AIAA JOURNAL
Vol. 46, No. 8, August 2008

Computational Framework for Coupling Compressible and Low Mach Number Codes

Yulia V. Peet* and Sanjiva K. Lele†
Stanford University, Stanford, California 94305

DOI: 10.2514/1.32884

A hybrid multicode computational method is developed that allows combining compressible and low Mach number codes into a single computational solver. The proposed methodology can be used for integrated simulations of multicomponent engineering problems. The unsteady coupling between the two codes is performed via exchanging time-dependent state information through the interfaces using overlapping meshes. The proposed coupling procedure is validated on laminar steady and unsteady test problems. Both constant-density and variable-density regimes of the low Mach number code are investigated. Results of the test cases performed in frameworks of one-way coupling and two-way coupling are documented. Several methods of implementation of interface conditions for the compressible code are compared by looking at the levels of dilatation, as well as numerical errors. The injection method (when all the variables are simply transformed from one code to another through an interpolation procedure) is chosen for its superior stability, accuracy, and simplicity of implementation. Numerical results for the coupled calculations obtained with the injection method show good agreement with both standalone calculations and analytical solutions.

Nomenclature

A	=	area of the computational domain
c	=	speed of sound
c_p	=	specific heat at constant pressure
c_v	=	specific heat at constant volume
k	=	thermal diffusivity
L	=	length, error norm
M	=	Mach number
$\mathbf{n}$	=	unit normal vector
Pr	=	Prandtl number
p	=	pressure
Q	=	heat conduction and heat release rate
Re	=	Reynolds number
r	=	distance from the center of the disturbance
S_{ij}	=	strain rate tensor
T	=	temperature
t	=	time
$\mathbf{u}$	=	velocity vector
(u, v, w)	=	velocity components in a Cartesian coordinate system
(u_x, u_r, u_θ)	=	velocity components in a cylindrical coordinate system
(u_1, u_2, u_3)	=	velocity components in a rotated Cartesian coordinate system
V	=	volume of the computational domain
$\mathbf{x}$	=	coordinate vector
(x, r, θ)	=	cylindrical coordinate system
(x, y, z)	=	Cartesian coordinate system
(x_1, x_2, x_3)	=	rotated Cartesian coordinate system
α	=	angle between the computational domains of the two codes
Δt	=	time-step size
Δx	=	grid size
γ	=	ratio of specific heats
δ	=	boundary-layer thickness
ϵ_2, ϵ_3	=	residual terms in low Mach number expansions
μ	=	dynamic viscosity
ν	=	kinematic viscosity
ρ	=	density

Subscripts

d	=	disturbance
i, j, k	=	indexes of the coordinates
n	=	direction normal to the boundary
r	=	reference
w	=	wall
∞	=	freestream

Superscripts

c	=	compressible
lm	=	low Mach number
(0), (1), (2)	=	order with Mach number
$*$	=	dimensional
$'$	=	fluctuation

I. Introduction

WITH the advance in performance and capabilities of modern computers, the drive toward large-scale integrated simulations of complex flow systems is growing. Examples of such integrated simulations include prediction of an aerothermal flow through an entire gas turbine engine [1,2], computations of helicopter blade-wake/blade-vortex interactions [3,4], simulation of film cooling of the turbine blades [5], etc. The different flow physics in each individual component of these systems (compressor and turbine versus combustor in the jet engine, near-blade region versus wake region in the helicopter simulations, and plenum and film holes versus blade exterior in the cooling systems) calls for the coupling of different solvers, such as compressible and low Mach number codes, large eddy simulation with Reynolds-averaged Navier–Stokes methods, unsteady Reynolds-averaged Navier–Stokes with Reynolds-averaged Navier–Stokes methods, etc. This area of multicode multiphysics coupling is emerging and has not yet received much attention in the literature. With the hybrid multicode approach, each part of the simulated phenomena, usually corresponding to a separate

Received 20 June 2007; revision received 18 March 2008; accepted for publication 2 April 2008.

*Ph.D. Student, Department of Aeronautics and Astronautics; currently Postdoctoral Research Fellow, Université Pierre et Marie Curie, Paris 6, Paris 75005, France. Member AIAA.

†Professor, Department of Aeronautics and Astronautics. Member AIAA.

geometrical unit, is treated with its own numerical code that is best suited for describing its physics. Different numerical codes are combined by an unsteady coupling procedure, allowing for an exchange of time-dependent state information through the interfaces.

Geometrical complexity of many industrial applications tackled by computational fluid dynamics codes promoted a development of a multidomain approach, or domain decomposition method, in the 1980s. Difficulty in creating a single computational grid with a high resolution in all the critical places led to splitting a computational domain into separate blocks and combining them through interface conditions. There are two different approaches to the domain decomposition method. In the first approach, the overlapping or overset grids are used and some interpolation scheme is implemented to transform the interface values from one grid to another [6–8]. Another approach uses nonoverlapping computational grids, where the location of grid points is exactly matched at the interfaces [9–11]. It is argued by the advocates of the matching grids that in this case, special difference schemes satisfying summation by parts property can be used, resulting in a cancellation of errors at the interfaces and assuring the strict stability of the method [9,10]. However, in complex geometries, it is not always possible to ensure a perfect matching of the component grids, and use of overlapping meshes is often inevitable.

However, the difference between the domain decomposition method and the current multicode approach is that the set of equations solved in each subdomain in the former method is exactly the same. Berger [6,7], Chesshire and Henshaw [8], and Carpenter et al. [9] looked at hyperbolic partial differential equations; Pfeiffer et al. [12] and Chang and Chien [13] studied elliptic problems. The domain decomposition method has been applied to the computation of compressible flows by Kopriva [14], Hesthaven [15], and Nördstrom and Carpenter [16]. Incompressible flows were treated by Manna et al. [11] and Strikwerda and Scarbnick [17]. In the present methodology, a set of partial differential equations is different in different subdomains: namely, compressible versus low Mach number equations. Mathematical grounds developed for the coupling between numerical solutions of identical equation sets are not easily extendable to the situation with different equation sets [18]. The situation is complicated by the fact that a different number of equations in each system results in a different rank of the discrete operators for each equation set.

Although a theoretical background for coupling between compressible and low Mach number codes is yet to be available, the present paper proposes a computational methodology for such a coupling and documents numerical experiments performed with this method on a wide range of steady and unsteady laminar problems. Both constant- and variable-density regimes of the low Mach number code are investigated. In Sec. II, the numerics of individual codes to be coupled is briefly outlined. Coupling procedure and details of the interface condition formulation are described in Sec. III. Validation of the coupling procedure starts with the test cases performed in a one-way coupling framework, followed by the two-way coupling tests in Sec. IV. Conclusions are stated in Sec. V. The proposed procedure has been further applied to large eddy simulations of film-cooling flow for a realistic film-cooling configuration, which is documented in [5].

II. Description of Individual Codes

A. Compressible Code

The compressible code used in the present multicode solver was developed by Xiong [19]. Compressible Navier–Stokes equations written in primitive variables are solved numerically:

$$\rho_{,t} + (\rho u_i)_{,i} = 0 \tag{1}$$

$$\rho u_{i,t} + \rho u_j u_{i,j} = -p_{,i} + 1/Re[(2\mu S_{ij})_{,j} - (2/3\mu u_{j,j})_{,i}] \tag{2}$$

$$\rho T_{,t} + \rho u_j T_{,j} + (\gamma - 1)\rho T u_{j,j} = \gamma/(PrRe)[kT_{,i}]_{,i} + \gamma(\gamma - 1)M^2/Re[2\mu S_{ij}S_{ij} - 2/3\mu S_{ii}S_{jj}] \tag{3}$$

$$p = \rho T/(\gamma M^2) \tag{4}$$

The preceding equations are nondimensionalized with the relations

$$\rho = \rho^*/\rho_r^*, \qquad u_i = u_i^*/u_r^*, \qquad T = T^*/T_r^*$$
$$p = p^*/(\rho_r^* u_r^{*2}), \qquad \mu = \mu^*/\mu_r^*, \qquad k = k^*/k_r^* \tag{5}$$
$$x_i = x_i^*/L_r^*, \qquad t = t^* u_r^*/L_r^*$$

where superscript $*$ refers to the dimensional quantities, and subscript r denotes the reference variables. Variables without an $*$ represent nondimensional variables throughout this paper. Nondimensional parameters are introduced as follows:

$$M = \frac{u_r^*}{c_r^*}, \qquad Re = \frac{\rho_r^* u_r^* L_r^*}{\mu_r^*}, \qquad \mathrm{Pr} = \frac{\mu_r^* c_{pr}^*}{k_r^*} \tag{6}$$

The spatial discretization of Eqs. (1–4) is achieved by mapping the body-fitted mesh coordinates from physical space to the uniform computational space. A fourth-order-accurate central-difference scheme is used for first and second derivatives in the computational space. A fully implicit, approximately factorized, second-order-accurate time-marching scheme is used to advance the governing equations (1–4) in time. A linearized dual-time-stepping subiteration scheme is employed to accelerate the convergence of subiterations. With a fourth-order discretization scheme and implicit treatment, block pentadiagonal matrix arise with additional blocks at the boundary nodes to account for the high-order boundary treatment. The resulting coefficient matrix can be inverted efficiently using the direct Gaussian elimination method. Further details of the numerical implementation of the compressible code can be found in [19].

