AIAA JOURNAL
Vol. 45, No. 9, September 2007

Manufactured Solution for Computational Fluid Dynamics Boundary Condition Verification

Ryan B. Bond,* Curtis C. Ober,† Patrick M. Knupp,‡ and Steven W. Bova§
Sandia National Laboratories, Albuquerque, New Mexico 87185

DOI: 10.2514/1.28099

Order-of-accuracy verification is necessary to ensure that software correctly solves a given set of equations. One method for verifying the order of accuracy of a code is the method of manufactured solutions. This study documents the development of a manufactured solution that allows verification of not only the Euler, Navier–Stokes, and Reynolds-averaged Navier–Stokes equation sets, but also some of their associated boundary conditions: slip, no-slip (adiabatic and isothermal), and outflow (subsonic, supersonic, and mixed). To demonstrate the usefulness of this manufactured solution, it has been used for order-of-accuracy verification in a compressible computational fluid dynamics code. All of the results shown are on skewed, nonuniform, three-dimensional meshes. The manufactured solution and sequence of meshes are designed to allow asymptotic results to be obtained with reasonable computational cost. In addition to the order of accuracy of the full code for various equation sets and boundary conditions, the order of accuracy of code portions used to calculate solution gradients has been measured as well.

Nomenclature

A, B, C	=	generic constants in manufactured solution
$A_f, B_f, A_g, B_g, A_p, B_p, C_p, A_t, A_v, B_v, C_v$	=	constants in manufactured solution
a	=	speed of sound
C	=	generic constant
C_1, C_2	=	coefficients in Sutherland's law
C_p	=	specific heat at constant pressure
$c_{b1}, c_{b2}, c_{v1}, \sigma, c_{\omega1}, c_{\omega2}, c_{\omega3}$	=	closure coefficients for Spalart–Allmaras model
D	=	diffusion in Spalart–Allmaras model
d	=	normal distance to the wall
e	=	specific internal energy
e_t	=	specific total internal energy
$F, G, \tilde{G}, H$	=	functions for surface definitions
$f_{v1}, f_{v2}, f_\omega, \chi, g, r, \tilde{S}$	=	auxiliary functions for Spalart–Allmaras model
f, g	=	generic functions in manufactured solution
I	=	identity matrix
k	=	thermal conductivity
k_t	=	turbulent thermal conductivity
M	=	Mach number
m	=	generic integer exponent
N	=	number of mesh nodes in each dimension
$\hat{n}, n$	=	unit normal vector, normal position component
p	=	static pressure
Pr	=	Prandtl number
Pr_t	=	turbulent Prandtl number
q_j	=	heat flux vector (tensor notation)
R	=	specific gas constant
r	=	radius of curvature
S^*	=	source term derived from manufactured solution
S_D	=	destruction in Spalart–Allmaras model
S_P	=	production in Spalart–Allmaras model
T	=	static temperature
t	=	time
u	=	generic solution to a partial differential equation
u^*	=	generic manufactured solution
u, v, w	=	Cartesian velocity components
u_j	=	velocity vector (tensor notation)
v_n	=	normal velocity component
v_t	=	tangential velocity component
x_j	=	position vector (tensor notation)
x, y, z	=	Cartesian position components
$\mathcal{A}_1, \mathcal{A}_2, \mathcal{B}_1, \mathcal{B}_2$	=	auxiliary functions in manufactured solution
$\mathcal{B}$	=	generic boundary operator
$\mathcal{D}$	=	generic differential operator
$\mathcal{F}$	=	inviscid flux hypervector
γ	=	ratio of specific heats
δ_{ij}	=	Kronecker delta
ϵ	=	numerical error
κ	=	Kármán constant
μ	=	molecular viscosity
μ_t	=	turbulent (eddy) viscosity
$\tilde{\nu}$	=	working variable for Spalart–Allmaras model
ρ	=	density
τ_{ij}	=	specific Reynolds stress tensor
ϕ	=	generic flow variable
$\Omega, \partial\Omega$	=	generic domain, boundary of generic domain
Ω_{ij}	=	mean rotation tensor

Received 29 September 2006; revision received 20 February 2007; accepted for publication 24 March 2007.

*Aerosciences Department, Mail Stop 0825; rbbond@sandia.gov. Senior Member AIAA.
†Exploratory Simulations Technologies, Mail Stop 0316; ccober@sandia.gov. Senior Member AIAA.
‡Optimization and Uncertainty Estimation Department, Mail Stop 1111; pknupp@sandia.gov.
§Thermal/Fluid Computational Engineering Sciences, Mail Stop 0382; swbova@sandia.gov.

