AIAA JOURNAL
Vol. 43, No. 5, May 2005

Technical Notes

TECHNICAL NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes cannot exceed six manuscript pages and three figures; a page of text may be substituted for a figure and vice versa. After informal review by the editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Comparative Study of Single-Block vs Multiblock Jet Flow Computations

Ali Uzun* and M. Yousuff Hussaini†
Florida State University,
Tallahassee, Florida 32306-4120
and
Craig L. Streett‡
NASA Langley Research Center,
Hampton, Virginia 23681

Introduction

ALTHOUGH spectral methods with their exponential order of accuracy have been the prevalent tool for direct numerical simulation and large-eddy simulation (LES) of turbulent flows,[1,2] high-order-accurate compact finite-difference schemes[3,4] have become increasingly popular in the computation of turbulent flow and aeroacoustics problems of practical interest. Such schemes, while retaining the simplicity of low-order methods, offer significant advantages, such as good spectral-like resolution with consequent reduction in dispersion and dissipation errors. Although the application of compact schemes to a relatively simple domain is straightforward, their application to complex geometry inevitably requires a multiblock approach. The main idea of a multiblock approach is to break up a complex domain into several relatively simple subdomains so that the single-block computation can be simultaneously carried out in each subdomain. The implicit nature of compact schemes (stemming from the implicit solution of a system of equations along a given spatial direction required to compute the spatial derivatives along that direction) poses certain issues at the interfaces of these subdomains. Gaitonde and Visbal[5,6] and Visbal and Gaitonde[7] have addressed these issues in their multiblock implementation strategy through overlapping grids that permit data exchange between adjacent subdomains during the course of the computations. This technique is attractive in that it allows high-order-accurate simulations to be performed on complicated geometries using multiblock structured grids.

Although the aforementioned studies[5–7] have shown that the multidomain computations can recover the single-block computation results fairly well in relatively simple problems such as two-dimensional vortex convection, their application to a realistic three-dimensional turbulence simulation has yet to be demonstrated. It is the purpose of the present Note to report a comparative study of single-block vs multiblock three-dimensional jet flow computations using high-order-accurate compact schemes and demonstrate that multiblock computations can satisfactorily recover the single-block solution of a realistic jet flow problem. The turbulent nature of a three-dimensional jet makes the problem a lot more complicated and challenging than a single vortex convecting in a two-dimensional domain. In the multiblock computation of three-dimensional turbulent flow, there will be turbulence passing from a block to its neighbor, and the multiblock strategy should be able to allow the passing of turbulence from block to block as smoothly as possible. Obviously, the simple problems studied in the past cannot evaluate the effectiveness of the multiblock scheme in transferring realistic turbulence from block to block. Thus, it is of interest to see how the multiblock computation strategies that have worked well for simple problems will perform when it comes to solving a more realistic turbulence problem. The computations in this work are based on large-eddy simulations because the grids used in the simulations resolve only the large scales of turbulence. The spatial filter implemented in the single-block and multiblock codes is treated as an implicit subgrid-scale (SGS) model,[8] and no explicit SGS modeling is used as this work concerns algorithm rather than subgrid-scale model.

Governing Equations for LES

The governing equations for LES are obtained by applying a spatial filter to the Navier–Stokes equations in order to remove the small scales. The Favre-filtered unsteady, compressible, nondimensionalized Navier–Stokes equations formulated in curvilinear coordinates[9] are solved in this study using the numerical method described in the next section.

Numerical Method

A single-block LES code that was previously developed by the first author during his doctoral dissertation research[9] has been recently extended to a multiblock version. A multiblock implementation strategy similar to that of Gaitonde and Visbal[5,6] is incorporated in the multiblock version of the LES code with a seven-point grid overlap between neighboring blocks.

We now consider an individual block of the multiblock grid or the single-block grid. The nonuniformly spaced curvilinear grid in physical space is transformed to a uniform grid in computational space, and the discretized governing equations are solved on the uniform grid. The previous single-block LES code[9] was based on the standard sixth-order-accurate compact scheme from Lele.[3] As the optimized prefactored compact schemes recently developed by Ashcroft and Zhang[4] offer improved dispersion characteristics compared to the standard sixth- and eighth-order compact schemes, they are employed in the single-block and multiblock versions of the LES code. The prefactorization strategy splits the central implicit schemes into forward and backward biased operators. The prefactored optimized schemes require the solution of two independent bidiagonal matrices. The spatial derivative at a given grid point i is given by

$$D_i = \tfrac{1}{2}\left(D_i^F + D_i^B\right) \tag{1}$$

where D_i^F and D_i^B represent the forward and backward operators, respectively. The prefactored compact scheme we use is a fourth-order scheme with a five-point stencil. Hence, the forward and backward

Received 14 September 2004; accepted for publication 23 November 2004.