B. Low Mach Number Code

The low Mach number code used in the multicode solver was written by Pierce [20]. The set of equations solved numerically is the low Mach number approximation [21] of the compressible Navier–Stokes equations written in conservative form:

$$\rho_{,t} + (\rho u_i)_{,i} = 0 \tag{7}$$

$$(\rho u_i)_{,t} + (\rho u_j u_i)_{,j} = -p_{,i}^{(2)} + 1/Re[(2\mu S_{ij})_{,j} - (2/3\mu u_{j,j})_{,i}] \tag{8}$$

$$(\rho T)_{,t} + (\rho u_j T)_{,j} = 1/(PrRe)[kT_{,i}]_{,i} \tag{9}$$

$$p^{(0)} = \rho T \tag{10}$$

To obtain low Mach number approximation, all nondimensional flow variables in the Navier–Stokes equations were expanded in the power series of Mach number M, and the coefficients of the monomials $M^l, l = 0, 1, 2, \ldots$ were set to zero. Equations corresponding to the zeroth-order approximation $l = 0$ are Eqs. (7–10), or low Mach number equations. For all the variables except for the pressure, only zeroth-order terms enter the low Mach number equations (7–10). For those variables, the superscript denoting the order of the variable is dropped for brevity (variables without a superscript are zeroth order by default). In the derivation of Eqs. (7–10), additional assumptions were made that the total heat conduction and heat release rate

$$\int_V Q^{(0)}\,\mathrm{d}V$$

as well as the total volume flow

$$\int_{\partial V} \mathbf{u}^{(0)} \cdot \mathbf{n}\, \mathrm{d}A$$

are zero in the computational domain

$$\int_V Q^{(0)}\, \mathrm{d}V = 0 \qquad \int_{\partial V} \mathbf{u}^{(0)} \cdot \mathbf{n}\, \mathrm{d}A = 0 \tag{11}$$

Conditions (11) lead to the nullification of a time derivative of a zeroth-order pressure,

$$\frac{\mathrm{d}p^{(0)}}{\mathrm{d}t} = 0 \tag{12}$$

otherwise present in the low Mach number equations [22].

Nondimensionalization of Eqs. (7–10) is performed using the same relations (5) as in the compressible formulation, except for the nondimensionalization of pressure. Indeed, if pressure is normalized with $\rho_r^* u_r^{*2}$, as in relations (5), the nondimensional pressure determined by the equation of state becomes infinite if the Mach number is zero [see Eq. (4)]. To remove the singularity in the nondimensional pressure in the zero Mach number limit, different normalization for pressure is employed:

$$p = p^*/p_r^* \tag{13}$$

where p_r^* is a reference pressure. Freestream conditions are chosen to provide the reference values in relations (5) and (13).

For numerical discretization of Eqs. (7–10), velocity components are staggered with respect to the density and other scalars in both space and time. A second-order central-difference scheme is used for the integration of momentum equations, whereas the quadratic upstream interpolation for convective kinematics (QUICK) scheme is employed for scalar advection to avoid the formation of spatial oscillations. A semi-implicit second-order scheme, similar to the Crank–Nicolson or trapezoidal scheme, with Newton–Raphson subiterations is used for time advancement. Advection and diffusion terms in the radial and azimuthal directions and pressure in all directions are treated implicitly, whereas all the other terms are treated explicitly. To advance the momentum equations, the fractional-step method is used. The Poisson equation for pressure is solved with the multigrid method. In three-dimensional formulation, Eqs. (7–10) are cast in cylindrical coordinates for simulating coaxial combustors [20] and film-cooling supply holes [5]. For additional details of the numerical method, the reader is referred to [20].

Both compressible and low Mach number codes are written in large eddy simulation formulation with the dynamic Smagorinsky eddy-viscosity model [23] used for treatment of subgrid-scale terms.

III. Coupling Procedure

Consider a situation in which a low Mach number code domain and a compressible code domain overlap, as in Fig. 1. The low Mach number code domain is located at the left and the compressible code domain is located at the right in Fig. 1. The local coordinate systems used in each code are also shown. The low Mach number code uses a cylindrical coordinate system (x, r, θ), whereas the compressible code uses the Cartesian coordinate system (x, y, z). Assume that boundary conditions at the boundaries of each computational domain not contained within another code domain, shown as solid lines in Fig. 1, are known. The coupling procedure therefore consists of specifying boundary conditions in each code at the boundaries contained within another code domain, shown as dashed lines in Fig. 1, which will be referred to hereafter as interface boundaries, or interfaces. To ensure that unsteady features of the simulated phenomena are retained during coupling, exchange of state information through the interfaces is performed at every time step. This exchange takes place at the beginning of a time step, after which each code performs inner iterations with the state information at the interfaces used as boundary conditions. The numerical solution inside the region of overlap encompassed by the interface boundaries is obtained by both codes.

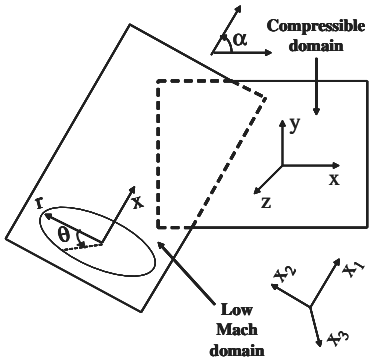


Fig. 1 Schematic of an overlap of the computational domains during coupling.

We now describe the interface conditions specified at the interface boundaries in each code. Hereafter, the superscript c is used to denote the variables in the compressible code and the superscript lm refers to the variables in the low Mach number code. This particular choice of the interface conditions is validated in Sec. IV, where comparison with other methods is performed with application to test problems.

A. Interface Conditions for the Compressible Code

Variables obtained numerically in the compressible code are $\{\rho, u, v, w, T\}$. Specification of these five quantities $\{\rho^c, u^c, v^c, w^c, T^c\}$ is required at the grid nodes corresponding to the interface boundaries of the compressible code. One can interpolate the low Mach number code solution to the grid nodes of the interface boundary to obtain the values $\{\rho^{\mathrm{lm}}, u^{\mathrm{lm}}, v^{\mathrm{lm}}, w^{\mathrm{lm}}, T^{\mathrm{lm}}\}$ at these nodes. The question is how to reconstruct the compressible values from the low Mach number values. Although several methods of reconstruction were considered and tested, as described in Sec. IV, the one that shows the best performance in terms of accuracy and stability is also the simplest. It consists of transforming all five variables from the low Mach number code to the compressible code, which is referred to as injection in the present paper:

$$u^c = u^{\mathrm{lm}}, \qquad v^c = v^{\mathrm{lm}}, \qquad w^c = w^{\mathrm{lm}}$$
$$\rho^c = \rho^{\mathrm{lm}}, \qquad T^c = T^{\mathrm{lm}} \tag{14}$$

Note that the injection method is overposed if used as a boundary condition for the compressible code in a subsonic case. It is, however, physically justified when used as an interface condition, because the interface is an interior region of the fully coupled solution and the full transfer of variables from one computational block to another is possible.

B. Interface Conditions for the Low Mach Number Code

1. Velocity and Temperature

In the low Mach number code, interface conditions are required at the interface boundaries for the three components of velocity (u_x, u_r, u_θ), temperature T, and a second-order pressure $p^{(2)}$ or its gradients to solve the Poisson equation for pressure. Density is obtained from the temperature through the equation of state and therefore no interface condition for density is required.

In the present method, we use injection for velocity components, setting the velocity values at the interface in the low Mach number code equal to the corresponding values interpolated from the compressible code:

$$u_x^{\mathrm{lm}} = u_x^c, \qquad u_r^{\mathrm{lm}} = u_r^c, \qquad u_\theta^{\mathrm{lm}} = u_\theta^c \tag{15}$$

For the temperature, the following interface condition is employed:

$$\begin{cases} T^{\mathrm{lm}} = T^c & \text{if } u_n^{\mathrm{lm}} \le 0 \\ T_{,t}^{\mathrm{lm}} + u_{\mathrm{conv}}^{\mathrm{lm}} T_{,n}^{\mathrm{lm}} = 0 & \text{if } u_n^{\mathrm{lm}} > 0 \end{cases} \tag{16}$$

where u_n^{lm} is velocity normal to the boundary. In other words, temperature injection is used for the part of the interface boundary corresponding to an inflow ($u_n^{\mathrm{lm}} \le 0$), and a convective outflow condition for the temperature is used for the part of the interface boundary corresponding to an outflow ($u_n^{\mathrm{lm}} > 0$). It is necessary to use the convective condition for the temperature at the outflow to prevent the formation of wiggles in the temperature field, which would form in case of injection due to an upwindlike QUICK approximation employed for the scalar advection. Here, $u_{\mathrm{conv}}^{\mathrm{lm}}$ is a convection velocity defined as the maximum outflow velocity over the boundary:

$$u_{\mathrm{conv}}^{\mathrm{lm}} = \max_{u_n^{\mathrm{lm}}>0} u_n^{\mathrm{lm}} \tag{17}$$

According to Eq. (17), $u_{\mathrm{conv}}^{\mathrm{lm}}$ is constant over the interface boundary in the present formulation. The regions of inflow and outflow are located dynamically every time step by looking at u_n^{lm}.

