AIAA JOURNAL
Vol. 41, No. 12, December 2003

Characteristic Interface Conditions for Multiblock High-Order Computation on Singular Structured Grid

Jae Wook Kim* and Duck Joo Lee†
Korea Advanced Institute of Science and Technology, Daejeon 305-701, Republic of Korea

A structured grid with an embedded body usually has a certain point where an abrupt change in the slope of the grid line exists. The grid metrics are discontinuous at such a point because of the discrepancy between the left- and the right-hand limits of the gradients. This leads to grid singularity. It may cause serious numerical oscillations especially when high-order finite difference schemes are applied to solving conservation-form governing equations in generalized coordinates. This problem is handled by decomposing the computational domain into blocks along the singular lines and imposing interface conditions at the block interfaces for communication between the blocks. A set of high-order finite difference schemes is used in each block: central differences on the interior nodes and one-sided differences on the near-interface nodes. The differencing stencils do not cross the block interfaces, and each block is isolated without the singularity, which results in no oscillations. For communication between the isolated blocks, the interface conditions are derived from the characteristic relations of the compressible Euler or Navier–Stokes equations. The exactness and the feasibility of the interface conditions are investigated for the high-order multiblock computation on structured grids containing singular points.

I. Introduction

STRUCTURED grids have been widely used for computational fluid dynamics (CFD) applying a variety of finite difference schemes (FDSs). The quality of grid generation has a great effect on the accuracy of numerical solutions. Many algorithms for the generation of structured grids are reviewed and summarized by Fletcher,[1] Hoffmann and Chiang,[2] and Anderson et al.[3] Such algorithms are designed to provide sufficiently fine grid systems for a general computational domain surrounded by arbitrary boundaries with complex bodies. However, most grids have a singularity problem even when the body configuration is relatively simple. The grid singularity exists where an abrupt change in the slope of a grid line appears. The grid metrics are discontinuous at the singular point because of the discrepancy between the left- and the right-hand limits of the gradients. This means that the grid metrics are not unique at the point. The discontinuity of the grid metrics is an issue to be discussed both mathematically and numerically. Some attempts to use a whole Cartesian grid to avoid this kind of problem were proposed by Kurbatskii and Tam,[4] and Dyson and Goodrich.[5] These new approaches wait to be tested and validated through further applications.

For practical CFD with FDSs, the partial differential equations (PDEs) are solved in the transformed domains with generalized coordinates. In most cases, the conservative forms of the PDEs are used, and all of the grid metrics and the transformation Jacobian are contained in the differential operator with the flow variables. The grid singularity of the type mentioned then leads to a discontinuous solution. The discontinuous solution clearly does not satisfy the PDE in a mathematical sense at all points because the derivatives are not defined at the discontinuity. Moreover, such a mathematical difficulty increases when the flowfield has discontinuous properties such as shock waves or contact surfaces. The flow singularity has been well treated by a number of approaches during the last three decades as reported by LeVeque.[6] However, little work has been contributed for the grid singularity. Treatments for the grid singularity are described in the present paper.

In conventional CFD, to simplify the calculation, the grid metrics at the singular points are often approximated to single values by averaging the left- and the right-hand limits. This certainly incurs numerical errors from a geometric point of view. In addition, the grid singularity increases the complexity, especially when it comes to discretization, that is, numerical differentiation. It acts like a standing shock in the transformed flux variables, and spurious oscillations can be generated as the numerical differentiation is carried out near the singular points. Well-established CFD solvers have produced accurate solutions with stable low-order FDSs and artificial dissipation that can suppress the errors. However, high-order FDSs devoted to high resolution in unsteady aerodynamics and aeroacoustics can easily yield spurious results because of their high sensitivity to the grid quality. It becomes more serious when the high-order FDS is based on the central differences rather than the upwind differences. To this time, applications of the high-order central FDSs have been limited to quite simple structured grids without singular points.[4,5,7–11]

In the present work, the grid singularity is avoided by a multiblock computing technique with suitable interface conditions. The computational domain is decomposed into blocks along the singular grid lines, where the left and the right blocks have the left- and right-hand limits of the grid metrics, respectively. High-order FDSs are used in each block, which employ central differences on the interior nodes and one-sided differences on the near-interface nodes. The differencing stencils do not cross the block interfaces. As a consequence, each block is isolated from the grid singularity, and no spurious oscillations are generated in the numerical differentiation. The next step is imposing interface conditions at the block interfaces for correct physical communication between the isolated blocks. The interface conditions are derived from the characteristic relations of the compressible Euler or Navier–Stokes equations. The characteristic waves calculated in one block are interchanged or replaced by those in the next block at the interface according to their directions.

The organization of the present paper is as follows. In Sec. II, the governing equations are described transformed from Cartesian to generalized coordinates. In Sec. III, the numerical algorithms used are introduced, especially the high-order FDSs for evaluating the space derivatives. In Sec. IV, the detailed procedure is presented

Received 24 February 2003; presented as Paper 2003-3122 at the AIAA/CEAS 9th Aeroacoustics Conference, Hilton Head, SC, 12–14 May 2003; revision received 23 July 2003; accepted for publication 6 August 2003.

*Research Professor, Department of Aerospace Engineering, 373-1 Guseong, Yuseong; jwk@kaist.ac.kr. Member AIAA.

†Professor, Department of Aerospace Engineering, 373-1 Guseong, Yuseong; djlee@mail.kaist.ac.kr. Member AIAA.

for avoiding the grid singularity by decomposing the computational domain into blocks. In Sec. V, the derivation of the block interface conditions is presented based on the characteristic relations. In Sec. VI, the interface conditions are applied to some sample computations. Finally, concluding remarks are made in Sec. VII.

II. Governing Equations

The governing equations are the unsteady compressible Euler or Navier–Stokes equations. The flux vector form of the governing equations, transformed to the computational domain, may be expressed in generalized coordinates as

$$\frac{\partial \hat{\boldsymbol{Q}}}{\partial t} + \frac{\partial \hat{\boldsymbol{E}}}{\partial \xi} + \frac{\partial \hat{\boldsymbol{F}}}{\partial \eta} + \frac{\partial \hat{\boldsymbol{G}}}{\partial \zeta} = \hat{\boldsymbol{S}}_V \tag{1}$$

where the caret indicates the transformed properties. The vectors of the conservative variables and the Euler fluxes in the generalized coordinates can be represented as

$$\hat{\boldsymbol{Q}} = \boldsymbol{Q}/J, \qquad \hat{\boldsymbol{E}} = (\xi_x \boldsymbol{E} + \xi_y \boldsymbol{F} + \xi_z \boldsymbol{G})/J$$
$$\hat{\boldsymbol{F}} = (\eta_x \boldsymbol{E} + \eta_y \boldsymbol{F} + \eta_z \boldsymbol{G})/J, \qquad \hat{\boldsymbol{G}} = (\zeta_x \boldsymbol{E} + \zeta_y \boldsymbol{F} + \zeta_z \boldsymbol{G})/J \tag{2}$$

and $\hat{\boldsymbol{S}}_V$ is a source term that consists of the viscous flux derivatives for the Navier–Stokes equations, but is zero for the Euler equations. All of the components of $\hat{\boldsymbol{S}}_V$ are fully described in Refs. 1–3. The vectors of the conservative variables and the Euler fluxes in Cartesian coordinates are given by