*Postdoctoral Research Associate, School of Computational Science. Member AIAA.

†Sir James Lighthill Professor of Mathematics and Computational Science and Engineering, School of Computational Science.

‡Aerospace Technologist, Computational Modeling and Simulation Branch.

operators are given as follows:

$$\tfrac{1}{2}D_i^F = (1/2\beta\Delta\xi)[b(f_{i+1}-f_i)+d(f_{i-1}-f_i) + e(f_{i-2}-f_i)] - (\alpha/2\beta)D_{i+1}^F \quad (2)$$

$$\tfrac{1}{2}D_i^B = (1/2\beta\Delta\xi)[b(f_i-f_{i-1})+d(f_i-f_{i+1}) + e(f_i-f_{i+2})] - (\alpha/2\beta)D_{i-1}^B \quad (3)$$

where the coefficients α, β, b, d, e are given by Ashcroft and Zhang,[4] f_i denotes the value of f (the quantity whose spatial derivative is being computed) at grid point i, and $\Delta\xi$ is the uniform grid spacing along a given spatial direction in the computational space. The preceding scheme is applicable to grid points $i=3$ through $i=N-2$, where N is the number of grid points along the spatial direction. Additional boundary stencils are required at grid points $i=1$, $i=2$, $i=N-1$, and $i=N$. The third-order boundary stencils we use are given by Ashcroft and Zhang.[4] At grid points $i=2$ and $i=N-1$, the following explicit backward and forward stencils are used:

$$D_2^B = \frac{1}{\Delta\xi}\sum_{i=1}^{4} s_i f_i, \qquad D_{N-1}^B = \frac{1}{\Delta\xi}\sum_{i=N-3}^{N} e_i f_i \quad (4)$$

$$D_2^F = \frac{1}{\Delta\xi}\sum_{i=1}^{4} -e_{N+1-i} f_i, \qquad D_{N-1}^F = \frac{1}{\Delta\xi}\sum_{i=N-3}^{N} -s_{N+1-i} f_i \quad (5)$$

The coefficients $s_i\,(i=1,\ldots,4)$ and $e_i\,(i=N-3,\ldots,N)$ are given by Ashcroft and Zhang.[4] At grid points $i=1$ and $i=N$, the following one-sided explicit schemes are employed to compute the spatial derivative:

$$D_1 = \frac{1}{\Delta\xi}\sum_{i=1}^{4} c_i f_i, \qquad D_N = \frac{1}{\Delta\xi}\sum_{i=N-3}^{N} -c_{N+1-i} f_i \quad (6)$$

where $c_1=-33/18$, $c_2=3$, $c_3=-3/2$, and $c_4=1/3$.

The viscous stress terms appearing in the governing equations are obtained using the first derivatives of the velocity components. For computational efficiency, the terms in the inviscid and viscous fluxes are first added together, and then the total flux is differentiated to compute the right-hand side of the governing equations. As a consequence, the second derivatives in the viscous fluxes are essentially evaluated by applying the first-derivative operator twice. Although this approach is not as accurate as when a compact scheme is used to directly compute the second derivatives in the viscous terms, it is much cheaper to implement in curvilinear coordinates.

The LES codes employ the following sixth-order tridiagonal spatial filter used by Visbal and Gaitonde[7]:

$$\alpha_f \bar{f}_{i-1} + \bar{f}_i + \alpha_f \bar{f}_{i+1} = \sum_{n=0}^{3} \frac{a_n}{2}(f_{i+n} + f_{i-n}) \quad (7)$$

where $\bar{f}_i$ denotes the filtered value of quantity f at grid point i, and

$$a_0 = 11/16 + 5\alpha_f/8, \qquad a_1 = 15/32 + 17\alpha_f/16$$
$$a_2 = -3/16 + 3\alpha_f/8, \qquad a_3 = 1/32 - \alpha_f/16 \quad (8)$$

The parameter α_f must satisfy the inequality $-0.5 < \alpha_f < 0.5$. A less dissipative filter is obtained with higher values of α_f within the given range. With $\alpha_f = 0.5$, there is no filtering effect.