2. Pressure

Because of the different roles played by pressure in compressible and low Mach number equations, formulating an interface condition for pressure requires special care. Because of different normalization of pressure employed in the two codes, one has to refer back to dimensional variables to derive an interface condition. Dimensional pressure obtained by the compressible code is

$$p^{*c}(\mathbf{x}, t) = p^c(\mathbf{x}, t) \cdot \rho_r^* u_r^{*2} \tag{18}$$

An equivalent dimensional pressure in the low Mach number formulation is

$$p^{*\mathrm{lm}}(\mathbf{x}, t) = p^{\mathrm{lm}}(\mathbf{x}, t) \cdot p_r^* \tag{19}$$

where $p^{\mathrm{lm}}(\mathbf{x}, t)$ has an asymptotic expansion

$$p^{\mathrm{lm}}(\mathbf{x}, t) = p^{(0)\mathrm{lm}}(t) + \gamma M^2 p^{(2)\mathrm{lm}}(\mathbf{x}, t) + \epsilon_3(\mathbf{x}, t) \tag{20}$$

according to the low Mach number asymptotic analysis [22]. Here, $\epsilon_3(\mathbf{x}, t)$ represents the residual terms in the expansion, $\epsilon_3(\mathbf{x}, t) \sim \mathcal{O}(M^3)$. In a general case, zeroth-order pressure $p^{(0)\mathrm{lm}}(t)$ is a function of time, but constant in space. It plays the role of a global thermodynamic pressure. According to Eq. (12), the time dependence of a zeroth-order pressure is dropped out in the present formulation due to assumptions (11). Second-order pressure $p^{(2)\mathrm{lm}}(\mathbf{x}, t)$, or dynamic pressure, is decoupled from the density and temperature. Its role is similar to the pressure in an incompressible flow: the divergence of the momentum equation (8) yields a Poisson equation for the second-order pressure.

Because dimensional values of pressure p^{*c} and $p^{*\mathrm{lm}}$ should be equal at the interface, equating the right-hand sides of Eq. (18) and Eq. (19) and taking into account that $p_r^* = \rho_r^* u_r^{*2}/(\gamma M^2)$, one obtains at the interface

$$\gamma M^2 p^c(\mathbf{x}, t) = p^{(0)\mathrm{lm}}(t) + \gamma M^2 p^{(2)\mathrm{lm}}(\mathbf{x}, t) + \epsilon_3(\mathbf{x}, t) \tag{21}$$

In the present method, the zeroth-order pressure is set equal to the reference pressure

$$p^{*(0)\mathrm{lm}} = p_r^* \tag{22}$$

leading to the condition

$$p^{(0)\mathrm{lm}} = 1 \tag{23}$$

in nondimensional variables. For the second-order pressure $p^{(2)\mathrm{lm}}(\mathbf{x}, t)$, interface conditions are required for solving the Poisson equation. In the present formulation, the Poisson equation is solved with Neumann boundary conditions. Taking spatial derivatives of Eq. (21) and neglecting the third-order terms, the following approximate Neumann interface conditions can be derived:

$$\partial p^{(2)\mathrm{lm}}(\mathbf{x}, t)/\partial x_n = \partial p^c(\mathbf{x}, t)/\partial x_n \tag{24}$$

Conditions (23) and (24) are sufficient to obtain numerical solution of the low Mach number equations (7–10).

3. Total Value of Pressure

If one wishes to compare the values of pressure obtained by the two codes or to use the pressure from the low Mach number code in the formulation of interface conditions for the compressible code, one needs to estimate the total value of pressure $p^{\mathrm{lm}}(\mathbf{x}, t)$ at the desired grid locations of the low Mach number domain. To do that, first note that with Neumann boundary conditions, the second-order pressure $p^{(2)\mathrm{lm}}(\mathbf{x}, t)$ can be calculated only up to an additive function $C(t)$, constant in space:

$$p^{(2)\mathrm{lm}}(\mathbf{x}, t) = \tilde{p}^{(2)\mathrm{lm}}(\mathbf{x}, t) + C(t) \tag{25}$$

where $\tilde{p}^{(2)\mathrm{lm}}(\mathbf{x}, t)$ is a partial solution of the Poisson equation. Therefore, according to Eq. (20), $p^{\mathrm{lm}}(\mathbf{x}, t)$ can be written as

$$p^{\mathrm{lm}}(\mathbf{x}, t) = p^{(0)\mathrm{lm}}(t) + \gamma M^2 \tilde{p}^{(2)\mathrm{lm}}(\mathbf{x}, t) + \epsilon_2(\mathbf{x}, t) \tag{26}$$

where the function $\epsilon_2(\mathbf{x}, t)$ is defined as

$$\epsilon_2(\mathbf{x}, t) = \gamma M^2 C(t) + \epsilon_3(\mathbf{x}, t) \tag{27}$$

Note that at order M^2 the function $\epsilon_2(\mathbf{x}, t)$ depends only on time and not on space. To construct $p^{\mathrm{lm}}(\mathbf{x}, t)$, one needs to know $\epsilon_2(\mathbf{x}, t)$, which is not part of a solution of the low Mach number equations (7–10); $\epsilon_2(\mathbf{x}, t)$ can be calculated in the region of overlap by using the pressure from the compressible code to account for the missing information. However, evaluation of $\epsilon_2(\mathbf{x}, t)$ at every grid point in the overlap region during each computational time step is prohibitively expensive, because three-dimensional interpolation is required. A simpler method for reconstructing $\epsilon_2(\mathbf{x}, t)$ is proposed: $\epsilon_2(\mathbf{x}, t)$ is evaluated only at one specific location inside the overlap region, some reference point $\mathbf{x}_0$. The same value of $\epsilon_2(\mathbf{x}, t)$ is assumed for the rest of the low Mach number domain; that is,

$$\epsilon_2(\mathbf{x}, t) = \epsilon_2(\mathbf{x}_0, t) \tag{28}$$

By neglecting spatial variation of ϵ_2 we still retain the second order of accuracy with the Mach number, as already noted. With this procedure for reconstructing ϵ_2, one can derive the following expression for p^{lm}:

$$p^{\mathrm{lm}}(\mathbf{x}, t) = \gamma M^2 [p^c(\mathbf{x}_0, t) + \tilde{p}^{(2)\mathrm{lm}}(\mathbf{x}, t) - \tilde{p}^{(2)\mathrm{lm}}(\mathbf{x}_0, t)] \tag{29}$$

It is important not to forget to change normalization if the low Mach number pressure is supplied to the compressible code for formulating interface conditions:

$$p^c = p^{\mathrm{lm}} \cdot p_r^* / \left(\rho_r^* u_r^{*2}\right) = p^{\mathrm{lm}}/(\gamma M^2) \tag{30}$$

C. Error With Mach Number

Because low Mach number asymptotic expansions retain the terms up to the second order with Mach number, $\mathcal{O}(M^2)$, the inherent difference between the analytical solutions of the compressible and the low Mach number equations is of that order. Therefore, numerical solutions of the compressible and low Mach number codes are expected to have at least the same level of discrepancy. The preceding formulated interface conditions do not introduce any additional error with Mach number. Indeed, interface conditions (14) and (15) are valid to $\mathcal{O}(M^2)$ according to the difference between compressible and low Mach number variables due to the low Mach number asymptotic expansions. In the formulation of condition (24) for the pressure gradient, only third-order terms are omitted; neglecting a spatial variation of $\epsilon_2(\mathbf{x}, t)$ in Eq. (28) does not introduce any additional error with Mach number, because spatial variation of $\epsilon_2(\mathbf{x}, t)$ comes from the third-order function $\epsilon_3(\mathbf{x}, t)$ [see Eq. (27)]. Therefore, the same order of approximation, $\mathcal{O}(M^2)$, is retained in the formulation of interface conditions as in the low Mach number asymptotics.

D. Interpolation

To reconstruct the state information to be supplied at the interfaces for formulating conditions (14–16) and (24), one needs to interpolate the variables from one domain to the interface boundary nodes of another domain. In the present method, trilinear interpolation is used. It was shown in [5] that increasing an order of interpolation does not lead to an increase of accuracy of the coupled solution for a couple of reasons. First, the low Mach number code is only second-order accurate in space and a second order of interpolation is consistent with its accuracy. Second, model error $\mathcal{O}(M^2)$ caused by the difference of the solutions of compressible and low Mach number equations usually dominates the error due to interpolation, because the latter can be controlled by grid refinement, but the former is fixed with Mach number. Interpolation procedure used in the present method is not conservative; that is, it does not guarantee the conservation of mass, momentum, and energy fluxes across the multicode interface in a discrete sense. Two different sources contribute to the mismatch in flux values across the interface: a mismatch in the corresponding discrete quantities (mass, momentum, or energy) due to the difference in model equations (compressible or low Mach number) and a mismatch due to an interpolation and discretization causing the integral values of these quantities over the interface to be different even if the mismatch due to the model was absent. An overall error introduced by the lack of conservation is therefore no larger than other model, interpolation, or discretization errors.

1. *Interpolation from the Low Mach Number to the Compressible Code*

First consider interpolation from the low Mach number code domain to the interface boundary of the compressible code domain. An intersection of the interface boundary of the compressible code with the staggered grid corresponding to the low Mach number code is shown in Fig. 2 in two dimensions. Grid nodes of the interface boundary, to which low Mach number values should be interpolated, are specified as black circles. To perform interpolation, the grid cell of the low Mach number code surrounding each interface boundary node of the compressible code is identified at the preprocessing step. Because of the staggering of the low Mach number variables, the corresponding grid cells will be different for an interpolation of different variables. For example, to interpolate u values to the third interface boundary point from the left in Fig. 2 (third circle from the left), the grid cell with the corners at u velocity locations is used. To interpolate the values of v, the grid cell with v velocity in the corners is used, etc. This way, all five variables that constitute the state information for formulating interface conditions (14) (density, temperature, and the three components of velocity) are obtained at each interface boundary point of the compressible code. However, interpolated velocities correspond to the low Mach number code local coordinate system. For a three-dimensional case, the coordinate system is cylindrical (x, r, θ) and the corresponding interpolated velocities are (u_x, u_r, u_θ). Transformation of these velocities into the coordinate system associated with the compressible code (x, y, z) is performed in two steps, which are schematically shown in Fig. 3a. After interpolation to the nodes of the compressible interface

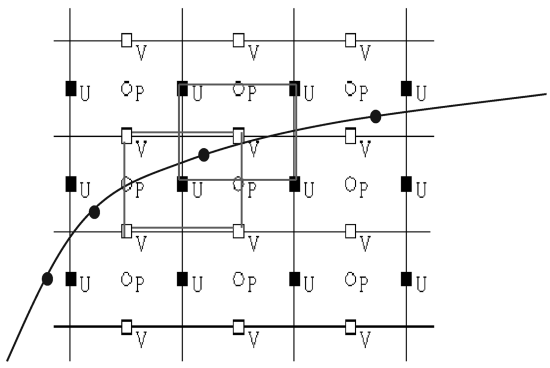


Fig. 2 Intersection of the compressible interface boundary with the staggered grid of the low Mach number code.

