of 0.755 and an angle of attack of

DOI: 10.1002/fld

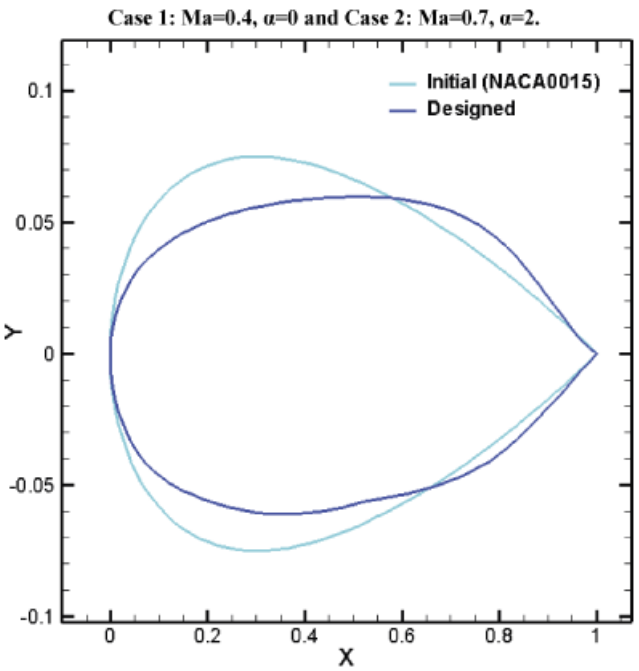

Figure 5. The initial shape and the designed shape after 30 design cycles of reducing drag without lift reduction for $Ma = 0.8$, $\alpha = 1$.

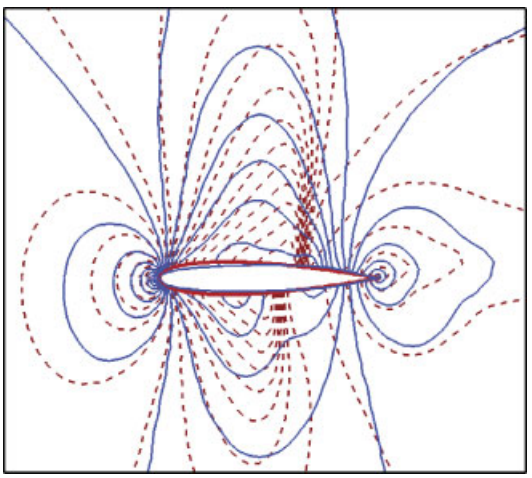

Figure 6. Comparison of the Mach number contours for $Ma = 0.8$, $\alpha = 1$ before design and after design. - - - -, before design; —, after design.

DOI: 10.1002/fld

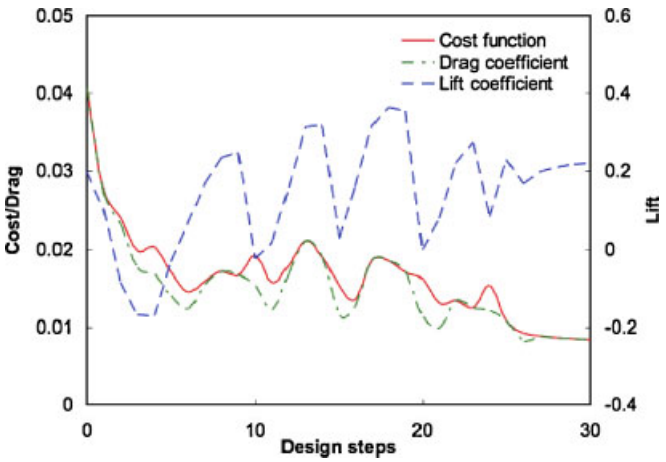

Figure 7. Convergence history of the cost function, drag and lift in the design of reducing drag without lift reduction for $Ma=0.8$, $\alpha=1$.