I. Introduction

THE purpose of this work is to document the development of a manufactured solution capable of testing the governing equations and boundary conditions (BCs) commonly implemented in computational fluid dynamics (CFD) codes. A critical aspect of the

manufactured solution development is the definition of a set of meshes that allow the BCs to be tested in the most general way and also allow asymptotic results to be obtained with reasonable computational expense. To demonstrate that this manufactured solution is both correct and useful, it has been applied to the verification of Premo [1], a compressible fluid dynamics code being developed at Sandia National Laboratories. Premo is used to determine aerodynamic performance of complex geometries, and it is built within the SIERRA framework [2] to allow multiphysics coupling (e.g., heat transfer and fluids for ablation or structures and fluids for aeroelasticity). It is a parallel, unstructured, edge-based, finite volume code. Thus two separate but related things are discussed: first, a manufactured solution for verification of CFD codes and, second, a verification study of Premo using this manufactured solution. The development of the manufactured solution is independent of the specific discretization of Premo; however, explanation of the verification study of Premo does require some discussion about Premo's discretization and internal structure. These details are presented only when necessary: a complete discussion of Premo can be found in [1]. Coding mistakes or algorithmic weaknesses identified in the verification study may be unique to unstructured, edge-based, finite volume codes (e.g., gradient reconstruction algorithms), but the manufactured solution and general process can be used in the same manner to verify codes with other discretization schemes.

In [3,4], Bond et al. give earlier versions of the manufactured solution used in the present work. In [3–5], Bond et al. show preliminary results. The current version of the manufactured solution and current sequence of meshes allow asymptotic behavior to be more easily observed for the order-of-accuracy verification tests with reasonable computational costs. The current manufactured solution is detailed in Sec. VI, and the current sequence of meshes is described in Sec. V.

In [3], two issues were identified: the slip condition showed unordered behavior (i.e., nonpositive order of accuracy) and the Euler equations using least squares gradient reconstruction demonstrated first-order behavior when second-order was expected. After [3] was published, two coding mistakes were found and corrected. The mistake associated with the slip BC involved the misuse of a unit normal calculation function intended for planar surfaces only. Bond et al. [4] repeated the order-verification test for the slip BC after the known coding mistake was corrected, but unfortunately, positive order of accuracy (i.e., consistency) was still not observed. The poor least squares results in [3] were caused by an indexing mistake in the gradient calculation. In [4], Bond et al. demonstrated that second-order behavior was observed after the coding mistake was corrected. The work was continued in [4] by applying the manufactured solution to the verification of the Navier–Stokes equations (another coding mistake was found and corrected during this process). Preliminary results were shown in [5] for the Reynolds-averaged Navier–Stokes (RANS) equations. The current work uses the same manufactured solution as [5], but one additional mesh level is included for the interior equation set tests. For the BCs, many of the issues uncovered in [3–5] have been resolved by the introduction of characteristic-based BCs (described in the Appendix), and new results are presented in Sec. VIII.A and VIII.B.

The method of manufactured solutions (MMS) provides order-of-accuracy verification of the implemented equation sets along with their various BCs [6,7]. Such verification is a rigorous way to ensure that the governing equations are solved correctly. The goal is to document that the observed order of accuracy of certain algorithms matches their formal order of accuracy. Such documentation builds confidence in the user community that those algorithms have been properly implemented and do not have inherent weaknesses which reduce their orders of accuracy. One important but often overlooked benefit of formal order verification is that it can not only find implementation errors, it can also identify weaknesses in correctly implemented formulations and algorithms. In many cases, the failure of a code to match a desired order of accuracy may require a different formulation as a remedy, rather than correction of a coding mistake.

MMS derives its name from a solution which is manufactured for the sole purpose of verifying the order of accuracy of the code. It is a special case of the method of exact solutions, where discrete solutions on a sequence of uniformly refined meshes are compared with an exact solution. Manufactured solutions are used in the following cases:

1) There are no classical exact solutions (i.e., those obtained by forward solution, as opposed to manufactured solutions which are obtained by backward solution).

2) Classical exact solutions are not sufficiently general to fully test a code.

3) Classical exact solutions are not smooth enough to allow proper implementation and error analysis.

Unlike a classical exact solution, a manufactured solution does not actually need to satisfy the original governing equation set for a given set of initial and boundary conditions. Instead, the original equation set is modified through the addition of a source term so that the manufactured solution is an exact solution to the equation set with this additional source term. The particular form of the source term depends on the manufactured solution selected. This form is found by applying the equation operators to the manufactured solution to obtain an analytic formula for the source. The source term is then added to the original equation set to balance it. If functions for calculating the source term are added to the code, it can be used to solve the equation set with the additional source term. Then the discrete solutions produced by the code can be compared to the manufactured solution to determine the discretization error. A comparison of the discretization error on a series of uniformly refined meshes will either verify that the observed order of accuracy matches the formal order of accuracy, or it will not. In the latter case, it may indicate a coding mistake or improper formulation. Other possibilities are discussed in [6,7]. This assumes that the formal order of accuracy is known with certainty, which may not be the case. For cases where it is not, there is probably a desired order of accuracy which the code is meant to achieve.