$$\boldsymbol{Q} = (\rho, \rho u, \rho v, \rho w, \rho e_t)^T$$
$$\boldsymbol{E} = \left(\rho u, \rho u^2 + p, \rho vu, \rho wu, (\rho e_t + p)u\right)^T$$
$$\boldsymbol{F} = \left(\rho v, \rho uv, \rho v^2 + p, \rho wv, (\rho e_t + p)v\right)^T$$
$$\boldsymbol{G} = \left(\rho w, \rho uw, \rho vw, \rho w^2 + p, (\rho e_t + p)w\right)^T$$

where the total energy per unit mass is defined as $e_t = p/[(\gamma - 1)\rho] + (u^2 + v^2 + w^2)/2$ and $\gamma = c_p/c_v$ is the ratio of specific heats. For air, $\gamma = 1.4$, and this value is used in the present computation. In Eq. (2), the transformation Jacobian J and the grid metrics $\xi_x, \ldots, \zeta_z$ are given by

$$J = 1/[x_\xi(y_\eta z_\zeta - y_\zeta z_\eta) + x_\eta(y_\zeta z_\xi - y_\xi z_\zeta) + x_\zeta(y_\xi z_\eta - y_\eta z_\xi)]$$

with

$$\begin{pmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{pmatrix} = J \begin{pmatrix} y_\eta z_\zeta - y_\zeta z_\eta & z_\eta x_\zeta - z_\zeta x_\eta & x_\eta y_\zeta - x_\zeta y_\eta \\ y_\zeta z_\xi - y_\xi z_\zeta & z_\zeta x_\xi - z_\xi x_\zeta & x_\zeta y_\xi - x_\xi y_\zeta \\ y_\xi z_\eta - y_\eta z_\xi & z_\xi x_\eta - z_\eta x_\xi & x_\xi y_\eta - x_\eta y_\xi \end{pmatrix}$$

III. Numerical Algorithms

High-order FDSs are used to evaluate the flux derivatives in the present computation on a structured grid. The main scheme is a pentadiagonal type of central compact FDS.[12] It is a generalization of the seven-point stencil Padé scheme and is used on the interior nodes. It can be expressed as

$$\beta f'_{i-2} + \alpha f'_{i-1} + f'_i + \alpha f'_{i+1} + \beta f'_{i+2} = \frac{1}{h}\sum_{m=1}^{3} a_m (f_{i+m} - f_{i-m}) \tag{3}$$

where f_i is an objective function for the flux variables and f'_i is its space derivative at the ith node. The grid spacing h is a constant independent of the index i in the computational domain where all of the grid points are equally spaced. Equation (3) can be solved by inverting a pentadiagonal matrix. The matrix must be completed at the boundaries. Therefore, noncentral or one-sided formulations other than Eq. (3) are needed on the boundary and the near-boundary nodes.[13] These may be represented as

$$f'_0 + \alpha_{0,1} f'_1 + \beta_{0,2} f'_2 = \frac{1}{h}\sum_{\substack{m=0 \\ m\neq 0}}^{3} a_{0,m}(f_m - f_0)$$

for $i = 0$ (boundary node) (4)

$$\alpha_{1,0} f'_0 + f'_1 + \alpha_{1,2} f'_2 + \beta_{1,3} f'_3 = \frac{1}{h}\sum_{\substack{m=0 \\ m\neq 1}}^{4} a_{1,m}(f_m - f_1)$$

for $i = 1$ (5)

$$\beta_{2,0} f'_0 + \alpha_{2,1} f'_1 + f'_2 + \alpha_{2,3} f'_3 + \beta_{2,4} f'_4 = \frac{1}{h}\sum_{\substack{m=0 \\ m\neq 2}}^{5} a_{2,m}(f_m - f_2)$$

for $i = 2$ (6)

The coefficients in Eqs. (3–6) are listed in Table 1. They are optimized, as described in Refs. 12 and 13, to achieve maximum resolution characteristics with fourth-order accuracy [second order in Eq. (4) for numerical stability]. A fourth-order, four-stage Runge–Kutta scheme is used for marching the solutions in time.

High-order FDSs in space and time resolve a wider range of wave number or frequency than low-order schemes. However, even the present schemes do not resolve the highest wave number or frequency range effectively. An adaptive nonlinear artificial dissipation model[14] is used to remove the unwanted numerical oscillations that may develop from the unresolved range. The artificial dissipation model is implemented only at the last (fourth) stage of the Runge–Kutta scheme to minimize computational costs. In addition to the stringent requirements on the high-order numerical schemes, an accurate and robust calculation depends heavily on the suppression of any waves that may result from unwanted reflections on

Table 1 Optimized coefficients for compact FDSs

Equation (3)	Equation (4)	Equation (5)	Equation (6)
$a_1 = 0.6511278808920836$	$a_{0,1} = -3.061503488555582$	$a_{1,0} = -0.5401943305881343$	$a_{2,0} = -0.1327404414078232$
$a_2 = 0.2487500014377899$	$a_{0,2} = 5.917946021057852$	$a_{1,2} = 0.8952361063034303$	$a_{2,1} = -0.6819452549637237$
$a_3 = 0.00614479661269978$	$a_{0,3} = 0.4176795271056629$	$a_{1,3} = 0.2553815577627246$	$a_{2,3} = 0.7109139355526556$
$\alpha = 0.5775233202590945$	$\alpha_{0,1} = 5.870156099940824$	$a_{1,4} = 0.00754902939458253$	$a_{2,4} = 0.2459462758541114$
$\beta = 0.08953895334666784$	$\beta_{0,2} = 3.157271034936285$	$\alpha_{1,0} = 0.1663921564068434$	$a_{2,5} = 0.003965415751510620$
		$\alpha_{1,2} = 0.7162501763222718$	$\beta_{2,0} = 0.03447751898726934$
		$\beta_{1,3} = 0.08619830787164529$	$\alpha_{2,1} = 0.4406854601950040$
			$\alpha_{2,3} = 0.6055509079866320$
			$\beta_{2,4} = 0.08141498512587530$

computational boundaries. The boundary conditions for a time-dependent problem should be physically correct and numerically well posed. Generalized characteristic boundary conditions[15] are used as the time-dependent boundary conditions in the present computation. Nonreflecting inflow/outflow and the inviscid/viscous wall conditions are imposed at the boundaries of the computational domain and the body surfaces, respectively.

A validation of the accuracy of the high-order compact FDSs, the adaptive nonlinear artificial dissipation model, and the generalized characteristic boundary conditions is described in Refs. 12–15. Multidimensional steady/unsteady and inviscid/viscous computations were considered in various benchmark problems: linear wave convection in Ref. 12, acoustic radiation from an axisymmetric baffle in Ref. 13, shock–sound interaction in a transonic nozzle in Ref. 14, and wake flow and acoustic radiation from a circular cylinder in Ref. 15. It was shown that the schemes produced very accurate numerical solutions in comparison with analytic solutions and experimental data.

IV. Decomposition of Computational Domain

Usually, as mentioned in Sec. II, most computational domains with bodies have particular points where grid lines are abruptly bent owing to the body configurations or the grid types. Figure 1 illustrates a simple example of a structured grid with a wedge-type body. The circles in Fig. 1 represent the particular points at which sudden slope changes of the grid lines occur. Obviously, the left- and the right-hand limits of the grid metrics disagree with each other at these points. The discontinuity of the grid metrics leads to a singularity problem.