0.016°. The drag coefficient for the lower airfoil is chosen as the cost function here as given by Equation (A14) with $\omega_1=1$, $\omega_2=0$. This case is a challenging test since there is a strong shock across the overlapping grid between the two airfoils. The grid generation for this case is simple and straightforward due to the overset grid strategy, as two grids with each for one airfoil can be independently created and then assembled. The four different blocks of grid with more than 10 000 points are distributed over 3 CPUs for parallel computing.

Figure 8(a) and (b) shows the overlapping grid system for the dual NACA0012 airfoils before and after IHC, respectively. Figure 9 illustrates the Mach number contours for the initial solution and those for the designed shape. The convergence history of cost function for this case is presented in Figure 10. As shown in Figure 10, the drag coefficient for the lower airfoil is reduced by up to 44% of the initial values after five design cycles. It assumes a flatter shape on the upper surface at the forward side of the lower airfoil and the Mach contours show a weaker shock to the rear of the airfoil.

3.4. Inverse design for a turbine stator

This is a full three-dimensional case for an inverse design of a turbine stator. The use of overset grid is particularly pertinent for this class of problem as grid generation is a challenge for highly curved blades that would have to match with the side walls, inlet and outlet plane if a matched multi-block grid system is used. With an overset method, the grid over the surface could be created independent of the background grid. The design objective is to reproduce a prescribed target pressure distribution on the stator surface with the cost function defined by Equation (A1). The isentropic exit Mach number is 0.844, and the inflow angles are $-28°$. The configuration and the overset grid for the turbine stator are shown in Figures 11 and 12. The grid containing a total of 170 thousand points consists of 6 blocks, which are distributed over 6 CPUs for parallel computing. The difference between the initial shape and the target shape is only on the suction side. It is known that the shape on the pressure side has relatively small influences on the pressure

DOI: 10.1002/fld

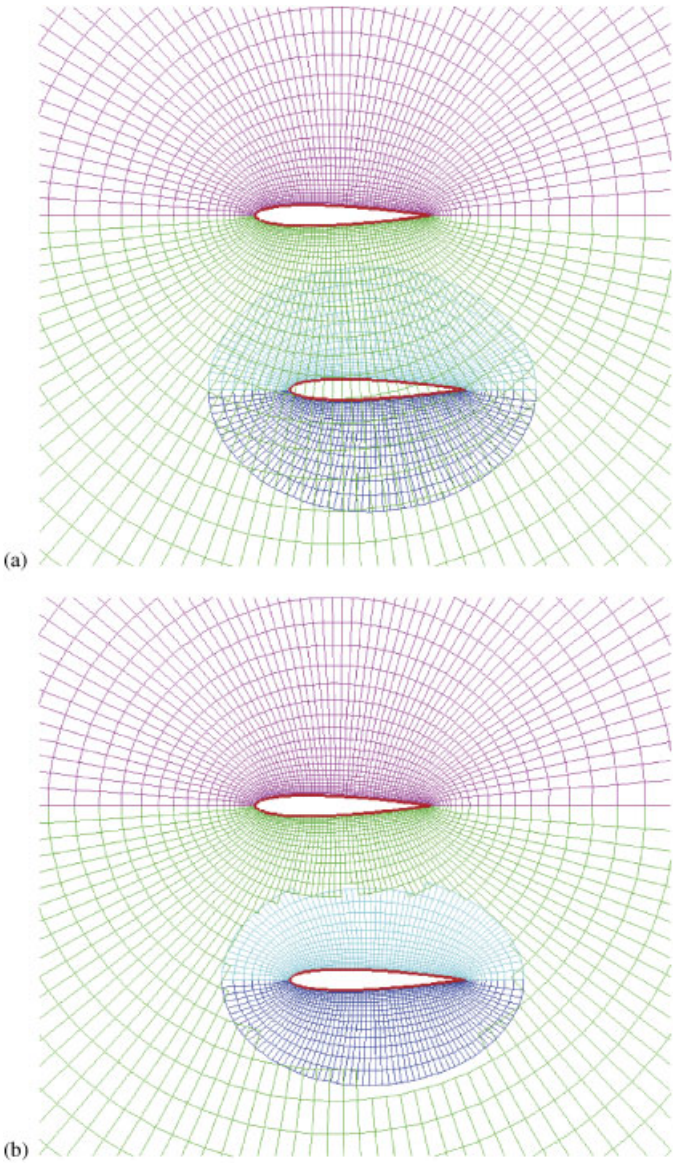

Figure 8. The overset grid for dual NACA0012 airfoils: (a) before implicit hole cutting and (b) after implicit hole cutting.

distributions. In this case, the gradient information is calculated for each vertex point in the axial direction.