The RANS equations are given as

$$\frac{\partial(\bar{\rho})}{\partial t} + \frac{\partial(\bar{\rho}\tilde{u}_j)}{\partial x_j} = 0 \tag{1}$$

$$\frac{\partial(\bar{\rho}\tilde{u}_i)}{\partial t} + \frac{\partial(\bar{\rho}\tilde{u}_i\tilde{u}_j + \bar{p}\delta_{ij} - \overline{\tau_{ij}})}{\partial x_j} = 0 \tag{2}$$

$$\frac{\partial(\bar{\rho}\tilde{e}_t)}{\partial t} + \frac{\partial(\bar{\rho}\tilde{u}_j\tilde{e}_t + \bar{p}\tilde{u}_j - \tilde{u}_i\overline{\tau_{ij}} + \overline{q_j})}{\partial x_j} = 0 \tag{3}$$

where the bar over a symbol indicates Reynolds averaging and the tilde over a symbol indicates Favre averaging. These transport equations are closed by auxiliary relations for thermodynamic and transport properties. The definitions for internal energy (e) and total energy (e_t) for this work are

$$e = \frac{1}{\gamma - 1}RT \tag{4}$$

$$e_t = e + \frac{u_i u_i}{2} \tag{5}$$

The ideal gas law is used for the equation of state:

$$p = \rho RT \tag{6}$$

The preceding relations reference the ratio of specific heats (γ) and the specific gas constant (R). For air, these quantities are 1.4 and 287.0 $\mathrm{N \cdot m/kg \cdot K}$, respectively.

The viscous stress tensor and heat flux vector are given by

$$\overline{\tau_{ij}} = (\bar{\mu} + \mu_t)\left[\left(\frac{\partial\tilde{u}_i}{\partial x_j} + \frac{\partial\tilde{u}_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}\frac{\partial\tilde{u}_k}{\partial x_k}\right] \tag{7}$$

$$\overline{q_i} = -(\bar{k} + k_t)\frac{\partial \bar{T}}{\partial x_i} \tag{8}$$

The molecular viscosity (μ) is given by Sutherland's law:

$$\mu = \frac{C_1 T^{1.5}}{T + C_2} \tag{9}$$

whereas the eddy viscosity (μ_t) is determined by the choice of turbulence model. The molecular and turbulent thermal conductivities are determined by assuming a constant Prandtl number (Pr) and turbulent Prandtl number (Pr_t):

$$k = \frac{\gamma R}{\gamma - 1}\frac{\mu}{Pr} \tag{10}$$

$$k_t = \frac{\gamma R}{\gamma - 1}\frac{\mu_t}{Pr_t} \tag{11}$$

For this work, the Spalart–Allmaras turbulence model [8] was used. In compressible form, the single transport equation for the working variable $\tilde{\nu}$ can be expressed as

$$\frac{\partial(\bar{\rho}\tilde{\nu})}{\partial t} + \frac{\partial(\bar{\rho}\tilde{u}_j\tilde{\nu})}{\partial x_j} = D + S_P - S_D \tag{12}$$

where

$$S_P = c_{b1}\bar{\rho}\,\tilde{\nu}\,\tilde{S} \tag{13}$$

$$S_D = c_{\omega 1} f_\omega \bar{\rho}\left(\frac{\tilde{\nu}}{d}\right)^2 \tag{14}$$

$$D = \frac{1}{\sigma}\frac{\partial}{\partial x_j}\left[(\mu + \bar{\rho}\,\tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right] + \frac{c_{b2}\bar{\rho}}{\sigma}\frac{\partial \tilde{\nu}}{\partial x_j}\frac{\partial \tilde{\nu}}{\partial x_j} \tag{15}$$

The closure coefficients and auxiliary relations are given by

$$c_{b1} = 0.1355 \qquad c_{b2} = 0.622 \qquad c_{v1} = 7.1 \qquad \sigma = \frac{2}{3}$$

$$c_{\omega 1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma} \qquad c_{\omega 2} = 0.3 \qquad c_{\omega 3} = 2 \qquad \kappa = 0.41$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \qquad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \qquad f_\omega = g\left[\frac{1 + c_{\omega 3}^6}{g^6 + c_{\omega 3}^6}\right]^{\frac{1}{6}}$$

$$\chi = \frac{\tilde{\nu}}{\nu} \qquad g = r + c_{\omega 2}(r^6 - r) \qquad r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}$$

$$\tilde{S} = \sqrt{2\Omega_{ij}\Omega_{ij}} + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}$$

The eddy viscosity is given by

$$\mu_t = \bar{\rho}\,\tilde{\nu}\,f_{v1} \tag{16}$$

To manufacture a solution to Eqs. (1–3), Eq. (12), and their auxiliary relations, analytic forms of six variables must be defined. These six variables must be independent of one another (e.g., this variable set can not contain ρ, u, and ρu because the third is simply a product of the first two). The most convenient set of variables (whether conservative, primitive, characteristic, or a combination of the three) may depend upon the BC being tested. For this work, the variables p, T, u, v, w, and $\tilde{\nu}$ were used. This manufactured solution may also be used for the Navier–Stokes set or the Euler set by simply dropping the extraneous turbulence variable.

A proposed form for a steady manufactured solution of any flow variable ϕ is

$$\phi = \phi_0[1 + f(Ax)f(By)f(Cz)] \tag{17}$$

where A, B, and C are constants. The function f must be chosen so that this form satisfies the requirements of Knupp and Salari in [6], pp. 44–46. To avoid symmetry within the solution, $A \neq B \neq C \neq 0$. If ϕ_0 is a constant, this form is a separated solution plus a constant [9]. The separated form makes deriving manufactured solutions with specific Neumann BCs (such as a vanishing normal derivative) easier. Constant values for ϕ_0 are not sufficient for all of the BCs of interest, but for an appropriately chosen nonconstant ϕ_0, ϕ will satisfy the desired BCs.