Because this filter has a seven-point right-hand side stencil, it obviously cannot be used at near-boundary points. Instead, the following sixth-order equation with a one-sided right-hand side stencil is used at grid points near the left boundary point $i=1$:

$$\alpha_f \bar{f}_{i-1} + \bar{f}_i + \alpha_f \bar{f}_{i+1} = \sum_{n=1}^{7} a_{n,i} f_n, \qquad i = 2, 3 \quad (9)$$

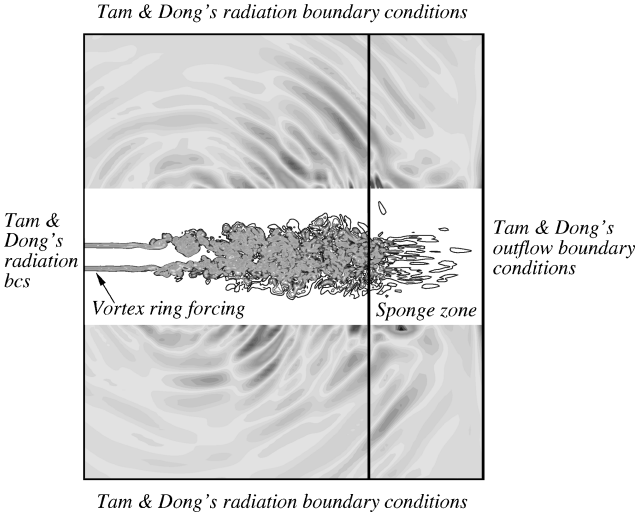

Fig. 1 Schematic of the boundary conditions. Vorticity magnitude contours are shown within the jet while divergence of velocity contours are shown in the outer part of the computational domain.

where the coefficients $a_{n,i}$ are given by Visbal and Gaitonde.[7] For the points near the right boundary point $i=N$, we apply a similar formulation.[7] The boundary points $i=1$ and $i=N$ are left unfiltered.

The standard fourth-order, four-stage explicit Runge–Kutta scheme is used for time advancement. We apply Tam and Dong's three-dimensional radiation and outflow boundary conditions[10] on the boundaries of the computational domain as illustrated in Fig. 1. The original two-dimensional boundary conditions of Tam and Dong were extended to three dimensions by Bogey and Bailly.[11] Tam and Dong's radiation boundary conditions are applied on the boundaries to which only acoustic disturbances are reaching. For the boundaries that entropy, vorticity and acoustic waves are crossing, Tam and Dong's outflow boundary conditions are used. In addition, we use the sponge zone method[12] in which grid stretching and artificial damping are applied to dissipate the vortices present in the flowfield before they hit the outflow boundary. This way, unwanted reflections from the outflow boundary are suppressed. Because the actual nozzle geometry is not included in the present calculations, randomized velocity perturbations in the form of a vortex ring are added into the jet shear layers at a short distance downstream of the inflow boundary in order to excite the three-dimensional instabilities in the jet and cause the potential core of the jet to break up at a reasonable distance downstream of the inflow boundary. This forcing procedure has been adapted from Bogey et al.[13]

A data-transposition strategy[9] is used for parallelizing the single-block code. For the multiblock code, the solution is advanced independently in each block at every time step. Information is exchanged among the blocks at the end of each stage of the Runge–Kutta time integration and after each application of the filter. More details of the multiblock computation strategy can be found in Gaitonde and Visbal.[5,6] Space does not permit us to discuss the performance comparison of the two codes here. However, for a given grid size and number of processors, the multiblock computation is faster than the single-block one because a lesser amount of data needs to be transferred among the blocks during the course of the multiblock computation. The data-transposition strategy[9] used for parallelizing the single-block code is relatively slow because it requires a significant amount of communication.