Figure 13 shows the comparison for the initial shape, the designed shape and the target shape while Figure 14 shows the history of the cost function with successive designs. In this case, the cost function initially drops rapidly and then slowly after 10 design cycles. The value of the cost

DOI: 10.1002/fld

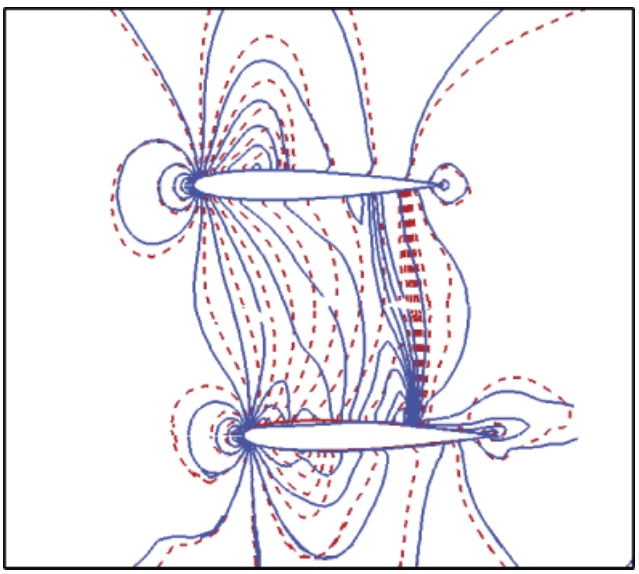

Figure 9. Comparison of the Mach number contours before design and after design for dual airfoils. - - - -, before design; —, after design.

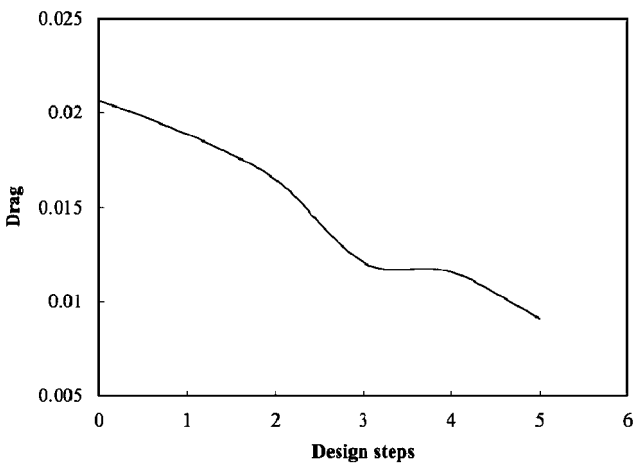

Figure 10. History of drag reduction for the second airfoil.

function reduces more than two orders in 20 cycles. From Figure 13, it can be seen that the designed shape is very close to the target shape and the differences observed between them are very minor. The current results are sufficiently encouraging to demonstrate the power of the overset method developed for the optimization of complex bodies constructed by composites of different simpler components.

DOI: 10.1002/fld

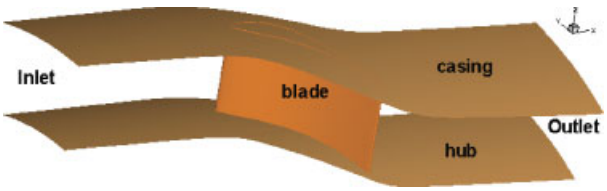

Figure 11. The configuration of a turbine stator.

Figure 12. The overset grid for a turbine stator: (a) before implicit hole cutting and (b) after implicit hole cutting.