II. MMS Testing of Boundary Conditions for Hyperbolic Systems of Equations

For hyperbolic or mixed-character equations, flow of information through the domain and its associated boundaries is restricted by the local mathematical character (i.e., whether the equations are hyperbolic, parabolic, or elliptic at a given point, and if they are hyperbolic, the direction of the characteristics). The concept of information, or small perturbations, propagating into or out of the domain is distinct from the flux of mass, momentum, or energy through the boundary. The eigenvalues of a system of hyperbolic or mixed-character equations determine the directions in which the information will propagate. For the three-dimensional Euler equations, there are five eigenvalues which dictate the flow of information at a bounding surface: one is the velocity normal to the boundary plus the speed of sound, $v_n + a$; one is the normal velocity minus the speed of sound, $v_n - a$, and the other three are the normal velocity, v_n (i.e., this eigenvalue is repeated).

Conditions that are enforced on domain boundaries are frequently referred to as BCs in the engineering literature, but some do not satisfy the mathematical definition of a BC. A BC in the mathematical sense is an additional constraint imposed at a boundary to ensure a unique solution to a well-posed set of equations. A good example of something that is not a BC is

$$\frac{\partial p}{\partial n} = \frac{\rho v_t^2}{r}$$

which, although it is enforced on a slip boundary, does not provide any additional constraint to the governing equations. Section III.A.1 gives a more thorough explanation of this equation and the slip condition.

The number of mathematical BCs which need to be enforced at a given boundary segment to determine a unique solution is equal to the number of negative eigenvalues. For the Euler equations at a supersonic outflow boundary, there are no negative eigenvalues, and so no mathematical BCs need to be enforced there. For a subsonic outflow or slip boundary, there is one negative eigenvalue; for a subsonic inflow boundary, there are four, and for a supersonic inflow, there are five.

Consider a homogeneous differential equation

$$\mathcal{D}(u) = 0 \tag{18}$$

on a domain Ω and its associated BCs

$$\mathcal{B}(u) = 0 \tag{19}$$

on the boundary of that domain $\partial\Omega$. The following discussion is simplified by starting with a homogeneous equation, but the conclusions apply equally to a nonhomogeneous equation. For a given manufactured solution u^*, the source term S^* (generated by operation of the differential operator $\mathcal{D}$ upon u^*) is used to create a new equation which the manufactured solution satisfies

$$\mathcal{D}(u^*) = S^* \tag{20}$$

On the interior of Ω, S^* corrects for the failure of u^* to satisfy the homogeneous equation. For the case where $\mathcal{D}(u) = 0$ is also enforced on $\partial\Omega$, S^* will likewise correct for the discrepancy between

$\mathcal{D}(u)$ and $\mathcal{D}(u^*)$ on $\partial\Omega$. However, S^* does not correct any discrepancy between u^* and the mathematical BCs (the number of which is equal to the number of negative eigenvalues).

If the interior equations implemented in a code are to be tested using the manufactured solution, then Dirichlet BCs can be set so that the BCs enforced by the code are equal to u^* on $\partial\Omega$ (i.e., u^* is defined without concern to its value on $\partial\Omega$, then $\mathcal{B}$ is defined so that $\mathcal{B}(u^*) = 0$ on $\partial\Omega$). A similar setup can be used for Neumann BCs, provided that the implementation allows for spatially and/or temporally varying values of the partial derivatives of u^* on $\partial\Omega$. However, if a specialized BC is to be verified using MMS, then $\mathcal{B}$ must represent the specialized BC. In this case, it is convenient to design u^* to satisfy $\mathcal{B}(u^*) = 0$ on the portion of $\partial\Omega$ where it is to be enforced. (There are alternative approaches, which are not discussed here.) Note that only BCs by the mathematical definition (constraints corresponding to negative eigenvalues) must be satisfied by u^*. When designing a manufactured solution for testing specialized BCs, it is helpful to have the solution satisfy as many different specialized BCs as possible (in different regions) so that the costs of generating the manufactured solution and implementing the source term are amortized over many BC order-of-accuracy tests.

III. Solid Surface Boundaries

To test the solid surface BCs, a surface must first be defined. Let $F = F(x, y, z)$ be a differentiable function from $\Re^3$ into $\Re$ such that $F = C$ defines a surface for any constant C. The unit normal is defined by

$$\hat{n} = \frac{\nabla F}{\|\nabla F\|} \tag{21}$$

A manufactured solution can be constructed so that any variable ϕ satisfies either a Dirichlet (i.e., ϕ specified) or Neumann (i.e., $\partial\phi/\partial n$ specified) BC along the surface. So that Eq. (21) will represent the outward facing normal, the domain of interest is taken to be on the side $F - C < 0$.