As a treatment for avoiding the grid singularity, the present paper suggests decomposing the domain into two blocks along the singular line as illustrated in Fig. 1. The singular points form a line interface in two dimensions and a surface interface in three dimensions. After the decomposition, each singular point is divided into two virtual points with different values of the grid metrics, although they are essentially at the same location. The left- and the right-hand limits of the grid metrics are allocated at the singular points of the left and right blocks, respectively. All of the flux derivatives in Eq. (1) are then evaluated independently in each block by using Eqs. (3–6). The differencing stencils do not cross the block interface. Therefore, the high-order FDSs avoid the discontinuity that would be an origin of spurious oscillations. In this way, the singularity problem vanishes in the numerical differentiation. However, this kind of decomposition leaves the blocks isolated without communication. The next step is imposing appropriate interface conditions for the exact communication between the blocks. This is presented in Sec. V.

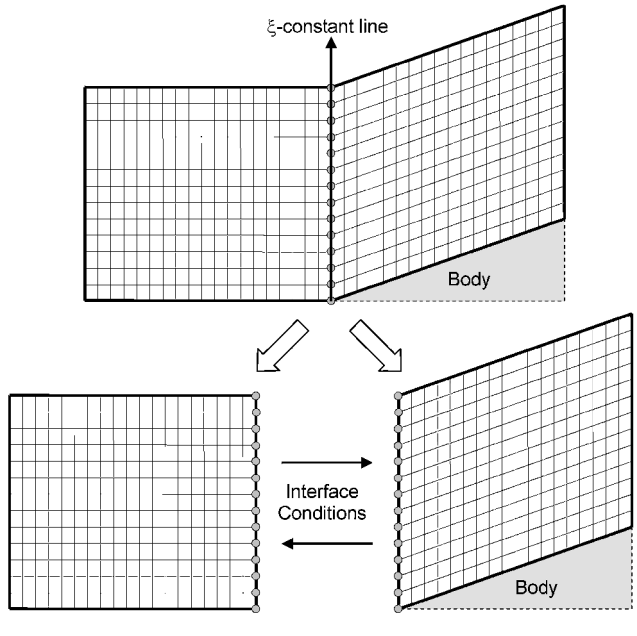


Fig. 1 Decomposing a computational domain into virtual blocks.

Before moving on to the next section, note that the grid singularity is expressed mathematically as $\nabla\xi^L \neq \nabla\xi^R$ ($\xi_x^L \neq \xi_x^R$, or $\xi_y^L \neq \xi_y^R$, or $\xi_z^L \neq \xi_z^R$) at the interface if the interface is represented by a ξ-constant line or surface. The superscripts L and R indicate the left- and the right-hand limits, respectively. However, one should realize that the gradient vectors if normalized by their own magnitudes are exactly the same because their directions are equal. The gradient vector $\nabla\xi$ analytically means a vector perpendicular to the ξ-constant line or surface, that is, the interface. Therefore, the normalized gradient vectors (unit vectors) on the left and the right side of the interface coincide with each other, which is expressed as

$$\nabla\xi^L/|\nabla\xi^L| = \nabla\xi^R/|\nabla\xi^R|$$

If the normalized gradient vector (unit vector) is defined as

$$\nabla\tilde{\xi} \equiv \nabla\xi/|\nabla\xi| \tag{7}$$

then, from Eq. (7), the following equation is satisfied at the interface:

$$\left(\tilde{\xi}_x^L, \tilde{\xi}_y^L, \tilde{\xi}_z^L\right) = \left(\tilde{\xi}_x^R, \tilde{\xi}_y^R, \tilde{\xi}_z^R\right) \tag{8}$$

Equation (7) plays an important role for deriving the interface conditions as described in the next section.

V. Characteristic Interface Conditions

This section presents the block interface conditions for correct physical communication between the blocks that are as yet isolated. They are derived from the characteristic relations that are obtained by transforming the governing equations into quasi-linear wave equations. An advantage of using the characteristic relations is that the flow properties can be classified into the physical wave modes. Characteristic relations have been used extensively for the boundary conditions in many publications during the past two decades.[15] In the present work, they are used for the interface conditions. The derivation of the interface conditions includes no assumption or simplification, which means the interface conditions are mathematically exact. The detailed procedure is explained in the following subsections.

A. Transformation to Characteristic Form

The starting point of the analysis in the derivation of the interface conditions is Eq. (1). Equation (1) can be transformed into a characteristic form in the direction normal to the interface where ξ keeps a constant value. The transformed characteristic equation is a quasi-linear wave equation with a source term. The resulting equation can be expressed as

$$\frac{\partial \boldsymbol{R}}{\partial t} + \bar{\lambda}\frac{\partial \boldsymbol{R}}{\partial \xi} = \boldsymbol{S}_C \tag{9}$$

where the overbar indicates a matrix. Equation (9) is derived using the following two identities in relation to Eq. (2):

$$\delta\boldsymbol{R} = \bar{\boldsymbol{P}}^{-1}\delta\boldsymbol{Q}, \qquad \bar{\lambda}\frac{\partial \boldsymbol{R}}{\partial \xi} = \bar{\boldsymbol{P}}^{-1}\left(\xi_x\frac{\partial \boldsymbol{E}}{\partial \xi} + \xi_y\frac{\partial \boldsymbol{F}}{\partial \xi} + \xi_z\frac{\partial \boldsymbol{G}}{\partial \xi}\right)$$

The source term in Eq. (9) is derived from that in Eq. (1) after some manipulation, as given by

$$\boldsymbol{S}_C = J\bar{\boldsymbol{P}}^{-1}\left\{\hat{\boldsymbol{S}}_V - \left[\boldsymbol{E}\frac{\partial}{\partial \xi}\left(\frac{\xi_x}{J}\right) + \boldsymbol{F}\frac{\partial}{\partial \xi}\left(\frac{\xi_y}{J}\right) + \boldsymbol{G}\frac{\partial}{\partial \xi}\left(\frac{\xi_z}{J}\right) + \frac{\partial \hat{\boldsymbol{F}}}{\partial \eta} + \frac{\partial \hat{\boldsymbol{G}}}{\partial \zeta}\right]\right\}$$

The characteristic differential variables and the corresponding convection speeds are represented as

$$\delta \boldsymbol{R} = \left(\delta\rho - \frac{\delta p}{c^2}, \delta\tilde{W}, \delta\tilde{V}, \frac{\delta p}{\rho c} + \delta\tilde{U}, \frac{\delta p}{\rho c} - \delta\tilde{U} \right)^T$$

$$\bar{\boldsymbol{\lambda}}(\text{diag}) = \left(U, U, U, U + c\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2} \right.$$

$$\left. \times\, U - c\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2} \right)^T \tag{10}$$

where the tilde indicates a quantity normalized by $|\nabla\xi|$ as represented in Eq. (7) and c is the speed of sound. The contravariant (normal) velocity and its differential are given by

$$U = \xi_x u + \xi_y v + \xi_z w, \qquad \delta\tilde{U} = \tilde{\xi}_x \delta u + \tilde{\xi}_y \delta v + \tilde{\xi}_z \delta w$$

whereas the velocity differentials in the parallel direction are expressed as

$$\delta\tilde{V} = -\tilde{\xi}_x \delta v + \tilde{\xi}_y \delta u, \qquad \delta\tilde{W} = \tilde{\xi}_x \delta w - \tilde{\xi}_z \delta u.$$

The matrix $\bar{\boldsymbol{P}}^{-1}$ that transforms the conservative variables into the characteristic variables and its inverse matrix $\bar{\boldsymbol{P}}$ can diagonalize the flux-Jacobian matrices in the direction normal to the interface, and the resulting diagonal terms become the convection speeds of the characteristics. The diagonalizing procedure is described precisely in Refs. 16 and 17. The transformation matrices and their constituting variables are given in the Appendix.