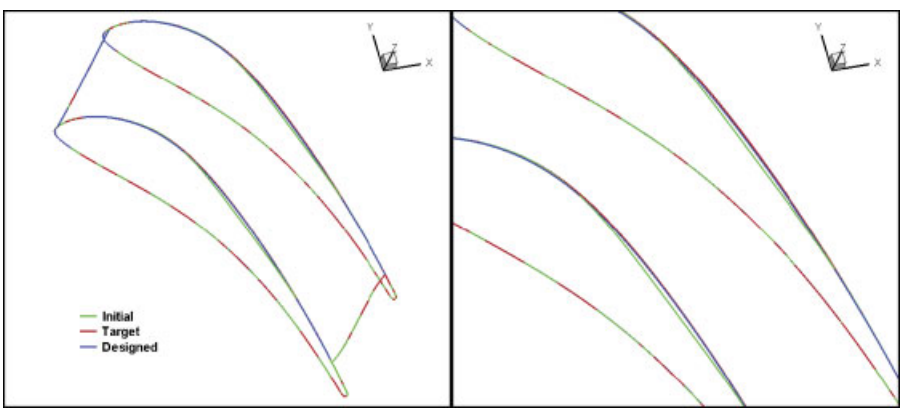

Figure 13. Comparison of the designed shape, the initial shape and the target shape for the inverse design of a turbine stator.

DOI: 10.1002/fld

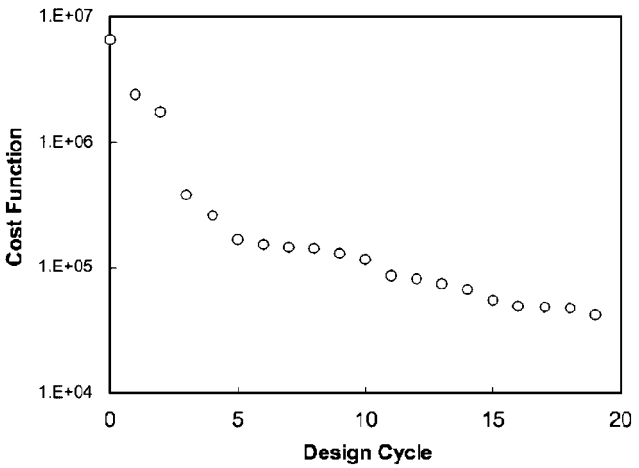

Figure 14. Convergence history of the cost function in the inverse design for a turbine stator.

4. CONCLUDING REMARKS

An aerodynamic shape optimization method that involves the continuous adjoint equation and the flow governing equations being solved using overlapped grids is examined in this paper. The use of overset grids greatly eases the design and optimization of complex aerodynamic configurations in engineering. The present overset approach makes use of the implicit hole cutting (IHC) concept [8, 9] within a multi-block grid framework. Parallel computation with MPI was also implemented to speed up the computation.

The present scheme was tested for several two- and three-dimensional shape optimization cases for external and internal flows, including inverse design to a desired pressure distribution and drag minimization with lift constrain. Two-dimensional test cases were made to test and demonstrate the capabilities of the three-dimensional code that was developed. The test cases show that the overall adjoint method works well within an overset grid system. The simplicity of the implicit hole cutting technique previously shown [9] is further demonstrated here for the adjoint equation. As highlighted above unlike other overset methods, the IHC method greatly eases both the organization and programming of the adjoint solver on the overlapped meshes. The method is purely a cell selection process based on the main criterion of cell size, and all grid points including hole interior points and hole fringe points are treated indiscriminately in the computation. This approach greatly facilitates the use of the adjoint method for engineering shape optimizations. It is particularly pertinent for complex shapes that are created via a composite of simpler shapes in an overset grid manner.

With overset grids, the question of whether it is absolutely necessary to solve the adjoint equations exactly like the flow equations for the entire domain till the far field is not considered here in this study. It may well be possible to solve the adjoint equations around the component one is seeking to optimize using only the overset grids around that particular component. This issue, of how the boundary conditions for the adjoint solver are to be implemented, is left as a subject for future investigation.