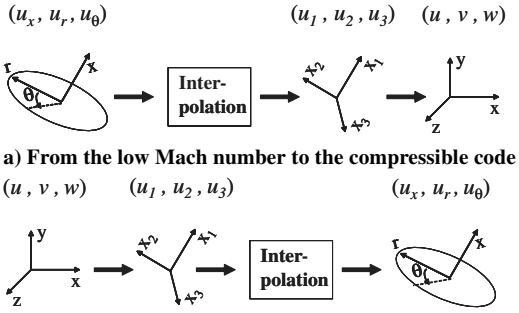


Fig. 3 Transformation of velocities between the coordinate systems of the two codes.

boundary, velocities (u_x, u_r, u_θ) are first transformed into velocities (u_1, u_2, u_3), which correspond to the local Cartesian system associated with the low Mach number code, (x_1, x_2, x_3), and then into velocities (u, v, w). The (x_1, x_2, x_3) coordinate system is obtained by the rotation of the system (x, y, z) counterclockwise to an angle α, where α is the angle between the x axis of the compressible domain and the centerline of the low Mach number domain (see Fig. 1).

2. *Interpolation from the Compressible to the Low Mach Number Code*

To interpolate the variables from the compressible domain to the interface boundaries of the low Mach number domain, the procedure is a little different. The difference comes from the different arrangement of variables with respect to the computational grid in the two codes. In the compressible code, the variables are collocated, and an interpolation of velocities from the low Mach number code to the grid nodes of the compressible interface boundary is performed before the transformations. In the low Mach number code, the variables are staggered. Because of the need for the transformation of velocities interpolated from the compressible code into the local coordinate system associated with the low Mach number code, $(u, v, w) \mapsto (u_x, u_r, u_\theta)$, to reconstruct each of the variables u_x, u_r, and u_θ, knowledge of all three components of velocity u, v, and w would be required at each location on the staggered grid, where u_x, u_r, and u_θ are defined. It means that an amount of interpolation operations would be 3 times larger than in the case of interpolation from the low Mach number to the compressible code described earlier. To reduce the amount of interpolation, which is an expensive procedure, especially in three dimensions, transformation $(u, v, w) \mapsto (u_1, u_2, u_3)$ is performed before interpolation at all the grid points of the compressible domain. By doing the transformation $(u, v, w) \mapsto (u_1, u_2, u_3)$ first, we only need to interpolate one velocity component, u_1, to obtain streamwise velocity u_x and two velocity components u_2 and u_3 to find the values of radial u_r and azimuthal u_θ velocities. The schematics of the procedure is sketched in Fig. 3b.

3. *Parallelism*

The parallel interface for coupling the codes is written on a message-passing interface platform. It constructs disjoint groups of processes: one group for each code. All communications within each group are performed using an intracommunicator. Message passing between the different groups is accomplished with the help of an intercommunicator. Time advancement of the two codes is synchronized by choosing the global time step equal to the smallest among the time steps of individual codes, as dictated by the stability requirement:

$$\Delta t = \min(\Delta t^c, \Delta t^{\mathrm{lm}}) \tag{31}$$

IV. Validation

Numerical test cases validating the coupling procedure are presented in this section. Two groups of test cases are considered: one-way coupled and two-way coupled. In the one-way coupling setting, only one code obtains information from the other code. The

other code operates in a standalone mode, without feeling any feedback. One-way coupling tests are necessary to check whether the numerical solutions of the two codes are close to each other when the Mach number is low (solutions are supposed to have a difference of the order of M^2, according to the accuracy of the low Mach number approximation). This check is important, because the solution in the region of overlap in Fig. 1 is computed by both codes, and the closeness of these solutions is necessary for the convergence of the coupled solution. It also implies that the Mach number of the physical problem in the region of overlap should be small; that is, the region of overlap should be placed inside the low-speed areas of the flow, such as boundary-layers, etc. In a two-way coupling mode, both codes exchange the state information through the interfaces. In this setting, the stability of the coupling procedure, as well as the accuracy of the coupled solution, can be investigated. In the present paper, both constant- and variable-density implementation of the coupling procedure is tested, because in some cases it is convenient to use a low Mach number code in an incompressible regime, and many computational codes are purely incompressible. Although the coupling procedure was described for the variable-density regime of the low Mach number code, its extension to the constant density, or incompressible regime, is straightforward. A list of implemented one-way and two-way coupling test cases for both incompressible and variable-density regimes is given in Table 1. The results of the test cases are described next.

A. One-Way Coupling

In the one-way coupling formulation, the low Mach number code domain is placed inside the compressible code domain and information is transferred from the compressible to the low Mach number code through all the boundaries of the low Mach number code domain. In the present setting, the compressible code domain represents a region exterior to the upper half of the model turbine blade with a cylindrical leading edge, and the low Mach number code domain is placed about eight diameters downstream of the leading edge, as can be seen in Fig. 4. The size, vertical location, and orientation of the low Mach number code domain depend on the test case considered, whereas the geometry of the compressible code domain is fixed for all the tests. Further details of geometry and computational meshes used in the one-way coupling setting can be found in [24]. The Mach number for the compressible code is $M = 0.05$.

1. *Incompressible–Compressible Regime*

For coupling tests, when the low Mach number code is run in an incompressible regime, velocity disturbance in the form of a circular Taylor vortex [25] is superimposed on the mean flow around the model turbine blade. For the circular Taylor vortex, only tangential (swirl) velocity is nonzero, denoted here as u':

$$u' = \frac{\mathcal{M} r}{16\pi\nu^2 t^2}\exp\left(\frac{-r^2}{4\nu t}\right) \tag{32}$$

where r is the distance from the center of the vortex. The circular Taylor vortex is an analytical solution of unsteady viscous incompressible equations and

$$\mathcal{M} = \int_0^\infty 2\pi r u' r\,\mathrm{d}r$$

is an invariant of the flow. Parameters t and $\mathcal{M}$ of Eq. (32) are chosen to set the initial radius of the vortex r_d (distance from the vortex center to the point of maximum velocity) so that $Re = U_\infty r_d/\nu_\infty = 330$, and the initial velocity disturbance level $u'_{\max}/U_\infty = 1\%$. The vortex is initialized in the compressible code domain upstream of the low Mach number code domain. It is convected with the mean flow and captured by the low Mach number code as it passes through it.

Table 1 List of numerical tests to validate the coupling procedure

Coupling regime	One-way coupling	Two-way coupling
Incompressible–compressible	Taylor vortex	Taylor vortex
Variable-density–compressible	Heated boundary-layer	Hot spot

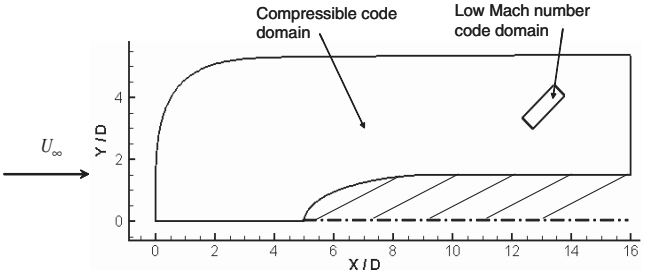


Fig. 4 Schematic of the computational domain for one-way coupling tests.

Two-dimensional and three-dimensional tests are conducted. In two dimensions, the low Mach number code domain is rectangular, with 128×32 mesh points distributed uniformly. Two cases are considered: when the x axis of the low Mach number domain is perpendicular and inclined at 45 deg to the direction of the main stream. For a three-dimensional configuration, the low Mach number domain is cylindrical, with uniform mesh of $128 \times 32 \times 64$ points and the symmetry axis perpendicular to the main stream. In a three-dimensional formulation, the flow is quasi-two-dimensional; that is, there is no variation of the flow parameters in a spanwise z direction in the coordinate system associated with the compressible code.