A. Conditions on Velocity

The two solid surface conditions discussed in this work are the slip condition ($\boldsymbol{v} \cdot \hat{n} = 0$) and the no-slip condition ($\boldsymbol{v} = 0$).

1. Slip Condition

The slip condition arises from the impermeability of the solid surface. It states that nothing can be convected through the surface. This can be expressed as $v_n = 0$, where v_n is the velocity component normal to the surface, or equivalently,

$$\boldsymbol{\mathcal{F}} \cdot \hat{n} = \begin{bmatrix} \rho\boldsymbol{v} \\ \rho\boldsymbol{v} \otimes \boldsymbol{v} + p\boldsymbol{I} \\ \rho h_t \boldsymbol{v} \end{bmatrix} \cdot \hat{n} = \begin{bmatrix} 0 \\ p\hat{n} \\ 0 \end{bmatrix} \tag{22}$$

where $\boldsymbol{\mathcal{F}}$ is the flux hypervector. For an Euler flow, this condition is the actual mathematical BC. The oft quoted

$$\frac{\partial p}{\partial n} = \frac{\rho v_t^2}{r}$$

(where v_t is the velocity component tangent to the surface and r is the surface radius of curvature in the streamwise direction) is not a mathematical BC. Rather, it is a consequence of the governing equations (in this case, the normal momentum equation). Because of this, it is not necessary for the manufactured solution to satisfy any constraint on the pressure (or any variable other than v_n).

For any surface defined by the equation $F = C_s$, impermeability implies that

$$\nabla F \cdot \boldsymbol{v} = F_x u + F_y v + F_z w = 0 \tag{23}$$

If analytic forms are chosen for u and v in the definition of the manufactured solution, then w can be defined as follows to ensure that the slip condition holds on any F-constant surface:

$$w = -\frac{F_x u + F_y v}{F_z} \tag{24}$$

This imposes an additional constraint on the function F: $F_z \neq 0$; $\forall(x, y, z) \in \Omega$. This can be easily enforced by defining $F(x, y, z) = \tilde{F}(x, y) - z$. Thus the surface $F = C_s$ can be expressed as $z = \tilde{F}(x, y) - C_s$. With this surface definition, u and v can be left in the general form of Eq. (17) with no additional constraints.

2. No-Slip Condition

The no-slip condition is actually slightly simpler. The following equation for u satisfies $u = 0$ on the surface $F = C_{n-s}$:

$$u = (C_{n-s} - F)u_0[1 + f(Ax)f(By)f(Cz)] \tag{25}$$

If v and w are also of the form given in Eq. (25), then the manufactured solution satisfies the no-slip condition along $F = C_{n-s}$. If, however, w is of the form of Eq. (24), then the manufactured solution will satisfy the no-slip condition on $F = C_{n-s}$ as well as the slip condition on any $F = C_s$ where $C_s \neq C_{n-s}$. This is convenient because it allows the same manufactured solution to be used for testing the slip and no-slip conditions. Note that $C_s \neq C_{n-s}$ is required so that a nonzero slip velocity will exist on $F = C_s$.

B. Additional Boundary Condition for the Navier–Stokes Equation Set

Because of the presence of thermal conductivity, the Navier–Stokes equation set requires the imposition of one additional BC at a no-slip surface. This BC is most often either a Dirichlet BC (i.e., T specified) or a Neumann BC (i.e., $\partial T/\partial n$ specified). The Neumann condition can also be expressed by specifying the heat flux. If Dirichlet and Neumann BCs are implemented in the code, and they are written very generally (i.e., as functions of spatial and temporal variables), then the manufactured solution choice for temperature does not need to satisfy special requirements at the boundary. In this case, the code can just be given the BC that the manufactured solution already satisfies. It can be expressed as either a Dirichlet or Neumann BC. However, if the code has a special case, such as uniform temperature ($T = T_c$) or adiabatic boundary ($\partial T/\partial n = 0$), then the manufactured solution must satisfy the special case to verify its implementation using the approach chosen for this work.

1. Adiabatic Boundary

The temperature at an adiabatic surface $F = C_a$ is subject to a Neumann BC $\partial T/\partial n = 0$. A manufactured solution for T which satisfies this can be used for adiabatic tests. Although more general solutions exist, one solution is given by Eq. (26) in Sec. III.B.3.

2. Isothermal Boundary

The isothermal boundary is formulated along a surface $F = C_i$ similar to the way that the no-slip condition on the velocity was formulated. In this case, it is necessary to ensure that $T = T_c$ at $F = C_i$ and that T has sufficient spatial variation away from the surface. Although more general solutions exist, one solution is given by Eq. (26) in Sec. III.B.3.