DOI: 10.1002/fld

APPENDIX A: DERIVATION OF ADJOINT EQUATION FOR THE DESIGN PROBLEMS

For an inverse design case in which the objective is a given pressure distribution on an airfoil in an inviscid flow, the cost function can be written as

$$I=\frac{1}{2}\int_{B_W}(p-p_d)^2|S_2|\,\mathrm{d}\xi_1\,\mathrm{d}\xi_3 \tag{A1}$$

where ξ_i with $i=1$ to 3 is introduced to describe the three-dimensional computational domain so that each boundary conforms to a constant value of one of these coordinates. S_i denotes the projection area of the cell face in the ξ_i direction. p_d is the desired pressure and B_W stands for the boundary surface in computational domain corresponding to the airfoil surface in the physical domain. For convenience, here $\xi_2=0$ is considered as the airfoil surface to be designed.

Only steady-state conditions are considered in the current study. The Euler equations in the computational domain for the steady state can be written as

$$\frac{\partial F_i}{\partial \xi_i}=0$$

where $F_i=S_{ij}f_j$ and f_j is the inviscid flux. Correspondingly, a weak form of the Euler equations in the computational domain follows:

$$\int_\Omega \frac{\partial \varphi^{\mathrm{T}}}{\partial \xi_i}F_i\,\mathrm{d}\Omega=\int_{\partial\Omega} n_i\varphi^{\mathrm{T}}F_i\,\mathrm{d}B \tag{A2}$$

where φ is an arbitrary differentiable function. Then the variation for the weak form of the Euler equation can be given as

$$\int_\Omega \frac{\partial \varphi^{\mathrm{T}}}{\partial \xi_i}\delta F_i\,\mathrm{d}\Omega=\int_{\partial\Omega} n_i\varphi^{\mathrm{T}}\delta F_i\,\mathrm{d}B \tag{A3}$$

where

$$\delta F_i=\delta(S_{ij}f_j)=S_{ij}\delta f_j+\delta(S_{ij})f_j=S_{ij}A_j\delta w+\delta(S_{ij})f_j=C_i\delta w+\delta(S_{ij})f_j \tag{A4}$$

$$C_i=S_{ij}A_j,\quad A_j=\frac{\partial f_j}{\partial w} \tag{A5}$$

The integrations in Equations (A2) and (A3) are performed in the computational domain, Ω. $\partial\Omega$ is the boundary of Ω. Since Equation (A3) should hold for an arbitrary choice of the test vector φ, we are free to choose φ to simplify the resulting expressions. Therefore, we can set $\varphi=\psi$, the Lagrange multiplier. Adding Equation (A3) to Equation (A1) and taking its variation, we obtain

$$\begin{aligned}\delta I=&\int_{B_W}(p-p_d)|S_2|\delta p\,\mathrm{d}\xi_1\,\mathrm{d}\xi_3+\frac{1}{2}\int_{B_W}(p-p_d)^2\delta(|S_2|)\,\mathrm{d}\xi_1\,\mathrm{d}\xi_3\\&-\int_\Omega\frac{\partial\psi^{\mathrm{T}}}{\partial\xi_i}\delta F_i\,\mathrm{d}\Omega+\int_{\partial\Omega}n_i\psi^{\mathrm{T}}\delta F_i\,\mathrm{d}B\end{aligned} \tag{A6}$$

DOI: 10.1002/fld

Substitute δF_i of Equation (A4) into Equation (A6)

$$\delta I = \int_{B_W} (p - p_d)|S_2|\delta p \, \mathrm{d}\xi_1 \, \mathrm{d}\xi_3 + \frac{1}{2}\int_{B_W} (p - p_d)^2 \delta(|S_2|) \, \mathrm{d}\xi_1 \, \mathrm{d}\xi_3$$
$$- \int_\Omega \left(\frac{\partial \psi^{\mathrm{T}}}{\partial \xi_i} C_i \delta w + \frac{\partial \psi^{\mathrm{T}}}{\partial \xi_i} \delta S_{ij} f_j \right) \mathrm{d}\Omega + \int_{\partial\Omega} n_i \psi^{\mathrm{T}} \delta F_i \, \mathrm{d}B \tag{A7}$$