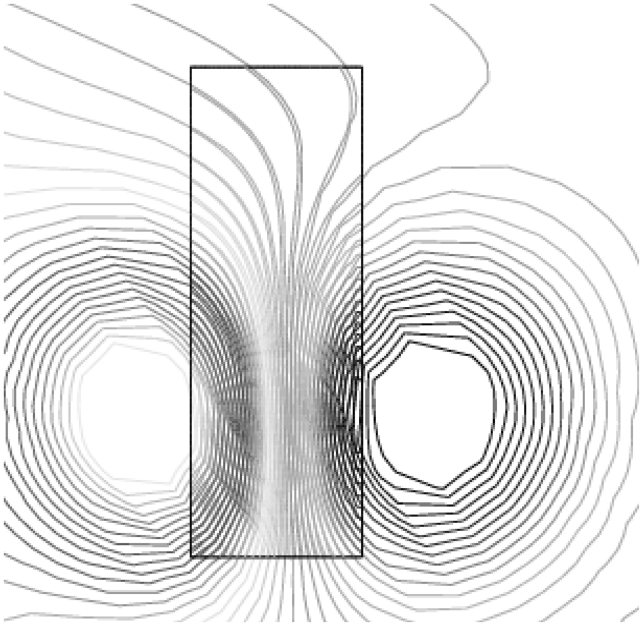
Typical snapshots of the vertical velocity v calculated with the two codes are overlaid in Fig. 5 for perpendicular and inclined domains in the two-dimensional case. The moment when the center of the vortex is inside the low Mach number code domain is shown. It can be seen that the contours of the vertical velocity of the two solutions are very close to each other. The same was observed for all other variables. In fact, contours of vertical velocity show the largest difference. A slight asymmetry in velocity contours in Fig. 5 is due to the mean flow nonuniformity as it bends around the blade leading edge. The mean flow nonuniformity, although it exists, is not very significant at eight diameters downstream of the leading edge, where the low Mach number code domain is located.

The maximum difference between the flow variables obtained by the two codes is calculated over the low Mach number code domain as $\Delta u_i = \max|u_i^c - u_i^{\mathrm{lm}}|$ for the velocity components and $\Delta p = \max|p^c - p^{\mathrm{lm}}|$ for the pressure. The difference in velocity normalized by the maximum initial swirl velocity, $\Delta u_i/u'_{\max}$, is plotted versus nondimensional computational time tU_∞/r_d in Fig. 6 for both two-dimensional and three-dimensional cases. The difference in pressure, normalized as $\Delta p/(\rho_\infty u'^2_{\max})$, is plotted in Fig. 7. Both velocity and pressure differences are larger when the center of the vortex is inside the low Mach number code domain, corresponding to the computational time $tU_\infty/r_d \sim 2-4$. From Figs. 6a and 7 one can see that the orientation of the low Mach number domain exerts little influence on the maximum difference between the solutions.

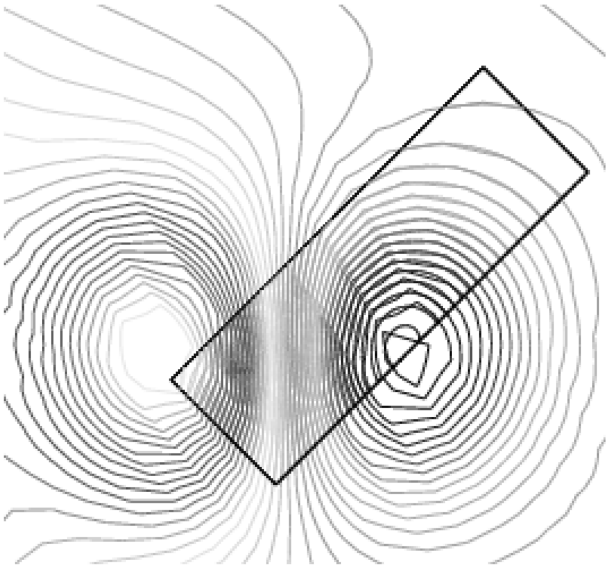
It can be noticed that the maximum difference between the solutions is about 20% of the 1% disturbance level, which constitutes 0.2% of the freestream velocity. A maximum discrepancy of 0.2% (0.002) is comparable with the value of $M^2 = 0.0025$. As already discussed, error of the order of M^2 is an inherent error between the solution of the low Mach number equations and the solution of the compressible equations, due to the approximations made in the low Mach number asymptotic analysis. This level of discrepancy is expected between the numerical solutions of the two codes.

2. *Variable-Density–Compressible Regime*

The laminar boundary layer above the wall heated to the temperature $T_w = 2T_\infty$ is chosen as a test case for the one-way

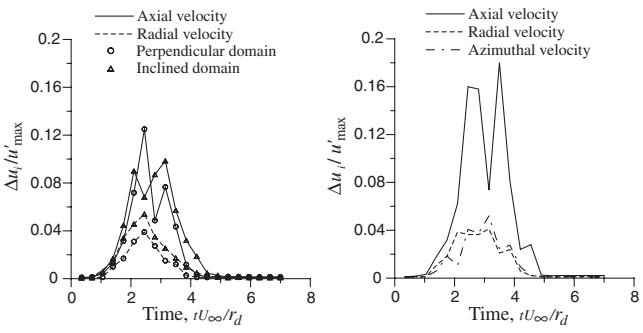


a) Perpendicular domain



b) Inclined domain

Fig. 5 Vertical velocity contours for the convecting Taylor vortex (two-dimensional case).



a) Two-dimensional case b) Three-dimensional case

Fig. 6 Maximum difference in velocity normalized by the maximum initial swirl velocity, $\Delta u_i/u'_{max}$, versus nondimensional computational time tU_∞/r_d for a convecting Taylor vortex.

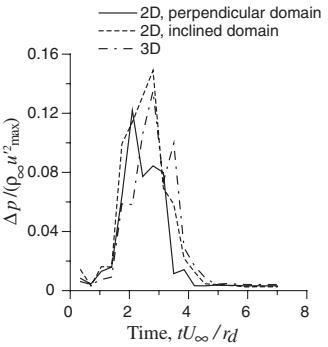


Fig. 7 Maximum difference in pressure, $\Delta p/(\rho_\infty u'^2_{max})$, versus nondimensional computational time tU_∞/r_d for a convecting Taylor vortex.

coupling, when the low Mach number code operates in a variable-density regime. The Reynolds number is $Re_\delta = U_\infty \delta/\nu_\infty = 500$ (δ is 99% boundary-layer thickness) and the Mach number is $M = 0.05$. Numerical solutions of the compressible and the low Mach number code are compared. Wall-normal profiles of the streamwise and vertical velocities and the temperature taken through the point with the maximum discrepancy between the solutions are shown in Fig. 8. It can be seen that the agreement between the two solutions is rather good.

B. Two-Way Coupling

For two-way coupling, the low Mach number code domain is located at the left of the compressible code domain with a region of overlap between them, as can be seen in Fig. 9. Information between the compressible and the low Mach number codes is exchanged through the interfaces corresponding to the outflow boundary of the low Mach number code domain and the inflow boundary of the compressible code domain. In a framework of the two-way coupling, different interface conditions supplied from the low Mach number to the compressible code are tested. Several methods are considered:

Method A is injection for $\{u, v, w, T\}$ and ρ is obtained from the continuity equation:

$$u^c = u^{\text{lm}}, \qquad v^c = v^{\text{lm}}, \qquad w^c = w^{\text{lm}}$$
$$T^c = T^{\text{lm}}, \qquad \rho^c - \text{from continuity} \tag{33}$$

Method B is injection for $\{\rho, u, v, w, T\}$:

$$u^c = u^{\text{lm}}, \qquad v^c = v^{\text{lm}}, \qquad w^c = w^{\text{lm}}$$
$$T^c = T^{\text{lm}}, \qquad \rho^c = \rho^{\text{lm}} \tag{34}$$

Method C is injection for $\{u, v, w, T\}$, ρ is calculated from the low Mach number pressure through the equation of state:

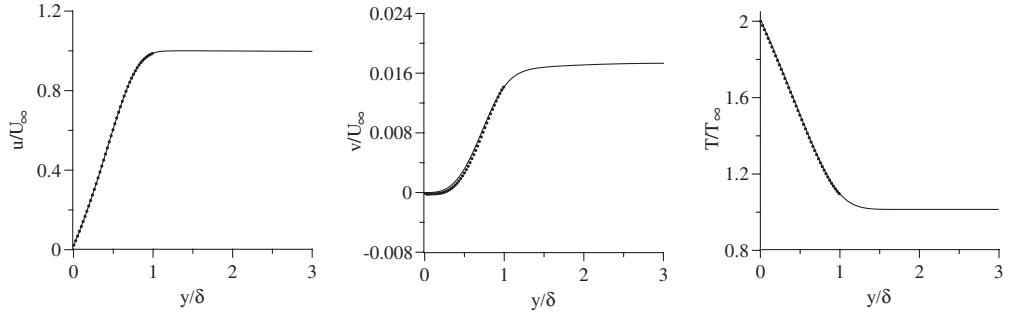
$$u^c = u^{\text{lm}}, \qquad v^c = v^{\text{lm}}, \qquad w^c = w^{\text{lm}}$$
$$T^c = T^{\text{lm}}, \qquad \rho^c = p^{\text{lm}}/T^{\text{lm}} \tag{35}$$

The difference in normalization of pressure is taken into account in Eq. (35) for formulating ρ^c [refer to Eq. (4) and Eq. (30)].

Method D is a Riemann invariant formulation in which Riemann invariants are formed from $\{\rho^{\text{lm}}, u^{\text{lm}}, v^{\text{lm}}, w^{\text{lm}}, T^{\text{lm}}\}$, and variables $\{\rho^c, u^c, v^c, w^c, T^c\}$ are reconstructed from those Riemann invariants according to the procedure described, for example, in [19]

Method E is a Riemann invariant formulation, but ρ is calculated from the low Mach number pressure through the equation of state $\rho = p^{\text{lm}}/T^{\text{lm}}$ and used instead of ρ^{lm} to form Riemann invariants as in method D.