Equation (9) represents the physical (entropy, vorticity, and acoustic) waves with different convection speeds in the direction normal to the interface. An advantage of using the characteristic equation is that the incoming and the outgoing waves can be classified easily by the signs of their convection speeds. An incoming wave calculated within an isolated block is inaccurate because the information outside the block is not used for calculating the convection term in Eq. (9). Therefore, the incoming waves of one block should be compensated by the outgoing waves of the other adjacent block through the strict interface conditions. These conditions are presented in the following subsection.

B. Characteristic Interface Conditions

A crucial interface condition is matching the primitive variables on the left and the right side of the interface, that is, $(\rho^L, \boldsymbol{v}^L, p^L) = (\rho^R, \boldsymbol{v}^R, p^R)$. This condition should be satisfied regardless of time. Therefore, their time derivatives should be matched at the interface, which means $\partial_t(\rho^L, \boldsymbol{v}^L, p^L) = \partial_t(\rho^R, \boldsymbol{v}^R, p^R)$. As a result, from Eqs. (8) and (10), it can be concluded that the time derivatives of the characteristic variables should be also matched:

$$\frac{\partial \boldsymbol{R}^L}{\partial t} = \frac{\partial \boldsymbol{R}^R}{\partial t} \tag{11}$$

In the meantime, if one defines another expression for the convection term in Eq. (9) as

$$\boldsymbol{L} = \bar{\lambda}\frac{\partial \boldsymbol{R}}{\partial \xi}$$

then Eq. (11) is represented by the convection and the source term through Eq. (9) as

$$\boldsymbol{L}^L - \boldsymbol{S}_C^L = \boldsymbol{L}^R - \boldsymbol{S}_C^R \tag{12}$$

Equation (12) describes the interface condition in terms of the space derivatives rather than the time derivatives in Eq. (11), and it is further used for communicating physical information through the interface.

A conventional technique dealing with the characteristic equation is compensating the convection term. Either the left- or the right-hand convection term should be corrected by the other one according to Eq. (12). As already mentioned, the decision of which should be chosen is based on the sign of the convection speed, and the incoming convection term is to be corrected while the outgoing term is maintained. The positive sign represents the outgoing wave from the left block and the incoming wave to the right block. On the contrary, the negative sign represents the incoming wave to the left block and the outgoing wave from the right block. This kind of analysis is for realistic communication between the blocks. In this way, the interface conditions can be summarized as follows:

$$L_m^L = L_m^R - S_{Cm}^R + S_{Cm}^L \quad \text{if} \quad \lambda_m^L \Big/ \left|\lambda_m^L\right| = \lambda_m^R \Big/ \left|\lambda_m^R\right| \le 0$$

$$L_m^R = L_m^L - S_{Cm}^L + S_{Cm}^R \quad \text{if} \quad \lambda_m^L \Big/ \left|\lambda_m^L\right| = \lambda_m^R \Big/ \left|\lambda_m^R\right| \ge 0 \tag{13}$$

where the subscript m represents the components of a vector, $m = 1, \ldots, 5$. Equation (13) is the final form of the characteristic interface conditions. The last process to be presented is the equations of identity relating the characteristic convection term and the flux derivative term.

C. Implementation Procedure

For practical application to actual computations, the detailed procedure for implementing the characteristic interface conditions is summarized by the following steps:

1) All of the flux derivatives in Eq. (1) are evaluated by the high-order FDSs given by Eqs. (3–6) in each isolated block, including the interface points. The normal-flux derivative $\partial\hat{\boldsymbol{E}}/\partial\xi$ on the interface is used as an initial guess to be corrected next.

2) The initial guess of the characteristic convection term on the interface is then evaluated by the following identity:

$$\boldsymbol{L} = J\bar{\boldsymbol{P}}^{-1}\left\{ \frac{\partial \hat{\boldsymbol{E}}}{\partial \xi} - \left[\boldsymbol{E}\frac{\partial}{\partial\xi}\left(\frac{\xi_x}{J}\right) + \boldsymbol{F}\frac{\partial}{\partial\xi}\left(\frac{\xi_y}{J}\right) + \boldsymbol{G}\frac{\partial}{\partial\xi}\left(\frac{\xi_z}{J}\right) \right] \right\}$$

3) The characteristic convection term on each side of the interface is then corrected by imposing the interface conditions given by Eq. (13). The new corrected characteristic convection term is then represented by $\boldsymbol{L}^*$.

4) The normal-flux derivative term is then recalculated by the corrected characteristic convection term using the following identity:

$$\left(\frac{\partial \hat{\boldsymbol{E}}}{\partial \xi}\right)^* = \frac{1}{J}\bar{\boldsymbol{P}}\boldsymbol{L}^* + \left[\boldsymbol{E}\frac{\partial}{\partial\xi}\left(\frac{\xi_x}{J}\right) + \boldsymbol{F}\frac{\partial}{\partial\xi}\left(\frac{\xi_y}{J}\right) + \boldsymbol{G}\frac{\partial}{\partial\xi}\left(\frac{\xi_z}{J}\right) \right]$$

The new corrected normal-flux derivative term with the asterisk is then returned to Eq. (1) for integrating the solutions in time.

5) Finally, after the time integration of the solutions, the primitive variables on the interface need to be refined by averaging the left- and the right-hand values, which is described as

$$(\rho^L, \boldsymbol{v}^L, p^L)^* = (\rho^R, \boldsymbol{v}^R, p^R)^* = (\rho^L + \rho^R, \boldsymbol{v}^L + \boldsymbol{v}^R, p^L + p^R)/2$$

where the asterisk denotes the refined values. This last adjustment is necessary because a certain level of roundoff errors exist in the numerical integration, which, otherwise, has no meaning in the mathematical sense.

This procedure is iterated in every stage of the time-marching steps. In this way, the exact matching condition on the interface and the physical communication between the blocks can be accomplished and accurate numerical solutions can be guaranteed. Numerical implementation is simple and straightforward in that it just needs some additional operations only on the interface points using ready-made matrices.

VI. Application to Actual Computations

In this section, the proposed interface conditions are applied to the actual computations. The mathematical exactness and the numerical accuracy are investigated by solving steady and unsteady benchmark

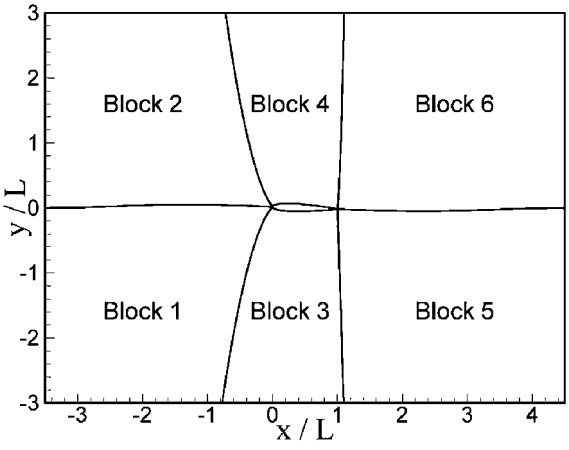


Fig. 2 Computational domain decomposed into six blocks around a NACA0012 airfoil with 2-deg angle of attack.