3. Adiabatic and Isothermal Boundary

Because much of the work in verification by MMS is taken up in calculating the source term and incorporating it into the code, the amount of work can be reduced if the manufactured solution is designed to test multiple BCs. The following solution satisfies both the adiabatic and isothermal conditions on the surface $F = C_{a/i}$:

$$T = T_c[1 + g(x, y, z)(C_{a/i} - F)^m] \tag{26}$$

where $m > 1$ and $g(x, y, z) \geq 0$. Note that this is a departure from the general form in Eq. (17). The term $(C_{a/i} - F)^m$ ensures that $T = T_c$ and $\partial T/\partial n = 0$ on $F = C_{a/i}$. The function $g(x, y, z)$ is necessary in

the event that the function $F(x, y, z)$ does not have enough nonzero partial derivatives to exercise all of the terms in the equations. For example, if $F(x, y, z) = 1/2\cos(2x)\cos(3y) - z$, then g must be a function of z so that the leading error terms for the discretization of $\partial^2 T/\partial z^2$ are nontrivial. If $g(z) = 1 + 1/4\sin(z)$, this condition is satisfied. Alternatively, m could simply be increased until T has sufficient variance in z, but that has the effect of raising T, making it more difficult to locate supersonic regions to test supersonic inflow and outflow conditions. Also, raising m would increase solution gradient components in all three directions, whereas the addition of $g(z)$ only increases the gradient component in the z direction. High solution gradients require higher mesh resolution (i.e., bigger meshes and more expensive computations) to reach the asymptotic regime. Note that some generality has been sacrificed by using a manufactured solution that is both adiabatic and isothermal on the no-slip boundary (i.e., $\nabla T = 0$ on $F = C_{a/i}$). Future work on this manufactured solution will address this problem.

C. Turbulence Variable(s) for the RANS Equations

For RANS cases, turbulence variable(s) also need to be defined at a solid surface. For the Spalart–Allmaras model, the following equation can be used

$$\tilde{\nu} = (C_{n-s} - F)\tilde{\nu}_0[1 + f(Ax)f(By)f(Cy)] \tag{27}$$

For any of the two-equation models (e.g., k–ϵ or k–ω), this same formula can be used for k. However, the definition of the other turbulence variable (ϵ or ω) may not be zero at the boundary, and so a slightly different form must be used.

IV. Inflow and Outflow Conditions for the Euler Equations

For the Euler equations, the number of incoming and outgoing characteristics at any given inflow or outflow surface is determined by the signs of the eigenvalues. Because the manufactured solution needs to match the BCs, the eigenvalues of the operator $\mathcal{D}$ on $\partial\Omega$ must conform to the signs dictating which BCs are matched (e.g., for an inflow surface, $v_n < 0$ and for an outflow surface, $v_n > 0$). Otherwise, the boundary can be over- or underspecified by the BCs imposed by the code. Each of these surfaces may be either subsonic ($M_n < 1$), supersonic ($M_n > 1$), or a mixture of the two. For example, if a surface is specified as an outflow surface, a Dirichlet condition on the pressure will be enforced on subsonic portions of the boundary with the remainder of the variables determined by the interior equations. All of the flow variables will be determined by the interior equations on a supersonic portion of the boundary. However, if $v_n < 0$ anywhere on this surface, then the necessary constraints arising from the three additional negative eigenvalues will be missing, and the problem will be ill-posed.

Because the previously defined function F used for defining the slip and no-slip surfaces cannot also be used for defining an inflow or outflow surface, a new function must be considered. Let $G = G(x, y, z)$ be a differentiable function from $\Re^3$ into $\Re$ such that $G = C$ defines a surface for any constant C. Once G has been defined, it can be used in the manufactured solution to ensure that given inflow and outflow conditions hold on G-constant surfaces.

A. Subsonic Outflow Condition

Because the previously derived manufactured solutions for u, v, w, and T leave only one degree of freedom for an Euler solution, an inflow condition cannot be tested because it has four or five incoming characteristics, depending on the Mach number. However, an outflow condition can be tested because it has one or no incoming characteristics, depending on the normal Mach number. The pressure profile can be fixed so that it is constant along a general surface $G(x, y, z) = C_o$.

$$p = p_c + (C_o - G)p_0[1 + f(Ax)f(By)f(Cz)] \tag{28}$$

It is also necessary that $v_n > 0$ and $M_n < 1$. However, these constraints are not very restrictive, and so they can be combined with the conditions used to define earlier versions of the velocity profiles. Because the solution is intended to be used for testing all previously mentioned BCs in addition to the outflow condition, w is still defined by Eq. (24). The resulting normal velocity is

$$v_n = \frac{\boldsymbol{v}\cdot\nabla G}{\|\nabla G\|} = \frac{[G_x - G_z(F_x/F_z)]u + [G_y - G_z(F_y/F_z)]v}{\|\nabla G\|} \tag{29}$$