Method B, described in Sec. III, was chosen as the basic method due to its superior performance. In the present section, the reader will see the facts that led to this conclusion. When the low Mach number code is run in an incompressible regime, constant density and temperature equal to their freestream (reference) values are supplied



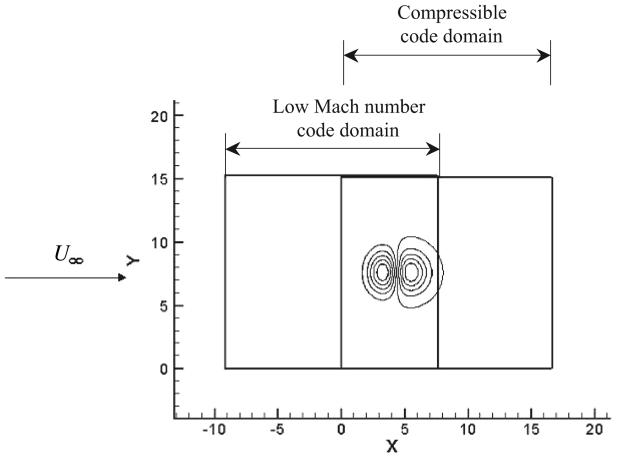
a) Streamwise velocity b) Vertical velocity c) Temperature

Fig. 8 Comparison of the boundary-layer profiles calculated with the two codes; compressible code (lines) and low Mach number code (symbols).

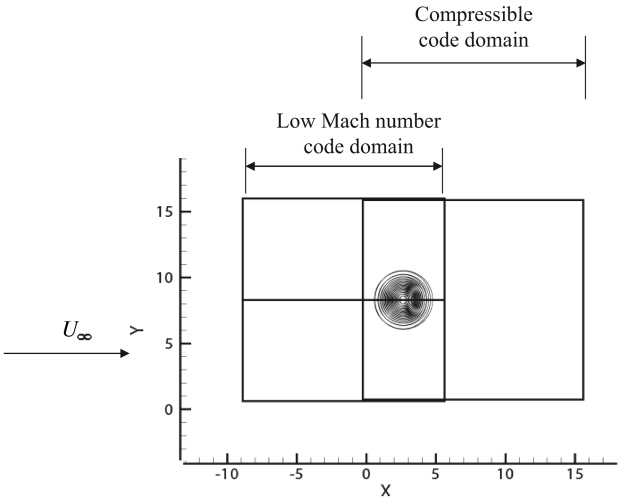
to the compressible domain instead of interpolated values ($\rho^{\text{lm}} = 1$ and $T^{\text{lm}} = 1$ in nondimensional variables).

1. Incompressible–Compressible Regime

Vortical disturbance in the form of a convecting Taylor vortex [Eq. (32)] superimposed on a uniform mean flow is considered to test the two-way coupling procedure with an incompressible formulation of the low Mach number code. All physical parameters of the problem are the same as in the one-way coupling setting described previously, except for the Mach number, which is now $M = 0.15$. Computational domain for this test case is shown in Fig. 9a, where all distances are nondimensionalized by the initial radius of Taylor vortex r_d. The moment when the vortex is inside the overlap region is shown in the figure. Two-dimensional calculations are performed using uniform computational meshes with 128×128 grid points for the low Mach number code and 96×144 grid points for the compressible code.



a) Vortical disturbance; vertical velocity contours are shown



b) Entropy disturbance; temperature contours are shown

Fig. 9 Computational domain for two-way coupling tests.

a. Variation of Interface Conditions. Interface conditions A through D are tested in a constant-density formulation of the low Mach number code. To assess the performance of the interface conditions and compare different methods, the levels of the dilatation in the compressible code are considered. Because the Taylor vortex [Eq. (32)] is a solution of incompressible equations, it is divergence-free, and it therefore represents purely vortical disturbance to the compressible flow. For a uniform mean flow and a small-disturbance amplitude, as in the present case, the linearized modal analysis of Kovásznay [26] is applicable, which shows that the vorticity mode does not produce any dilatation in the compressible field. Therefore, numerical values of the dilatation are expected to be small and represent a good measure of the performance of interface conditions.

Maximum dilatation in the compressible domain is plotted versus the x coordinate of the center of the vortex x_{vortex} as it convects through the domain in Fig. 10 for methods A through D. One can see right away that method A is unstable. Among methods B, C, and D, method C (when the density is obtained from the pressure) shows the smallest values of the dilatation for $x_{\text{vortex}} < 5$, but the levels of dilatation are the largest for $x_{\text{vortex}} > 5$ and they do not come back to zero after the exit of the disturbance. In fact, calculations with method C lead to an instability if continued further. Between methods B and D, dilatation levels are larger for the Riemann invariant method (D) than for the injection method (B). Therefore, the injection method (B), which is a simple interpolation of all five variables $\{\rho, u, v, w, T\}$ from the low Mach number to the compressible code, gives the best overall performance for the present test case.

b. Behavior of the Coupled Solution (Using the Injection Method). Vertical velocity profile versus x coordinate, taken along the horizontal line passing through the center of the vortex, is plotted in Fig. 11 for both the compressible and the low Mach number codes in the coupled calculation with the injection method. Results of the standalone calculations performed with the compressible and the low Mach number codes using the same computational meshes as in the coupled calculation are also plotted for reference. Plots for the three

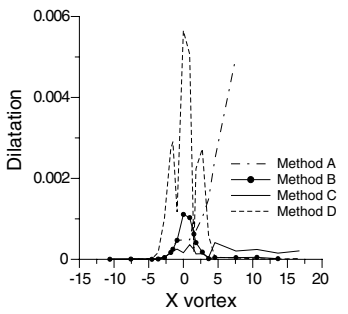
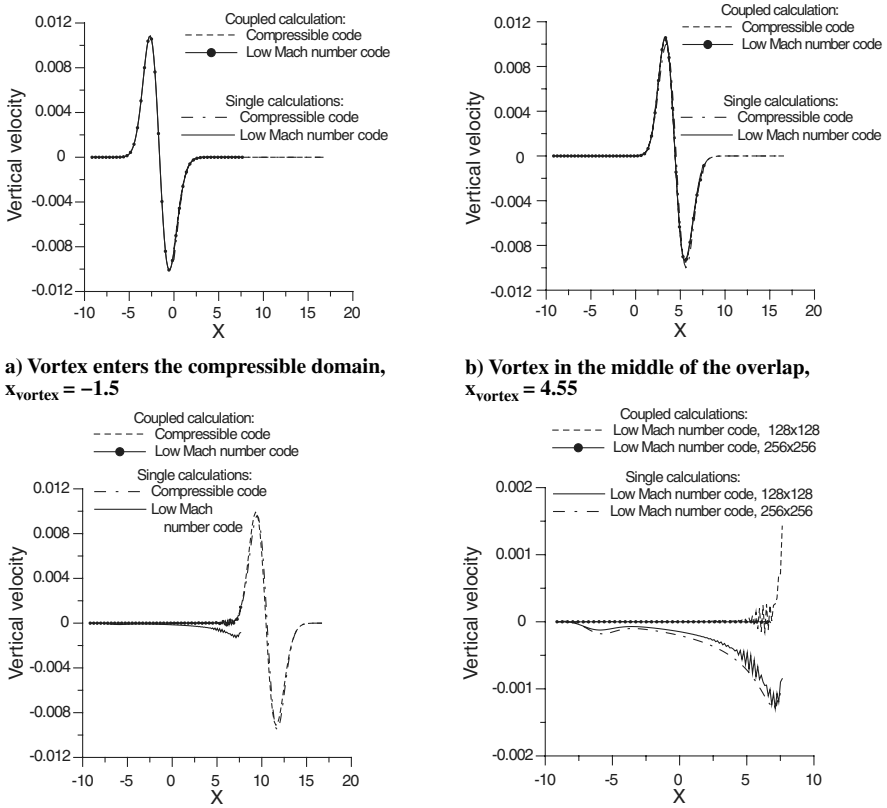


Fig. 10 Dilatation levels in the compressible code versus x coordinate of the vortex center.



a) Vortex enters the compressible domain, $x_{vortex} = -1.5$

b) Vortex in the middle of the overlap, $x_{vortex} = 4.55$

c) Vortex exits the low Mach number domain, $x_{vortex} = 10.6$

d) Vortex exits the low Mach number domain, grid size variation

Fig. 11 Vertical velocity profile along the horizontal line passing through the center of the vortex.

relative positions of the vortex are shown: when the vortex enters the compressible code domain [$x_{\text{vortex}} = -1.5$ (Fig. 11a)], when it is in the center of the overlap region [$x_{\text{vortex}} = 4.55$ (Fig. 11b)], and when it exits the low Mach number code domain [$x_{\text{vortex}} = 10.6$ (Fig. 11c)].

For the first two plots (Figs. 11a and 11b), the vortex center is located to the left from the outflow boundary of the low Mach number code domain, and the difference between all four solutions is very small. However, if we look at the velocity profiles when the vortex is leaving the low Mach number code domain (Fig. 11c), we see that the standalone low Mach number solution differs significantly from the three others. Moreover, some oscillations are noticeable in the low Mach number code solution for both single and coupled calculations. These effects are clearly due to the outflow conditions in the low Mach number code. An influence of outflow conditions is better seen in Fig. 11d, where only the solutions of the low Mach number code are plotted. To investigate whether the oscillations occur because of insufficient resolution in the low Mach number code, the number of grid points in the low Mach number code domain was increased from 128×128 to 256×256. Both standalone and coupled calculations were repeated on a finer grid, with all other parameters of the problem left unchanged. The results of the refined calculations are shown in Fig. 11d with the results of the original calculations. The magnitude of the oscillations is reduced with grid refinement from 10^{-3} to 10^{-6} of freestream velocity in both coupled and standalone settings, showing that the oscillations are indeed controlled by the resolution. Reduction of oscillations at the outflow with grid refinement was observed in other simulations using the low Mach number code. Numerical experiments suggest the following empirical rule for choosing the grid size Δx while using this code in practice:

$$\Delta x / L_d < 0.05 \tag{36}$$

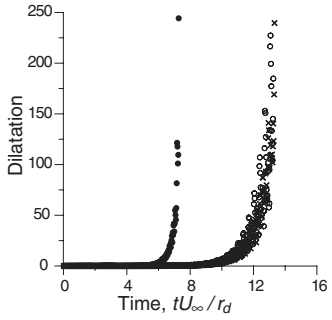
where L_d is the disturbance scale. This condition usually ensures that the oscillations in the low Mach number code due to the outflow boundary conditions are less than 0.1% of the mean value. It is interesting to note that in a standalone refined calculation, the velocity profile, although smooth, deviates even further from the correct value. Apparently, this deviation is due to the drawback of the outflow boundary conditions used in a standalone computation with the low Mach number code, which are convective outflow conditions for velocities [20]. Clearly, this type of outflow condition is not suitable for the case of outgoing organized disturbances.