In order to eliminate δw explicitly in δI, let

$$\frac{\partial \psi^{\mathrm{T}}}{\partial \xi_i} C_i = 0 \tag{A8}$$

which can also be written as

$$C_i^{\mathrm{T}} \frac{\partial \psi}{\partial \xi_i} = 0 \tag{A9}$$

To make Equation (A9) time dependent so that a time-marching scheme similar to the flow solver can be used, we set

$$\frac{\partial \psi}{\partial t} - C_i^{\mathrm{T}} \frac{\partial \psi}{\partial \xi_i} = 0 \tag{A10}$$

This is the so-called adjoint equation.

Then δI becomes

$$\delta I = \int_{B_W} (p - p_d)|S_2|\delta p \, \mathrm{d}\xi_1 \, \mathrm{d}\xi_3 + \frac{1}{2}\int_{B_W} (p - p_d)^2 \delta(|S_2|) \, \mathrm{d}\xi_1 \, \mathrm{d}\xi_3$$
$$- \int_\Omega \frac{\partial \psi^{\mathrm{T}}}{\partial \xi_i} \delta S_{ij} f_j \, \mathrm{d}\Omega + \int_{\partial\Omega} n_i \psi^{\mathrm{T}} \delta F_i \, \mathrm{d}B \tag{A11}$$

Next consider the boundary integral of flow variations in Equation (A11). Following the similar procedure, we collect the coefficient of each independent flow variation and set the coefficients to zero. Then the boundary conditions for ψ can be obtained. In this manner, we eliminate the explicit dependence of δI on flow variations on the boundaries. Then we can obtain the final form of δI

$$\delta I = \frac{1}{2}\int_{B_W} (p - p_d)^2 \delta(|S_2|) \, \mathrm{d}\xi_1 \, \mathrm{d}\xi_3 - \int_\Omega \frac{\partial \psi^{\mathrm{T}}}{\partial \xi_i} \delta S_{ij} f_j \, \mathrm{d}\Omega$$
$$- \int_{B_W} (\psi_2 \delta S_{21} + \psi_3 \delta S_{22} + \psi_4 \delta S_{23}) p \, \mathrm{d}\xi_1 \, \mathrm{d}\xi_3 \tag{A12}$$

with the boundary conditions on B_w for the adjoint equation (Equation (A10))

$$\psi_j n_j = p - p_d \tag{A13}$$

DOI: 10.1002/fld

where n_j are the components of the surface normal

$$n_j = \frac{S_{2j}}{|S_2|}$$

As we expect, all flow variations have disappeared and the change of the cost function in Equation (A12) only depends on the metric variations δS.

If the cost function is different, we can derive the corresponding formula following the same procedure as above. When the function to derive is a boundary integral, the only changes for the derived formula are the boundary conditions for the adjoint equations and Equation (A12) for the computation of δI. The adjoint equation maintains the same form. Considering a case in which the design objective is to decrease drag coefficient with or without the constraints of lift coefficient, we can let the cost function be a linear combination of C_d and C_l. For generality, we set

$$I = \omega_1 C_d + \omega_2 C_l \tag{A14}$$

where

$$\begin{aligned} C_d &= C_A \cos\alpha + C_N \sin\alpha = \frac{1}{S_{\text{ref}}} \int_{B_W} C_p (S_{21} \cos\alpha + S_{22} \sin\alpha)\, \mathrm{d}\xi_1\, \mathrm{d}\xi_3 \\ C_l &= C_N \cos\alpha - C_A \sin\alpha = \frac{1}{S_{\text{ref}}} \int_{B_W} C_p (S_{22} \cos\alpha - S_{21} \sin\alpha)\, \mathrm{d}\xi_1\, \mathrm{d}\xi_3 \end{aligned} \tag{A15}$$

ω_{12} and ω_2 are the weights. From Equation (A14), we can get $I = C_d$ with $\omega_1 = 1,\ \omega_2 = 0$ while we can have $I = C_l$ by setting $\omega_1 = 0,\ \omega_2 = 1$.