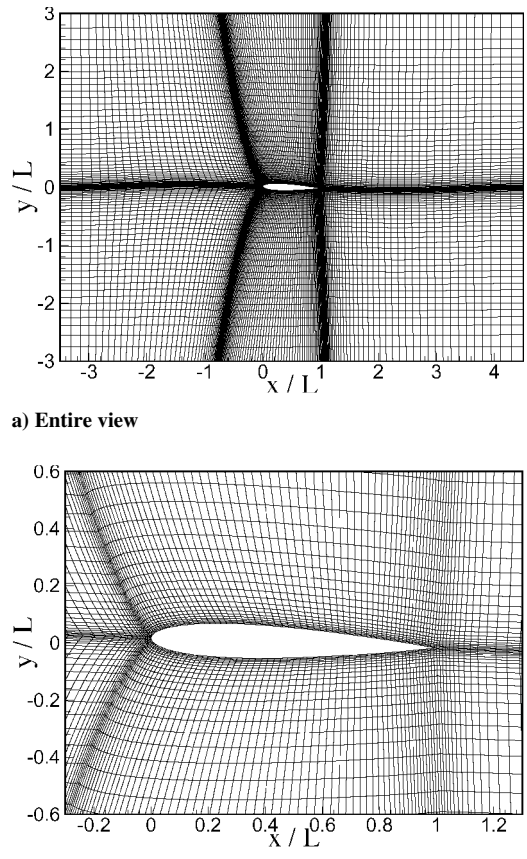
problems that have the numerical and experimental data that can be compared with the present results. Subsonic and transonic inviscid flows past an airfoil are used as the steady problems. A subsonic viscous flow past a circular cylinder is considered as the unsteady problem. This is well known for its regular vortex shedding, that is, the von Kármán vortex street behind the cylinder (see Refs. 18 and 19). The present computations are performed in two dimensions rather than three to reduce the computational complexity and effort. The results of computations are presented next.

A. Steady Inviscid Flows past an Airfoil

Inviscid flows past a NACA0012 airfoil are solved in a subsonic and a transonic regime with small angles of attack. These cases produce the steady-state solutions. Equation (1) without the source term is used in this computation. Two different flow cases are calculated for testing the generality of the interface conditions: $M_\infty = 0.625$ and $\alpha_\infty = 2$ deg and $M_\infty = 0.8$ and $\alpha_\infty = 1.25$ deg, where M_∞ and α_∞ are the freestream Mach number and the angle of attack, respectively. The first case is a subsonic shock-free case and the second is a transonic case that generates shock waves on the upper and lower surfaces of the airfoil. The present results are compared with the previous numerical data to validate the accuracy.

The computational domain is decomposed into six blocks around a NACA0012 airfoil as shown in Fig. 2, where L is the airfoil chord length. In Fig. 2, it seems that the treatment of the leading and trailing edges is somehow complicated because four blocks meet at those locations. In this case, there are four virtual points to be evaluated simultaneously at the leading and trailing edges. At the leading edge, for instance, the interface conditions are implemented for the $\partial\hat{\boldsymbol{E}}/\partial\xi$ terms between blocks 1 and 3 as well as blocks 2 and 4. The interface conditions are also implemented for the $\partial\hat{\boldsymbol{F}}/\partial\eta$ terms between blocks 1 and 3. Meanwhile, the $\partial\hat{\boldsymbol{F}}/\partial\eta$ terms of blocks 2 and 4 are calculated by the wall boundary conditions for simulating the leading edge. The same procedure applies at the trailing edge.

Figure 3 shows the grid system in H topology generated by algebraic mapping with transfinite interpolation.[1] The grid lines are abruptly bent on the six interfaces because of a distinctive feature of the H-topology grid generation. This kind of grid generation is used for the calculation of a helicopter rotor in its rotational direction. The numbers of grid meshes used are 40×40 in blocks 1, 2, 5, and 6 and 64×40 in blocks 3 and 4. The meshes are clustered toward the airfoil, and the smallest mesh size is $\Delta x/L = \Delta y/L = 0.005$ at the leading edge. The time step size is determined by the Courant–Friedrichs–Lewy (CFL) condition with the Courant number of 1.0. The number of iterations used to reach the steady mean solution is 40,000, and the maximum value of error residual $|p^{(n+1)} - p^{(n)}|/p^{(n)}$ falls below 1.0×10^{-7}, which is of the order of machine error.



a) Entire view

b) Zoomed view

Fig. 3 H-topology grid for calculating an inviscid flow past a NACA0012 airfoil.

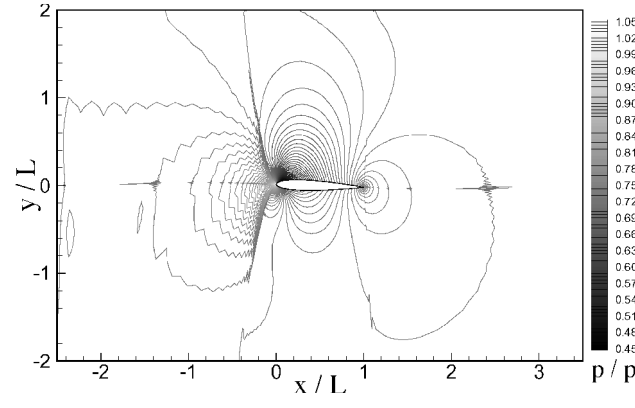
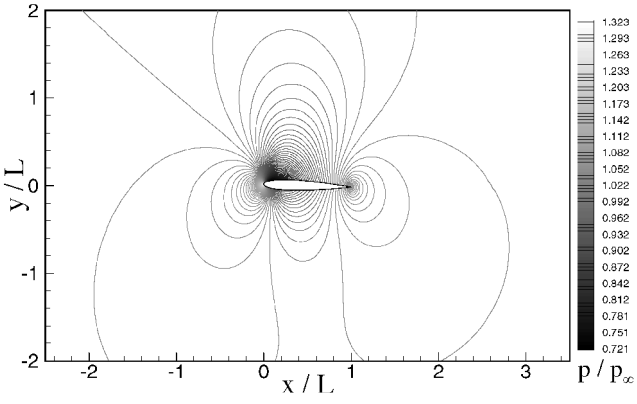


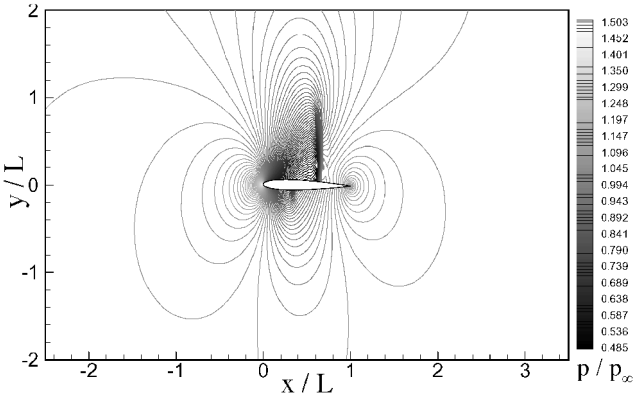
Fig. 4 Result of computation without domain decomposition and interface conditions: pressure contours for $M_\infty = 0.625$ and $\alpha_\infty = 2$ deg.