2. *Variable-Density–Compressible Regime*

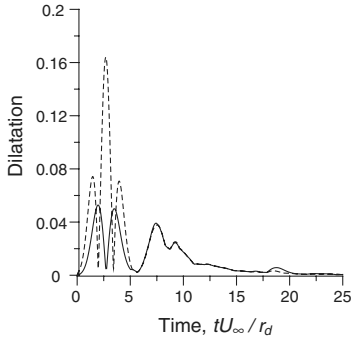
Temperature disturbance characterized by the Gaussian temperature distribution

$$T' = T_d \exp(-r^2/r_d^2) \tag{37}$$

convected by the uniform mean flow from the low Mach number to the compressible domain is considered to test the variable-density formulation of the coupling procedure (r is the distance from the center of the disturbance, r_d is the characteristic radius of the disturbance, and $T_d > 0$ is the disturbance amplitude). No pressure disturbance is introduced so that this disturbance constitutes an entropy mode according to the modal decomposition of Kovásznay [26]. Temperature contours at the moment when the hot spot is in the overlap region are shown in Fig. 9b. All distances are normalized by the disturbance radius r_d. Because the variable-density formulation of the low Mach number code requires three-dimensionality of the grid, a cylindrical grid of $128 \times 64 \times 64$ points with uniform distribution was used for the low Mach number code and a uniform Cartesian grid with $96 \times 144 \times 8$ points was employed for the compressible code. The flow is taken to be quasi-two-dimensional, with no variation of parameters in the z direction. The Mach number for the compressible code is $M = 0.15$, and the Reynolds number for both codes is $Re = U_\infty r_d / \nu_\infty = 8000$. With this Reynolds number, the ratio of convective and viscous time scales is about 10^{-4}, making viscous effects insignificant.



a) Unstable methods: •, density from continuity (method A); ×, density from pressure(method C); ◦, Riemann invariants with density from pressure (method E)



b) Stable methods: ——, injection (method B); – – –, Riemann invariants (method D)

Fig. 12 Dilatation levels in the compressible code versus nondimensional computational time tU_∞/r_d.

a. Variation of Interface Conditions. Methods A through E for supplying interface conditions to the compressible code described in the beginning of this section are tested for a disturbance amplitude $T_d = 1\%$ with the variable-density regime of the low Mach number code. We look again at the dilatation levels in the compressible code to compare different interface conditions. Because disturbance amplitude is again small, Kovásznay's [26] theory is still applicable, stating that the dilatation of an entropy mode is caused only by the viscous dissipation. Because viscous effects are small in the present problem, small values of the dilatation are again expected. Levels of the maximum dilatation in the compressible code are shown in Fig. 12. One can distinguish between unstable methods (Fig. 12a) and stable methods (Fig. 12b). Methods A, C, and E were found to be unstable. In these methods, density is not injected, but either obtained from the continuity equation or calculated from the low Mach number pressure. The same trend was observed in the incompressible formulation of the low Mach number code (Fig. 10). Perhaps it is connected with the fact that the pressure in the low Mach number code plays a different role than the pressure in the compressible code (dynamic variable enforcing the continuity equation versus thermodynamic variable coupled to density and temperature). Therefore, calculating density from the low Mach number pressure is not a correct procedure and leads to an unstable coupling. Two other methods, B and D, result in a bounded value of the dilatation within the considered execution time.

In the previous section, we found that for incompressible-compressible coupling, the dilatation errors for injection method (B) were smaller than for method D, which used Riemann invariants. The same conclusion is reached in the present test problem (Fig. 12b). To make a further comparison of the methods B and D, L_∞ and L_2 errors between numerical and analytical solutions for the temperature and for the streamwise velocity are analyzed next for the case of variable-density–compressible coupling.

Errors in temperature for both the low Mach number and the compressible codes are shown in Fig. 13 and those for streamwise velocity are shown in Fig. 14. It is seen that for the low Mach number code, the dependence of errors on the method is very small, because

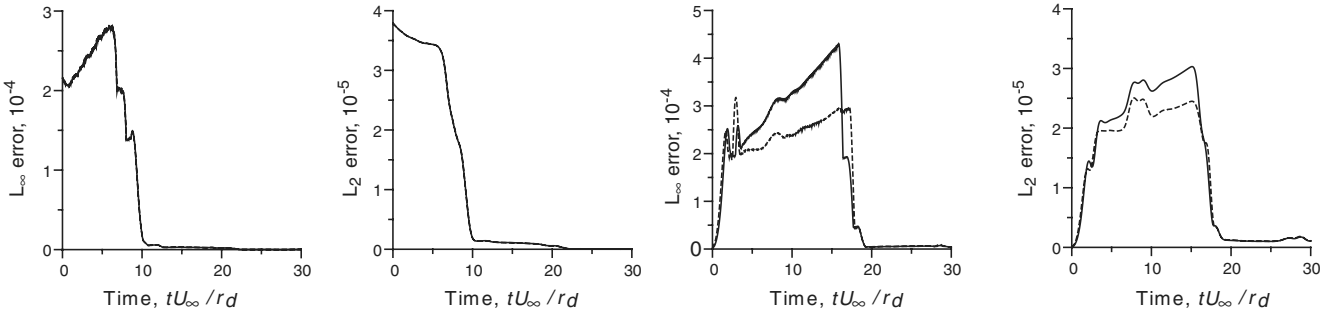


Fig. 13 Errors in temperature versus nondimensional computational time: a–b) low Mach number code and c–d) compressible code; injection (method B) (solid line) and Riemann invariants (method D) (dashed line).

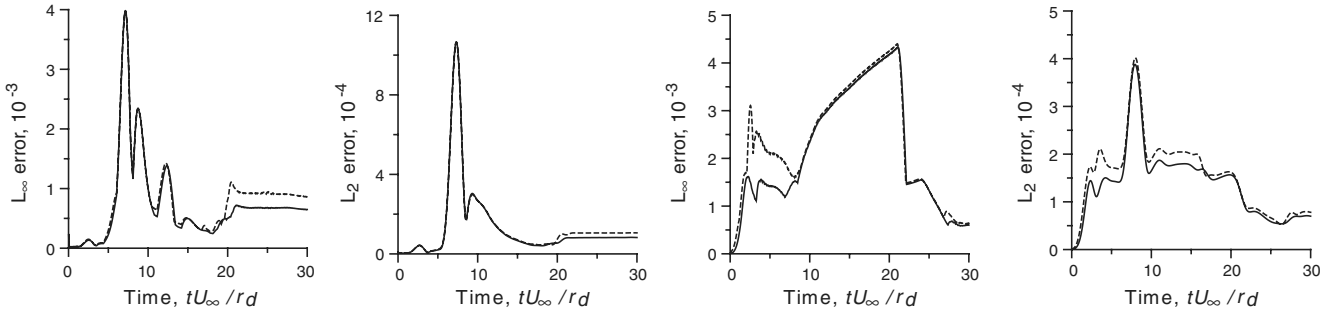
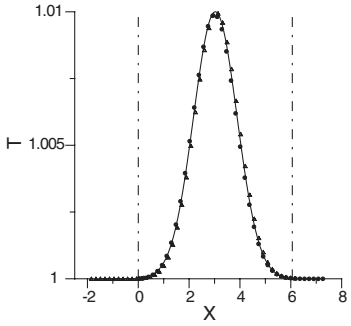
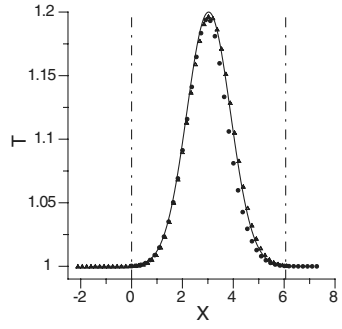


Fig. 14 Errors in streamwise velocity versus nondimensional computational time: a–b) low Mach number code and c–d) compressible code; injection (method B) (solid line) and Riemann invariants (method D) (dashed line).



a) Disturbance amplitude 1%



b) Disturbance amplitude 20%

Fig. 15 Temperature profiles through the center of the disturbance; compressible code (circles), low Mach number code (triangles), analytical solution (solid lines), and boundaries of the overlap region (dashed-dotted lines).

interface conditions in the low Mach number code are not varied in the present test problem and the influence through the feedback mechanism is apparently small. For the compressible code, the dependence on the method is more pronounced. Errors in temperature are slightly larger for the method B and in streamwise velocity for the method D. Comparison of L_∞ and L_2 errors does not reveal a clear superiority of one method over the other. Slightly better performance of the injection method in terms of the dilatation levels combined with the simplicity of implementation resulted in choosing the injection method as the basic method for specifying interface conditions in the compressible code during the coupling procedure.

b. Behavior of the Coupled Solution (Using the Injection Method). Temperature profiles obtained by the two codes when the injection method is used are plotted in Fig. 15. Profiles are taken along the horizontal line passing through the center of the disturbance when it is in the region of overlap. The case of a larger disturbance amplitude $T_d = 20\%$ is also computed for reference. Numerical solutions for the compressible and low Mach number codes, as well as the analytical solution, are shown in Fig. 15. Agreement between the two numerical solutions and the analytical solution is very good, showing good performance of the coupling procedure for both small- and large-disturbance amplitudes.