Then we have

$$I = \omega_1 C_d + \omega_2 C_l = \frac{1}{S_{\text{ref}}} \int_{B_W} C_p (\beta_1 S_{21} + \beta_2 S_{22})\, \mathrm{d}\xi_1\, \mathrm{d}\xi_3 \tag{A16}$$

with

$$\beta_1 = \omega_1 \cos\alpha - \omega_2 \sin\alpha$$

$$\beta_2 = \omega_1 \sin\alpha + \omega_2 \cos a$$

Following a similar derivation procedure described earlier for the inverse design problem to obtain a desired pressure distribution, we have the formula for the change of cost function for every design variable

$$\begin{aligned} \delta I = {} & \frac{1}{S_{\text{ref}}} \int_{B_W} C_p (\beta_1 \delta S_{21} + \beta_2 \delta S_{22})\, \mathrm{d}\xi_1\, \mathrm{d}\xi_3 - \int_{\Omega} \frac{\partial \psi^{\mathrm{T}}}{\partial \xi_i} \delta S_{ij} f_j\, \mathrm{d}\Omega \\ & - \int_{B_W} (\psi_2 \delta S_{21} + \psi_3 \delta S_{22} + \psi_4 \delta S_{23}) p\, \mathrm{d}\xi_1\, \mathrm{d}\xi_3 \end{aligned} \tag{A17}$$

with the boundary conditions on B_w for the adjoint equations

$$\psi_j S_{2j} = \frac{2}{S_{\text{ref}} \rho_\infty U_\infty} (\beta_1 S_{21} + \beta_2 S_{22}) \tag{A18}$$

DOI: 10.1002/fld

Once the solutions of the flow equations and the adjoint equations are obtained, the gradient information can be achieved by evaluating the geometric variation integrals as given in Equations (A12) or (A17).

ACKNOWLEDGEMENTS

We are very grateful to Prof. Feng Liu, Dr Hsiao-Yuan Wu, Dr Mani Sadeghi and Dr Shuchi Yang of University of California, Irvine for their unreserved help in code development and many valuable suggestions and fruitful discussions. We are also grateful to Dr Zhengke Zhang (formally of Temasek Laboratories) for discussion on constraint design problems.