Results

We present results from LES for a turbulent isothermal round jet at a Mach number of 0.9 and Reynolds number $Re_D = \rho_0 U_0 D_j/\mu_0 = 400,000$, where ρ_0, U_0, μ_0 are the jet centerline density, velocity, and viscosity at the nozzle exit, respectively, and D_j is the jet diameter. The jet centerline temperature is chosen to be the same as the ambient temperature and set to 286 K. A fully curvilinear grid

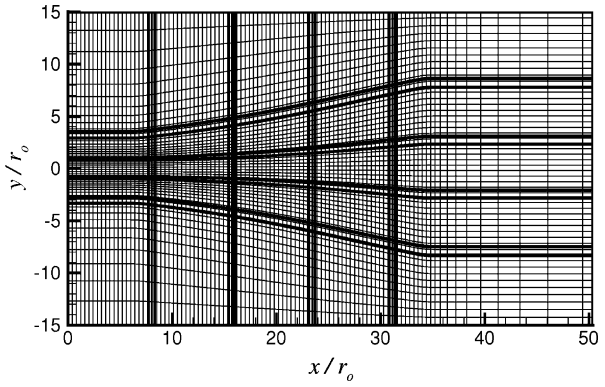

Fig. 2 Partitioning of the computational domain for the multiblock simulation. (Every fourth grid node is shown for clarity.)

consisting of approximately 16 million grid points ($N_x \times N_y \times N_z = 390 \times 200 \times 200$) was used in the simulation. The physical portion of the domain extended to 35 jet radii in the streamwise direction and ± 15 jet radii in the transverse directions. A top-hat mean streamwise velocity profile defined by a hyperbolic tangent function was imposed on the inflow boundary. There are about 14 grid points in the initial jet shear layer.

The LES is performed first as a single-block computation and then as multiblock computation. The single-block computation is performed on 200 processors in parallel. As mentioned earlier, a data-transposition strategy[9] is used for parallelizing the single-block computations. For the multiblock computation, the domain is divided into five partitions along each of the three spatial directions. Hence, the domain is partitioned into 125 approximately equal-sized blocks. Figure 2 shows a two-dimensional cross section of the computational grid on the xy plane and the alignment of the blocks. The thick dark lines correspond to the block boundaries. The contiguous blocks overlap over seven grid points. Initially, the grid points are clustered around the shear layer of the jet in order to accurately resolve the shear layer. After the potential core of the jet breaks up at around $x = 10$ jet radii, the grid is stretched in the y and z directions in order to redistribute the grid points and achieve a more uniform distribution.

Both LES computations are carried out without an explicit SGS model, and the tridiagonal spatial filter is treated as an implicit SGS model as the focus is on the algorithm rather than on the subgrid-scale model. The filtering parameter is set to $\alpha_f = 0.47$. Both simulations run for a total of 80,000 time steps. The initial transients exit the domain over the first 10,000 time steps. An ambient sound wave travels about six times the physical domain length in the streamwise direction in 10,000 time steps. We then collect the flow statistics over the next 70,000 time steps.

To compare the results of the two simulations, we first plot the jet mean centerline velocity decay and the half-velocity radius growth as shown in Fig. 3. In this figure, U_c is the mean jet centerline velocity at a given axial location, U_j is the jet centerline velocity at the nozzle exit, x is the streamwise location along the jet centerline, and r_0 is the initial jet radius. The half-velocity radius $r_{1/2}$ at a given downstream location is defined as the radial location where the mean streamwise velocity is one-half of the jet mean centerline velocity at that location. As can be seen from the figure, the jet centerline mean velocity decay and the half-velocity radius growth are similar in both simulations. However, the slope of the linear growth of the half-velocity radius profile or the so-called jet spreading rate in the fully turbulent downstream region seems to be slightly larger for the multiblock computation. The jet spreading rate is found to be 0.056587 and 0.063112 for the single- and multiblock computation, respectively. Hence, we see that the multiblock result differs by about 11.5% relative to the single-block result. One can also compute the jet decay coefficient by using the jet centerline decay profile. The jet decay coefficient is equal to one-half divided by the slope of the linear growth of the $U_j/U_c = (U_c/U_j)^{-1}$ profile in the fully turbulent region. The jet decay coefficient is 9.58 and 8.62 for

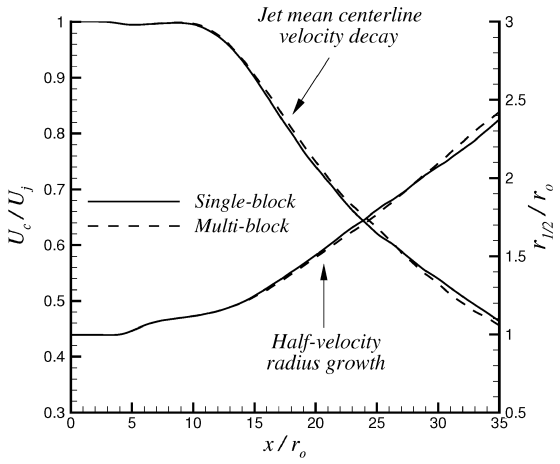

Fig. 3 Jet mean centerline velocity decay and half-velocity radius growth profiles.