V. Conclusions

Stable and accurate coupling methodology is proposed to combine compressible and low Mach number codes into a single solver. This development is motivated by the desire to extend computational capability of computational fluid dynamics tools to perform integrated simulations of large-scale multicomponent engineering flow systems. The coupling is accomplished by the exchange of unsteady interface conditions between the two codes on overlapping grids. The methodology is tested by its application to steady and unsteady laminar problems. Both constant-density (incompressible) and variable-density regimes of the low Mach number code are investigated. The first group of tests, performed in a framework of one-way coupling, when only the low Mach number code obtains information from the compressible code without providing feedback, confirmed that numerical solutions of the two codes in the region of overlap are close to each other, provided that the Mach number in the region of overlap is small. The closeness of the numerical solutions is necessary for the convergence of the fully coupled problem. An error on an order of M^2 was recovered, consistent with the analytical discrepancy between the solutions of the two equation sets. Two-way coupling tests, when the mutual information exchange between the two codes is activated, allowed comparison of several methods of specifying interface conditions into the compressible code. Unstable methods were identified in which the density supplied to the compressible code is either calculated from the low Mach number pressure or obtained from the continuity equation. The injection method, in which all the interpolated variables are simply transformed from one code to another, is found to be the best for its stability, accuracy, and simplicity of implementation. Numerical solutions of the coupled problem obtained with the injection method are compared with the solutions of standalone calculations as well as analytical solutions, and good agreement is observed. Further application of the developed solver to unsteady turbulent simulations of film-cooling flow from inclined cylindrical holes [5] confirms the applicability of this procedure to calculations of complex engineering problems.

Acknowledgments

This work was supported by the Advanced Simulation and Computing (ASC) program of the U.S. Department of Energy. The compressible code was developed with U.S. Air Force Office of Scientific Research support under grant no. F94620-01-1-0138. The computer time was provided by the parallel cluster of Stanford University built with the support of the Defense University Research Instrumentation Program and the U.S. Department of Defense Supercomputer Center.

References

[1] Schlüter, J. U., Wu, X., Kim, S., Shankaran, S., Alonso, J. J., and Pitsch, H., "A Framework for Coupling Reynolds-Averaged with Large-Eddy Simulations for Gas Turbine Applications," *Journal of Fluids Engineering*, Vol. 127, No. 4, July 2005, pp. 806–815. doi:10.1115/1.1994877

[2] Medic, G., You, D., Kalitzin, G., Pitsch, H., van der Weide, E., and Alonso, J. J., "Integrated Computations of an Entire Jet Engine," ASME/IGTI Turbo Expo, American Society of Mechanical Engineers, Paper GT 2007-27094.

[3] Hahn, S., Duraisamy, K., Iaccarino, G., Nagarajan, S., Sitaraman, J., Wu, X., Alonso, J. J., Baeder, J. D., Lele, S. K., Moin, P., and Scmitz, F., "Coupled High-Fidelity URANS Simulation for Helicopter Applications," *Annual Research Briefs*, Center for Turbulence Research, Stanford Univ., Stanford, CA, 2006.

[4] Nordström, J., Svärd, M., Shoeybi, M., Ham, F., Mattson, K., Iaccarino, G., vand der Weide, E., and Gong, J., "A Stable, Efficient, and Adaptive Hybrid Method for Unsteady Aerodynamics," *Annual Research Briefs*, Center for Turbulence Research, Stanford Univ., Stanford, CA, 2006.

[5] Peet, Y., "Film Cooling from Inclined Cylindrical Holes Using Large Eddy Simulations," Ph.D. Thesis, Dept. of Aeronautics and Astronautics, Stanford Univ., Stanford, CA, 2006.

[6] Berger, M. J., "Stability of Interfaces with Mesh Refinement," *Mathematics of Computation*, Vol. 45, No. 172, Oct. 1985, pp. 301–318. doi:10.2307/2008126

[7] Berger, M. J., "On Conservation at Grid Interfaces," *SIAM Journal on Numerical Analysis*, Vol. 24, No. 5, Oct. 1987, pp. 967–984. doi:10.1137/0724063

[8] Chesshire, G., and Henshaw, W. D., "A Scheme for Conservative Interpolation on Overlapping Grids," *SIAM Journal on Scientific Computing*, Vol. 15, No. 4, July 1994, pp. 819–845. doi:10.1137/0915051

[9] Carpenter, M. H., Nordström, J., and Gottlieb, D., "A Stable and Conservative Interface Treatment of Arbitrary Spatial Accuracy," *Journal of Computational Physics*, Vol. 148, No. 2, Jan. 1999, pp. 341–365. doi:10.1006/jcph.1998.6114

[10] Mattson, K., and Nördstrom, J., "Summation by Parts Operators for Finite Difference Approximations of Second Derivatives," *Journal of*

Computational Physics, Vol. 199, No. 2, 2004, pp. 503–540. doi:10.1016/j.jcp.2004.03.001

[11] Manna, M., Vacca, A., and Deville, M. O., "Preconditioned Spectral Multi-Domain Discretization of the Incompressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 201, No. 1, Nov. 2004, pp. 204–223. doi:10.1016/j.jcp.2004.05.011

[12] Pfeiffer, H. P., Kidder, L. E., Scheel, M. A., and Teukolsky, S. A., "A Multidomain Spectral Method for Solving Elliptic Equations," *Computer Physics Communications*, Vol. 152, No. 3, May 2003, pp. 253–273. doi:10.1016/S0010-4655(02)00847-0

[13] Chang, S. L., and Chien, L. S., "Domain Decomposition Algorithms for Fourth-Order Nonlinear Elliptic Eigenvalue Problems," *Journal of Computational Physics*, Vol. 191, No. 2, Nov. 2003, pp. 476–501. doi:10.1016/S0021-9991(03)00327-9

[14] Kopriva, D. A., "Multidomain Spectral Solution of Compressible Viscous Flows," *Journal of Computational Physics*, Vol. 115, No. 1, Nov. 1994, pp. 184–199. doi:10.1006/jcph.1994.1186

[15] Hesthaven, J. S., "A Stable Penalty Method for Compressible Navier-Stokes Equations, 3: Multidimensional Domain Decomposition Schemes," *SIAM Journal on Scientific Computing*, Vol. 20, No. 1, 1998, pp. 62–93. doi:10.1137/S1064827596299470

[16] Nördstrom, J., and Carpenter, M. H., "Boundary and Interface Conditions for High Order Finite Difference Methods Applied to Euler and Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 148, No. 2, Jan. 1999, pp. 621-645. doi:10.1006/jcph.1998.6133

[17] Strikwerda, J. C., and Scarbnick, C. D., "A Domain Decomposition Method for Incompressible Viscous Flow," *SIAM Journal on Scientific Computing*, Vol. 14, No. 1, Jan. 1993, pp. 49–67. doi:10.1137/0914004

[18] Mattson, K., Iourokina, I., and Ham, F., "Stable and Accurate Coupling Between Compressible and Incompressible Flow Solvers," *Annual Research Briefs*, Center for Turbulence Research, Stanford Univ., Stanford, CA, 2005.

[19] Xiong, Z., "Stagnation Point Flow and Heat Transfer Under Free-Stream Turbulence," Ph.D. Thesis, Dept. of Mechanical Engineering, Stanford Univ., Stanford, CA, 2004.

[20] Pierce, C. D., "Progress-Variable Approach for Large Eddy Simulation of Turbulent Combustion," Ph.D. Thesis, Dept. of Mechanical Engineering, Stanford Univ., Stanford, CA, 2001.

[21] Lele, S. K., "Direct Numerical Simulations of Compressible Turbulent Flows: Fundamentals and Applications," *Transition, Turbulence and Modeling*, Kluwer Academic, Stockholm, 1998, pp. 421–488.

[22] Müller, B., "Low Mach Number Asymptotics of the Navier-Stokes Equations and Numerical Implications," *30th Computational Fluid Dynamics*, VKI LS 1999–03, Von Kármán Inst. for Fluid Dynamics, Rhode-Saint-Genèse, Belgium, Mar. 1999.

[23] Germano, M., Piomelli, U., Moin, P., and Cabot, W. H., "A Dynamic Subgrid-Scale Eddy Viscosity Model," *Physics of Fluids A*, Vol. 3, 1991, pp. 1760–1765. doi:10.1063/1.857955

[24] Iourokina, I. V., and Lele, S. K., "Towards Large Eddy Simulation of Film-Cooling Flows on a Model Turbine Blade Leading Edge," AIAA Paper 2005-0670, Jan. 2005.

[25] Taylor, G. I., "On the Dissipation of Eddies," Aeronautical Research Council, Reports and Memorandum No. 598, 1918.

[26] Kovásznay, L. S. G., "Turbulence in Supersonic Flow," *Journal of the Aeronautical Sciences*, Vol. 20, 1953, pp. 657–682.

C. Kaplan
Associate Editor