If each of the two terms in the numerator are greater than zero, then their sum (and thus the normal velocity) is greater than zero. With the definitions $\mathcal{A}_1 \equiv G_x - G_z(F_x/F_z)$, $\mathcal{A}_2 \equiv G_y - G_z(F_y/F_z)$, $\mathcal{B}_1 \equiv \mathcal{A}_1/\sqrt{\mathcal{A}_1^2 + \mathcal{A}_2^2}$, and $\mathcal{B}_2 \equiv \mathcal{A}_2/\sqrt{\mathcal{A}_1^2 + \mathcal{A}_2^2}$, u and v can be defined as

$$u = (C_{n-s} - F)\mathcal{B}_1 u_0[1 + f(Ax)f(By)f(Cz)] \tag{30}$$

$$v = (C_{n-s} - F)\mathcal{B}_2 v_0[1 + f(Ax)f(By)f(Cz)] \tag{31}$$

This guarantees that the surface is an outflow surface ($v_n > 0$) if both $[1 + f(Ax)f(By)f(Cz)]$ and $(C_{n-s} - F)$ are positive. The only remaining requirement is that the flow be subsonic, which is true provided u_0 and v_0 are chosen appropriately. These definitions, however, are more complex than necessary. If the terms $\mathcal{B}_1$ and $\mathcal{B}_2$ have sufficient variance in x, y, and z, then u and v can be simplified to

$$u = (C_{n-s} - F)\mathcal{B}_1 u_0 \tag{32}$$

$$v = (C_{n-s} - F)\mathcal{B}_2 v_0 \tag{33}$$

B. Supersonic Outflow

Provided that u_0 and v_0 from Eqs. (32) and (33) and m and T_0 from Eq. (26) are defined appropriately, some regions will have $M_n > 1$, and others, $M_n < 1$. Thus different regions of the manufactured solution can be used to test the implementation of supersonic and subsonic outflow enforcement at the surface $G = C_o$. For the supersonic case, there is no mathematical BC enforced, and so there is no constraint upon the manufactured solution for p.

V. Computational Domains

The functions $F(x, y, z)$ and $G(x, y, z)$ are used to define some of the boundaries of the computational domain. A third function, $H(x, y, z)$, is also needed to construct a six-sided domain. The no-slip surface is the $F = 0$ surface, and the p-constant surface is the $G = 0$ surface. All other F-constant surfaces are slip surfaces. Definitions for F, G, and H are as follows:

$$F = \frac{1}{2}\cos(A_f x)\cos(B_f y) - z \tag{34}$$

$$G = x - \frac{1}{2}\cos(A_g y)\sin(B_g z) - \frac{\pi}{4} \tag{35}$$

$$H = -y \tag{36}$$

A simple definition of H is permissible because no specialized BCs are tested on H-constant surfaces, (i.e., Dirichlet BCs are enforced). The sufficiency of a simple definition for H depends upon the BCs being implemented generally in the spatial sense. If they are not, then something must be done to account for this lack of generality. For example, if a structured mesh code has six separate implementations of each BC corresponding to the six bounding surfaces of the computational domain, then each of the six implementations needs to be tested. If this manufactured solution is to be used for such a case,

then several orientations of the computational space (ξ, η, ζ) relative to the F, G, and H surfaces need to be used. This is not necessary for Premo, because it is an unstructured code with BCs implemented in a general manner.

The coefficients used in this work are $A_f = 1/2$, $B_f = 3/4$, $A_g = 1/4$, and $B_g = 1/2$, representing a reduction in the wave numbers by a factor of four from [3]. This reduction of curvature reduces the mesh resolution required to reach the asymptotic regime. If the observed order of accuracy is poor on coarser meshes, then coding mistakes can be identified and fixed before much computational time is invested in running MMS cases on finer meshes. Finer meshes can later be used to confirm the coarse-mesh results.

Earlier versions of the manufactured solution [3,4] had constant p and varying M_n along a curved surface which was used for testing of the outflow BC. The surface was defined by the function $G(x, y, z) = C_o$. This constant-p surface was necessary for testing the outflow BC for subsonic conditions ($M_n < 1$). Under subsonic conditions, data corresponding to one incoming characteristic (i.e., one negative eigenvalue) must be specified, and Premo, like many CFD codes, enforces a constant p on the outflow surface. However, because there is a more general case (nonconstant p) of the supersonic outflow condition than there is for the subsonic outflow condition, it is preferable to have the mesh boundary deviate from the $G = C_o$ (and thus constant p) surface in the supersonic region.

Testing the supersonic outflow boundary along a constant-p surface is insufficient because a more general case, nonconstant p, exists for this BC. In fact, a coding mistake was found which would have been missed by testing the supersonic outflow condition along a constant-p surface. One of the requirements of manufactured solution design is that the solution be useful for the most general case of code application [6]. The same principle applies to the definition of the mesh. The meshes used in [3,4] were, in effect, blind to this particular type of coding mistake. The mesh definition in the present work tests the most general cases: a subsonic, constant-p outflow boundary; a supersonic, variable-p outflow boundary; or the union of both into a mixed boundary.

To address the problem caused by using a constant-p solution on supersonic regions of the outflow boundary in [4], a new function $\tilde{G}$ was defined as follows:

$$\tilde{G} = \begin{cases} G & M_n \le 1 \\ G + \exp\left(\frac{1}{(M_n-2)^2-1}\right) & 1 < M_n < 2 \\ G + \exp(-1) & 2 \le M_n \end{cases} \tag{37}$$

This function was chosen because $\tilde{G} = G$ on the subsonic portions of the boundary (where constant p is desired), $\tilde{G} \neq G$ on the supersonic portions (where nonconstant p is desired), and $\tilde{G}$ is infinitely differentiable.