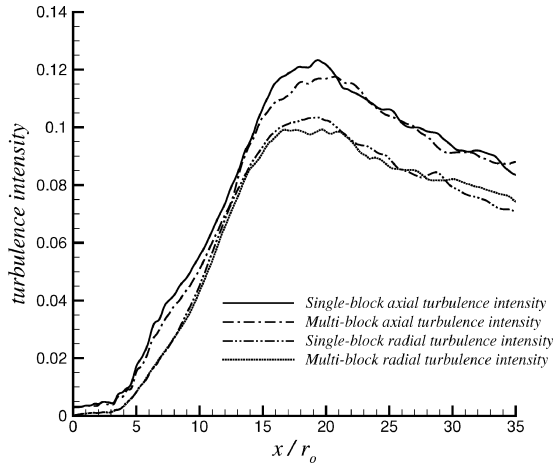

Fig. 4 Axial and radial turbulence intensity profiles along the centerline.

the single- and multiblock computation, respectively. This time, we see that the multiblock jet decay coefficient differs by about 10% relative to the single-block result.

Figure 4 depicts the axial and radial turbulence intensity profiles along the jet centerline. The axial and radial turbulence intensities at a given point are equal to the rms of the axial and radial velocity fluctuations, respectively, which are normalized by U_j. As can be seen from the figure, once again the single-block and multiblock results are quite similar. The maximum difference in the axial and radial turbulence intensity profiles is about 6.5 and 5.4%, respectively, relative to the single-block results.

Next, we pick a downstream region in the fully turbulent regime of the jet and analyze the mean streamwise velocity and Reynolds-stress profiles from both simulations. For this purpose, we choose the downstream location $x = 30r_0$. Properly scaled mean streamwise velocity and the normalized Reynolds stress σ_{xx}, σ_{rr}, σ_{rx} profiles at the downstream location of $x = 30r_0$ are plotted in Fig. 5. The self-similarity coordinate is taken as $r/r_{1/2}$, where r is the radial location and $r_{1/2}$ is the half-velocity radius. The vertical axis of the velocity profiles is normalized by the mean jet centerline velocity at the given axial location. From the figure, it can be observed that the single-block and multiblock mean streamwise velocity profiles are almost identical. The normalized Reynolds stresses in cylindrical coordinates are defined as follows:

$$\sigma_{xx} = \langle v_x' v_x' \rangle / U_c^2, \qquad \sigma_{rr} = \langle v_r' v_r' \rangle / U_c^2, \qquad \sigma_{rx} = \langle v_r' v_x' \rangle / U_c^2 \tag{10}$$

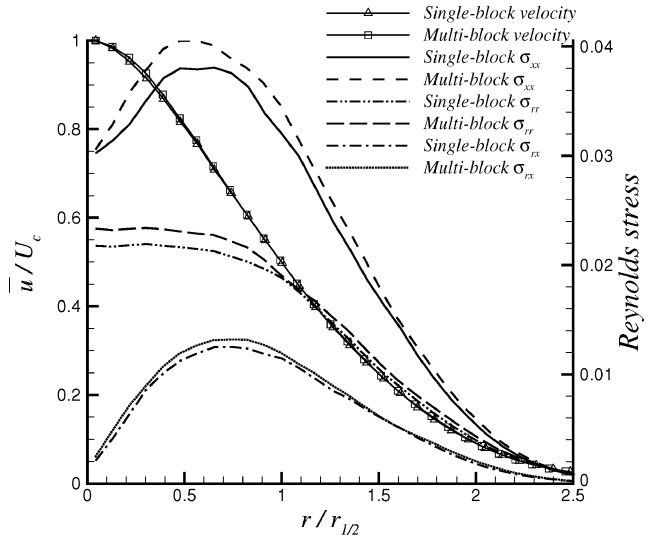

Fig. 5 Mean streamwise velocity profiles and normalized Reynolds-stress profiles at $x = 30\ r_0$.

where v_x' and v_r' are the axial and radial components of the fluctuating velocity, respectively; U_c is the mean jet centerline velocity at the given axial location; and $\langle\ \rangle$ denotes time averaging. From the figure, it can now be seen that the Reynolds-stress agreement between single-block and multiblock results is encouraging. Although the two sets of results are not exactly identical, the maximum difference between the two sets of results is less than 7% relative to the single-block results.