REFERENCES

1. Hicks RM, Henne PA. Wing design by numerical optimization. *Journal of Aircraft* 1978; **15**:407–412.
2. Reuther J, Cliff SE, Hicks RM, Van Dam CP. Practical design optimization of wing/body configurations using the Euler equations. *AIAA Paper 1992-2633*, 1992.
3. Jameson A. Optimum aerodynamic design using CFD and control theory. *AIAA Paper 1995-1729, AIAA 12th Computational Fluid Dynamics Conference*, San Diego, CA, June 1995.
4. Jameson A. Aerodynamic design via control theory. *Journal of Scientific Computing* 1988; **3**:233–260.
5. Quagliarella D, Cioppa AD. Genetic algorithms applied to the aerodynamic design of transonic airfoils. *AIAA Paper 1994-1896*, 1994.
6. Yamamoto K, Inoue O. Applications of genetic algorithm to aerodynamic shape optimization. *AIAA Paper 1995-1650*, 1995.
7. Kampolis IC, Papadimitriou DI, Giannakoglou KC. Evolutionary optimization using a new radial basis function network and the adjoint formulation. *Inverse Problems in Science and Engineering* 2006; **14**(4):397–410.
8. Lee YL, Baeder JD. Implicit hole cutting—a new approach to overset grid connectivity. *AIAA Paper 2003-4128, AIAA 16th Computational Fluid Dynamics Conference*, Orlando, FL, 2003.
9. Liao W, Cai J, Tsai HM. A multigrid overset grid flow solver with implicit hole cutting method. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:1701–1715.
10. Rogers SE, Suhs NE, Dietz WE. PEGASUS 5: an automated preprocessor for overset-grid computational fluid dynamics. *AIAA Journal* 2003; **41**(6):1037–1045.
11. Brown DL, Henshaw WD, Quinlan DJ. Overture: object-oriented tools for overset grid applications. *AIAA Paper 1999-3130*, 1999.
12. Wang ZJ, Parthasarathy V. A fully automated chimera methodology for multiple moving body problems. *International Journal for Numerical Methods in Fluids* 2000; **33**(7):919–938.
13. Anderson WK, Venkatakrishnan V. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Computers and Fluids* 1999; **28**:443–480.
14. Jameson A, Shankaran S, Martinelli L. An unstructured adjoint method for transonic flows. *AIAA Paper 2003-3955*, 2003.
15. Dadone A, Grossman B. Ghost-cell method for inviscid two-dimensional flows on Cartesian grids. *AIAA Journal* 2004; **42**(12):2499–2507.
16. Marian N, Michael JA. Adjoint formulation for an embedded-boundary Cartesian method. *NAS Technical Report, NAS-05-008*, 2005.
17. Elliott J. Discrete adjoint analysis and optimization with overset grid modeling of the compressible, high-re Navier–Stokes equations. *Proceedings of the 6th Overset Grids and Solution Technology Symposium*, Fort Walton Beach, FL, 2002.
18. Lee BJ, Kim C. Aerodynamic shape optimization using discrete adjoint formulation based on overset mesh technique. *Proceedings of European Conference on Computational Fluid Dynamics*, Egmond aan Zee, The Netherlands, 2006. Available from: http://proceedings.fyper.com/eccomascfd2006/.
19. Lei Z, Makino Y. CFD-based aerodynamic optimization for complex aircraft configuration. *CFD Journal* 2004; **13**(1):304–310.
20. Kim S, Alonso JJ, Jameson A. A gradient accuracy study for the adjoint-based Navier–Stokes design method. *AIAA Paper 1999-0299, AIAA 37th Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 1999.

DOI: 10.1002/fld

21. Nadarajah S, Jameson A. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. *AIAA Paper 2000-0667, AIAA 38th Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2000.
22. Jameson A. *Aerodynamic Shape Optimization using the Adjoint Method*. Lecture Series. Von Karman Institute for Fluid Dynamics: Brussels, Belgium, February 2003.
23. Jameson A, Schmidt W, Turkel E. Numerical solutions of the euler equations by finite volume methods using Runge–Kutta time-stepping schemes. *AIAA Paper 1981-1259*, 1981.
24. Liu F, Cai J, Zhu Y, Tsai HM, Wong ASF. Calculation of wing flutter by a coupled fluid–structure method. *Journal of Aircraft* 2001; **38**:334–342.
25. Jameson A, Baker T. Solution of the Euler equations for complex configurations. *AIAA Paper 1983-1929*, 1983.
26. Carlson RE, Fritsch FN. An algorithm for monotone piecewise bicubic interpolation. *SIAM Journal on Numerical Analysis* 1989; **26**:230–238.
27. Wu HY, Yang S, Liu F, Tsai HM. Comparison of three geometric representations of airfoils for aerodynamic optimization. *AIAA Paper 2003-4095*, 2003.
28. Tsai HM, Wong ASW, Cai J, Zhu Y, Liu F. Unsteady flow calculation with a parallel multiblock moving mesh algorithm. *AIAA Journal* 2001; **39**(6):1021–1029.
29. Fletcher R, Reeves CM. Function minimization by conjugate gradients. *The Computer Journal* 1964; **7**(2):149–154.
30. Yang S, Wu HY, Liu F, Tsai HM. Aerodynamic design of cascades by using an adjoint equation method. *AIAA Paper 2003-1068*, 2003.
31. Benjamin WW, Wu Z. The theory of discrete Lagrange multipliers for nonlinear discrete optimization. *Principles and Practice of Constraint Programming*. Springer: Berlin, 1999; 28–42.

DOI: 10.1002/fld