Without the domain decomposition and the interface conditions, it is found that direct use of the high-order FDSs across the singular points leads to a poor solution. The result of calculating the first problem without the treatment for the grid singularity is shown in Fig. 4. Although the solution does not blow up, the spurious numerical oscillations are generated near the six interfaces as shown in Fig. 4. The steady mean solution is not reached with this kind of approach. Using the low-order schemes and increasing the numerical dissipation can be a help, but it is not supposed to be an essential treatment. This miscellaneous test clearly shows the high sensitivity of the high-order FDSs to the grid singularity.

By means of the domain decomposition and the proposed interface conditions, accurate steady mean solutions can be obtained



a) $M_\infty = 0.625$, $\alpha_\infty = 2$ deg



b) $M_\infty = 0.8$, $\alpha_\infty = 1.25$ deg

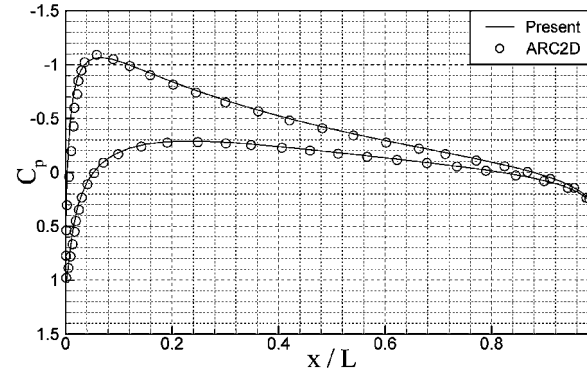
Fig. 5 Pressure contours of steady-state subsonic and transonic inviscid flows past a NACA0012 airfoil.

successfully. The pressure contour plots of the steady mean solutions are given in Fig. 5. Both the shock-free subsonic and the shock-existing transonic cases are solved well without the grid singularity problem. Employing the present methodology produces clear solutions without the spurious oscillations near the interfaces. For quantitative validation of the accuracy of the present solutions, the pressure coefficient distributions on the airfoil surfaces are plotted in Fig. 6. The present solutions are in very good agreement with the previous results obtained by ARC2D,[20] considering that the ARC2D calculation used twice as many grid meshes (256 meshes) around the airfoil surface as the present calculation (128 meshes). It is demonstrated that the present approach can avoid the grid singularity problem for providing high-order accurate solutions of steady inviscid flows.

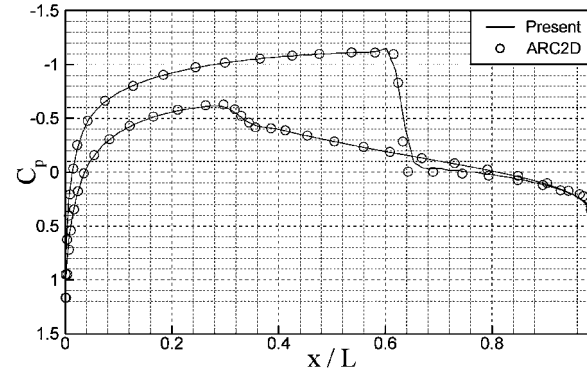
B. Unsteady Viscous Flow past a Circular Cylinder

A viscous flow past a circular cylinder is solved in the subsonic regime. The flow generates regular shedding of vortices known as the von Kármán vortex street. Equation (1) with the complete viscous term is used in this computation. Oscillating lift and drag forces on the cylinder surface due to the periodic vortex shedding are calculated. The Reynolds number $Re_\infty = 400$, based on the freestream velocity and the cylinder diameter, and the freestream Mach number $M_\infty = 0.3$ are imposed on the computation. In the literature, it has been shown that the street of regularly spaced vortices exists with laminar cores over the range of Reynolds numbers from 65 to approximately 400 (Ref. 19). Therefore, it can be considered that the flow is laminar on the whole in this problem. The present solution is compared with previous experimental data to validate the accuracy.

Just as in the earlier case, the computational domain is decomposed into six blocks around a circular cylinder instead of the airfoil

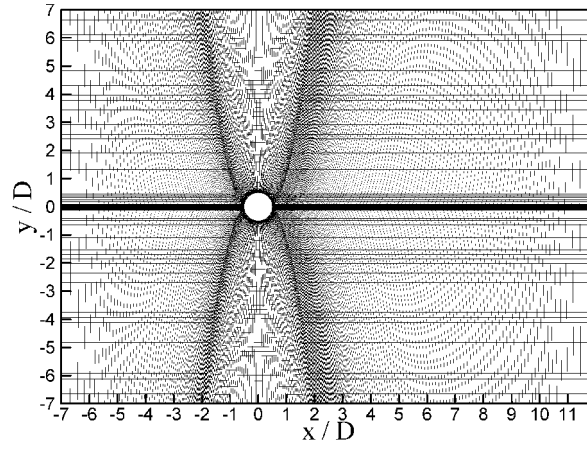


a) Shock-free subsonic case

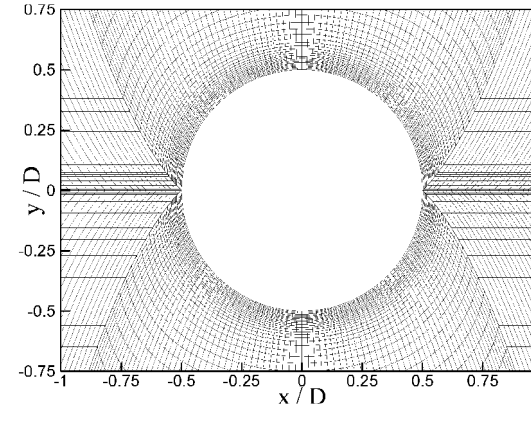


b) Shock-existing transonic case

Fig. 6 Distribution of pressure coefficients on airfoil surfaces.

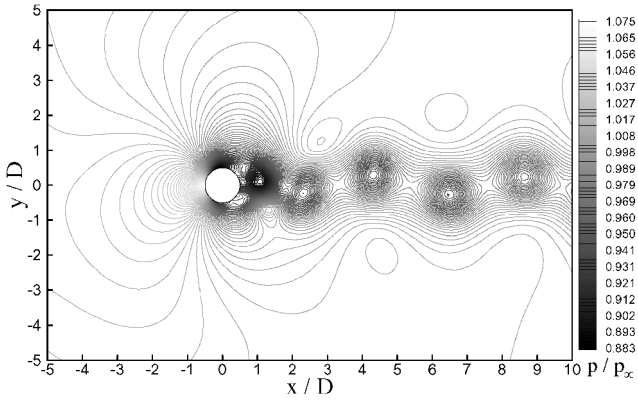


a) Entire view

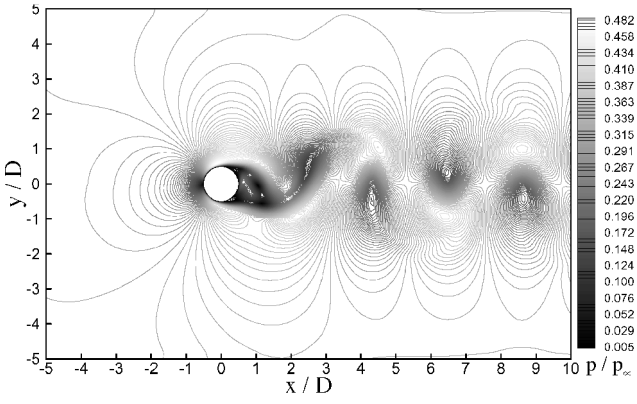


b) Zoomed view

Fig. 7 H-topology grid for calculating a viscous flow past a circular cylinder.