The differences observed between single- and multiblock results are not totally unexpected. Such differences are believed to be related to the fact that the multiblock implementation inevitably introduces some small errors into the computation as one-sided compact differencing, and spatial filtering formulas are applied near the block interface boundaries. Despite the observed differences, the multiblock computation results are found to be quite satisfactory in this study. We should once again remind the reader that we have used a seven-point overlap between neighboring blocks in this study. An overlap that contains more than seven points might help reduce the numerical errors associated with the multiblock computation. However, deeper overlaps containing more than seven points might not be practical for three-dimensional computations because such overlaps will undoubtedly require significant amount of message passing during the course of parallel computations.

Conclusions

A comparative study of single-block vs multiblock jet flow computations has been done. The results show that our implementation of the multiblock computation strategy using a seven-point overlap between neighboring blocks recovers the single-block computation reasonably well in a realistic turbulent flow.

Acknowledgments

The single-block computation presented in this Note was performed on the IBM-SP3 research computer of Indiana University, and the multiblock computation was performed on the IBM-SP4 computer at the Florida State University. It is our pleasure to acknowledge their support.

References

[1]Gatski, T. B., Hussaini, M. Y., and Lumley, J. L. (eds.), *Simulation and Modeling of Turbulent Flows*, Oxford Univ. Press, New York, 1996.

[2]Moin, P., and Mahesh, K., "Direct Numerical Simulation: A Tool in Turbulence Research," *Annual Review of Fluid Mechanics*, Vol. 30, 1998, pp. 539–578.

[3]Lele, S. K., "Compact Finite Difference Schemes with Spectral-Like Resolution," *Journal of Computational Physics*, Vol. 103, No. 1, 1992, pp. 16–42.

[4]Ashcroft, G., and Zhang, X., "Optimized Prefactored Compact Schemes," *Journal of Computational Physics*, Vol. 190, No. 2, 2003, pp. 459–457.

[5]Gaitonde, D. V., and Visbal, M. R., "Further Development of a Navier–Stokes Solution Procedure Based on Higher-Order Formulas," AIAA Paper 99-0557, 1999.

[6]Gaitonde, D. V., and Visbal, M. R., "Padé-Type Higher-Order Boundary Filters for the Navier–Stokes Equations," *AIAA Journal*, Vol. 38, No. 11, 2000, pp. 2103–2112.

[7]Visbal, M. R., and Gaitonde, D. V., "Very High-Order Spatially Implicit Schemes for Computational Acoustics on Curvilinear Meshes," *Journal of Computational Acoustics*, Vol. 9, No. 4, 2001, pp. 1259–1268.

[8]Visbal, M. R., and Rizzetta, D. P., "Large-Eddy Simulation on Curvilinear Grids Using Compact Differencing and Filtering Schemes," *Journal of Fluids Engineering*, Vol. 124, No. 4, 2002, pp. 836–847.

[9]Uzun, A., "3-D Large Eddy Simulation for Jet Aeroacoustics," Ph.D. Dissertation, School of Aeronautics and Astronautics, Purdue Univ., West Lafayette, IN, Dec. 2003.

[10]Tam, C. K. W., and Dong, Z., "Radiation and Outflow Boundary Conditions for Direct Computation of Acoustic and Flow Disturbances in a Nonuniform Mean Flow," *Journal of Computational Acoustics*, Vol. 4, No. 2, 1996, pp. 175–201.

[11]Bogey, C., and Bailly, C., "Three-Dimensional Non-Reflective Boundary Conditions for Acoustic Simulations: Far Field Formulation and Validation Test Cases," *Acta Acoustica*, Vol. 88, No. 4, 2002, pp. 463–471.

[12]Colonius, T., Lele, S. K., and Moin, P., "Boundary Conditions for Direct Computation of Aerodynamic Sound Generation," *AIAA Journal*, Vol. 31, No. 9, 1993, pp. 1574–1582.

[13]Bogey, C., Bailly, C., and Juvé, D., "Noise Investigation of a High Subsonic, Moderate Reynolds Number Jet Using a Compressible LES," *Theoretical and Computational Fluid Dynamics*, Vol. 16, No. 4, 2003, pp. 273–297.

K. Fujii
Associate Editor