a) Pressure contours



b) Mach number contours

Fig. 8 Unsteady solution for a viscous flow past a circular cylinder.

as in Fig. 2. Figure 7 shows the H-topology grid system, where D is the cylinder diameter. Usually, this kind of grid generation is not used for a single cylinder. However, it could be very useful for multiple cylinders in a domain, for example. The present application is only for verification of the method. The grid lines are more seriously bent at the interfaces than those of the airfoil problem because of the high curvature of the circular cylinder. The numbers of grid meshes used are 64×64 for blocks 1 and 2, 80×64 for blocks 3 and 4, and 128×64 for blocks 5 and 6. The grid meshes are so clustered toward the cylinder surface that the boundary layer can be resolved in about 20 points on an average in the radial direction. The minimum grid size at the rear side of the cylinder is given by $\Delta x/D = 0.005$ and $\Delta y/D = 0.02$. The computation is continued until the nondimensional time reaches $u_\infty t/D = 204.0$, where u_∞ is the freestream velocity. Time step size is determined by the CFL condition with Courant number of 1.0, and the number of iterations is 200,000.

As a result of the computation, pressure and Mach number contour plots at a time step are presented in Fig. 8 to visualize the von Kármán vortex street. It shows the attached boundary layer on the front surface of the cylinder, the mixing layers of the separated flows, and the downstream flow patterns well. The positions of the shed vortices are revealed clearly. Just like the preceding steady problems, employing the present methodology produces a clear solution for the unsteady viscous problem without spurious oscillations near the interfaces. However, there is a possible topic to be discussed for the viscous case. The characteristic interface conditions proposed here focus only on the Euler flux derivatives rather than the viscous flux derivatives. All of the viscous terms are considered as a source term. An additional interface condition regarding the viscous flux derivatives may be required in the future for precise computation of turbulent flows such as large-eddy simulation and direct numerical simulation in this kind of grid system.

Table 2 Comparison of Strouhal number, mean drag coefficient, and rms lift coefficient

Case	Sr	$\overline{C_D}$	$\overline{C_L{}^{\frac{1}{2}}}$
Experimental data	0.21 ± 0.005	1.2 ± 0.1	0.6 ± 0.1
Present results	0.210	1.216	0.622

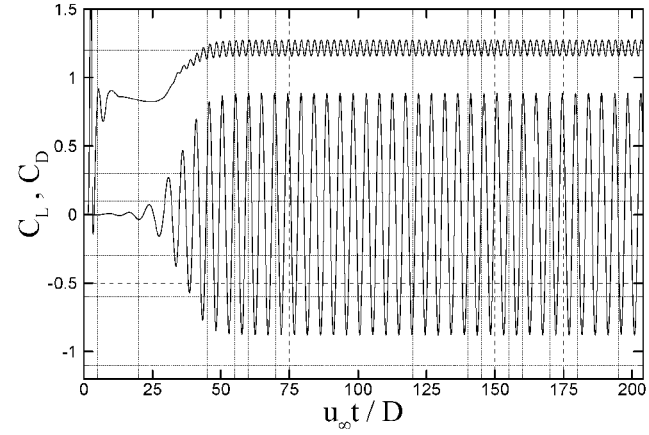


Fig. 9 Time-dependent signals of lift and drag coefficients over a circular cylinder.

The lift and drag forces exerted on the cylinder surface fluctuate in time due to the periodic vortex shedding. The Strouhal number $Sr = f_s D/u_\infty$ of the periodic vortex shedding can be measured by evaluating the frequency of the oscillating lift and drag forces. Time-dependent signals of the lift and drag coefficients are presented in Fig. 9. Those forces are calculated by integrating the distributions of pressure and viscous stresses on the cylinder surface. As shown in Fig. 9, after a transient state due to the impulsive initial conditions, constant frequency and constant magnitudes of the fluctuations are achieved in the periodic shedding mode. The calculated values of Strouhal number, mean drag coefficient, and RMS lift coefficient are listed in Table 2 compared with the experimental data obtained in low-speed flows of $Re_\infty = 400$ (Refs. 18 and 19). The present results are in very good agreement with the experimental measurements. It has been demonstrated that the present approach can avoid the grid singularity problem and provide high-order accurate solutions of unsteady viscous flows.

VII. Conclusions

High-order computations of compressible flows are successfully accomplished on singular structured grids. Decomposing a computational domain into two virtual blocks along singular grid points and employing noncentral or one-sided FDSs not crossing the block interfaces result in no spurious numerical oscillation that would have been generated by central FDSs across the singularity. The proposed interface conditions enable the correct physical communication between the blocks that are disconnected by the virtual decomposition. The mathematical exactness and the numerical accuracy of the proposed interface conditions have been sufficiently tested by actual computations. It is shown that the present approach can provide high-order accurate solutions of steady/unsteady and inviscid/viscous problems without difficulties. It is expected that the virtual multiblock decomposition with the proposed interface conditions can be applied to various high-order computations with complex bodies while avoiding the grid singularity problem. Further tests are desirable to verify the generality of the present methodology in future studies.

Appendix: Transformation Matrices and Constituting Variables

The transformation matrices between the conservative and the characteristic variables and their constituting variables are given as follows:

$$\bar{\boldsymbol{P}}^{-1}=\begin{pmatrix} \boldsymbol{B}_0\cdot\boldsymbol{l}_x & (\gamma-1)\frac{u}{c^2}\tilde{\xi}_x & (\gamma-1)\frac{v}{c^2}\tilde{\xi}_x+\frac{\tilde{\xi}_z}{\rho} & (\gamma-1)\frac{w}{c^2}\tilde{\xi}_x-\frac{\tilde{\xi}_y}{\rho} & -\frac{\gamma-1}{c^2}\tilde{\xi}_x \\ \boldsymbol{B}_0\cdot\boldsymbol{l}_y & (\gamma-1)\frac{u}{c^2}\tilde{\xi}_y-\frac{\tilde{\xi}_z}{\rho} & (\gamma-1)\frac{v}{c^2}\tilde{\xi}_y & (\gamma-1)\frac{w}{c^2}\tilde{\xi}_y+\frac{\tilde{\xi}_x}{\rho} & -\frac{\gamma-1}{c^2}\tilde{\xi}_y \\ \boldsymbol{B}_0\cdot\boldsymbol{l}_z & (\gamma-1)\frac{u}{c^2}\tilde{\xi}_z+\frac{\tilde{\xi}_y}{\rho} & (\gamma-1)\frac{v}{c^2}\tilde{\xi}_z-\frac{\tilde{\xi}_x}{\rho} & (\gamma-1)\frac{w}{c^2}\tilde{\xi}_z & -\frac{\gamma-1}{c^2}\tilde{\xi}_z \\ \frac{c}{\rho}\left(\frac{\gamma-1}{2}M^2-\frac{\boldsymbol{v}\cdot\boldsymbol{l}_\xi}{c}\right) & \boldsymbol{C}_+\cdot\boldsymbol{l}_x & \boldsymbol{C}_+\cdot\boldsymbol{l}_y & \boldsymbol{C}_+\cdot\boldsymbol{l}_z & \frac{\gamma-1}{\rho c} \\ \frac{c}{\rho}\left(\frac{\gamma-1}{2}M^2+\frac{\boldsymbol{v}\cdot\boldsymbol{l}_\xi}{c}\right) & \boldsymbol{C}_-\cdot\boldsymbol{l}_x & \boldsymbol{C}_-\cdot\boldsymbol{l}_y & \boldsymbol{C}_-\cdot\boldsymbol{l}_z & \frac{\gamma-1}{\rho c} \end{pmatrix}$$

$$\boldsymbol{B}_0=\left(1-\frac{\gamma-1}{2}M^2\right)\boldsymbol{l}_\xi-\frac{1}{\rho}(\boldsymbol{v}\times\boldsymbol{l}_\xi),\qquad \boldsymbol{C}_\pm=\pm\frac{\boldsymbol{l}_\xi}{\rho}-\frac{\gamma-1}{\rho c}\boldsymbol{v},\quad \boldsymbol{v}=(u,v,w)^T$$

$$\bar{\boldsymbol{P}}=\begin{pmatrix} \tilde{\xi}_x & \tilde{\xi}_y & \tilde{\xi}_z & \frac{\rho}{2c} & \frac{\rho}{2c} \\ u\tilde{\xi}_x & u\tilde{\xi}_y-\rho\tilde{\xi}_z & u\tilde{\xi}_z+\rho\tilde{\xi}_y & \frac{\rho}{2c}(u+\tilde{\xi}_x c) & \frac{\rho}{2c}(u-\tilde{\xi}_x c) \\ v\tilde{\xi}_x+\rho\tilde{\xi}_z & v\tilde{\xi}_y & v\tilde{\xi}_z-\rho\tilde{\xi}_x & \frac{\rho}{2c}(v+\tilde{\xi}_y c) & \frac{\rho}{2c}(v-\tilde{\xi}_y c) \\ w\tilde{\xi}_x-\rho\tilde{\xi}_y & w\tilde{\xi}_y+\rho\tilde{\xi}_x & w\tilde{\xi}_z & \frac{\rho}{2c}(w+\tilde{\xi}_z c) & \frac{\rho}{2c}(w-\tilde{\xi}_z c) \\ \boldsymbol{b}\cdot\boldsymbol{l}_x & \boldsymbol{b}\cdot\boldsymbol{l}_y & \boldsymbol{b}\cdot\boldsymbol{l}_z & \frac{\rho}{2c}(H+c\boldsymbol{v}\cdot\boldsymbol{l}_\xi) & \frac{\rho}{2c}(H-c\boldsymbol{v}\cdot\boldsymbol{l}_\xi) \end{pmatrix}$$

$$\boldsymbol{b}=\frac{|\boldsymbol{v}|^2}{2}\boldsymbol{l}_\xi+\rho(\boldsymbol{v}\times\boldsymbol{l}_\xi),\qquad H=\frac{|\boldsymbol{v}|^2}{2}+\frac{c^2}{\gamma-1}$$

where $\boldsymbol{l}_x$, $\boldsymbol{l}_y$, and $\boldsymbol{l}_z$ are the unit vectors in the x, y, and z directions, respectively, and $\boldsymbol{l}_\xi$ is the unit normal vector defined by Eq. (7) in Sec. IV.

References

[1]Fletcher, C. A. J., *Computational Techniques for Fluid Dynamics*, 2nd ed., Vol. 2, Springer-Verlag, New York, 1991, pp. 47–127.

[2]Hoffmann, K. A., and Chiang, S. T., *Computational Fluid Dynamics for Engineers*, 1st ed., Vol. 1, Engineering Education System, Wichita, KS, 1993, pp. 344–411.

[3]Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, 2nd ed., Taylor and Francis, Washington, DC, 1997, pp. 679–714.

[4]Kurbatskii, K. A., and Tam, C. K. W., "Cartesian Boundary Treatment of Curved Walls for High-Order Computational Aeroacoustics Schemes," *AIAA Journal*, Vol. 35, No. 1, 1997, pp. 133–140.

[5]Dyson, R. W., and Goodrich, J. W., "Automated Approach to Very High-Order Aeroacoustic Computations," *AIAA Journal*, Vol. 39, No. 3, 2001, pp. 396–406.

[6]LeVeque, R. J., *Numerical Methods for Conservation Laws*, 2nd ed., Birkhäuser Verlag, Boston, 1992, pp. 1–13.

[7]Kim, J. W., and Morris, P. J., "Numerical Simulation of Subsonic Inviscid Flow past Cones Using High-Order Schemes," *AIAA Journal*, Vol. 40, No. 10, 2002, pp. 1961–1968.

[8]Freund, J. B., Lele, S. K., and Moin, P., "Numerical Simulation of a Mach 1.92 Turbulent Jet and Its Sound Field," *AIAA Journal*, Vol. 38, No. 11, 2000, pp. 2023–2031.

[9]Hixon, R., and Mankbadi, R. R., "Validation of High-Order Prefactored Compact Scheme on Nonlinear Flows with Complex Geometries," *Third Computational Aeroacoustics Workshop on Benchmark Problems*, NASA CP 2000-209790, 2000, pp. 117–132.

[10]Tam, C. K. W., and Kurbatskii, K. A., "Microfluid Dynamics and Acoustics of Resonant Liners," *AIAA Journal*, Vol. 38, No. 8, 2000, pp. 1331–1339.

[11]Lockard, D. P., and Morris, P. J., "Radiated Noise from Airfoils in Realistic Mean Flows," *AIAA Journal*, Vol. 36, No. 6, 1998, pp. 907–914.

[12]Kim, J. W., and Lee, D. J., "Optimized Compact Finite Difference Schemes with Maximum Resolution," *AIAA Journal*, Vol. 34, No. 5, 1996, pp. 887–893.

[13]Kim, J. W., and Lee, D. J., "Implementation of Boundary Conditions for Optimized High-Order Compact Schemes," *Journal of Computational Acoustics,* Vol. 5, No. 2, 1997, pp. 177–191.

[14]Kim, J. W., and Lee, D. J., "Adaptive Nonlinear Artificial Dissipation Model for Computational Aeroacoustics," *AIAA Journal*, Vol. 39, No. 5, 2001, pp. 810–818.

[15]Kim, J. W., and Lee, D. J., "Generalized Characteristic Boundary Conditions for Computational Aeroacoustics," *AIAA Journal*, Vol. 38, No. 11, 2000, pp. 2040–2049.

[16]Warming, R. F., Beam, R. M., and Hyett, B. J., "Diagonalization and Simultaneous Symmetrization of Gas-Dynamic Matrices," *Mathematics of Computation*, Vol. 29, No. 132, 1975, pp. 1037–1045.

[17]Hirsch, C., *Numerical Computation of Internal and External Flows*, 1st ed., Vol. 2, Wiley, New York, 1992, pp. 132–223.

[18]Schlichting, H., *Boundary Layer Theory*, 7th ed., McGraw–Hill, New York, 1979, pp. 24–46.

[19]Blake, W. K., *Mechanics of Flow-Induced Sound and Vibration*, 1st ed., Vol. 1, Academic Press, New York, 1986, pp. 219–287.

[20]Pulliam, T. H., and Steger, J. L., "Implicit Finite Difference Simulation of Three-Dimensional Compressible Flow," *AIAA Journal*, Vol. 18, No. 2, 1980, pp. 159–167.

S. Mahalingam
Associate Editor