A six-sided domain can be defined by setting maximum and minimum values for F, $\tilde{G}$, and H. Figures 1–3 show the three domains used for testing. In each figure, the view on the left shows the volume mesh, whereas the view on the right shows the boundaries which were used for BC testing. Figure 1 shows the domain that was used for testing the slip condition along its $F_{\max}$ boundary with the Euler equations on its interior and the supersonic outflow condition along its $\tilde{G} = 0$ boundary with the Euler equations on its interior. It is defined in $(F, \tilde{G}, H)$ space as $-0.05 < F < 0$, $-0.05 < \tilde{G} < 0$, and $-0.025 < H < 0.025$. The domain shown in Fig. 2 was used to test a mixed subsonic and supersonic outflow condition along its $\tilde{G} = 0$ boundary with the Euler equations on the interior. This domain was also used to test the Euler, Navier–Stokes, and RANS interior equation sets with Dirichlet BCs before testing of the specialized BCs began. It is defined in $(F, \tilde{G}, H)$ space as $-2.655 < F < -2.605$, $-0.05 < \tilde{G} < 0$, and $-0.025 < H < 0.025$. The domain shown in Fig. 3 was used for testing the no-slip condition along its $F = 0$ boundary with the Navier–Stokes equations on its interior and the subsonic outflow condition along its $\tilde{G} = G = 0$ boundary with the Euler equations on its interior. It is defined in $(F, \tilde{G}, H)$ space as $-3.525 < F < -3.475$, $-0.05 < \tilde{G} < 0$, and $-0.025 < H < 0.025$. These domains are much smaller (in the physical space) than the domains used in [3,4]. The small domains were needed to produce convincingly asymptotic results for the observed order of accuracy without large computational cost.

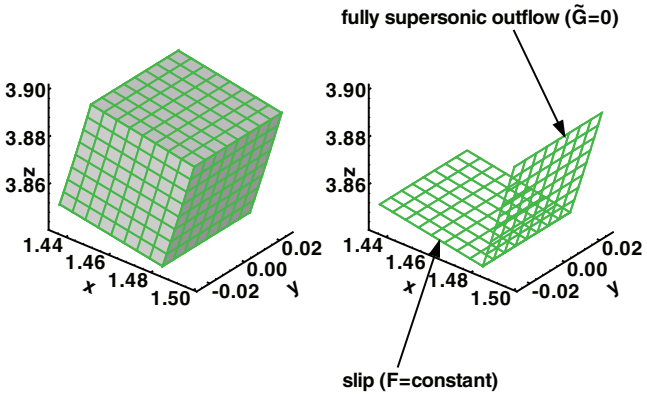

Fig. 1 Domain used for testing of slip condition and supersonic outflow.

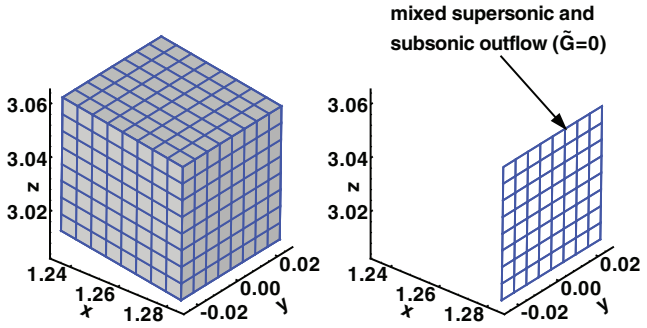

Fig. 2 Domain used for testing of mixed outflow.

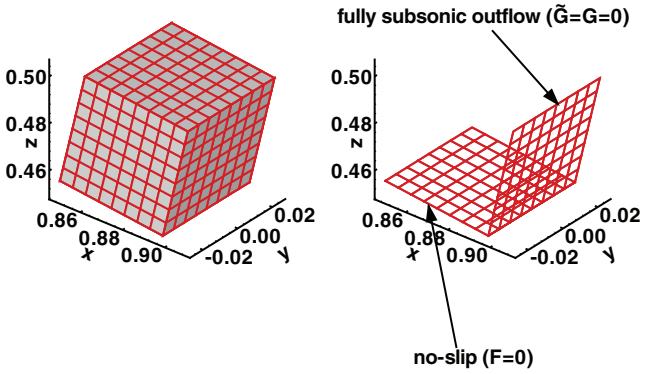

Fig. 3 Domain used for testing of no-slip condition and subsonic outflow.

The meshes on these computational domains are hexahedral and evenly spaced in $(F, \tilde{G}, H)$ space, with $N \in \{9, 17, 33, 65, 129\}$ nodes in each direction. This results in meshes which are nonuniform, skewed, and smooth in (x, y, z) space.

VI. Manufactured Solution

Once F, G, and H are defined, the manufactured solution can be defined as a function of F, G, and H. Because the manufactured solution is given in terms of G, and the meshes are defined in terms of $\tilde{G}$, the needed p variation is maintained for supersonic outflow tests. The definition of the manufactured solution is as follows:

$$\mathcal{A}_1 = G_x - G_z \frac{F_x}{F_z} \tag{38}$$

$$\mathcal{A}_2 = G_y - G_z \frac{F_y}{F_z} \tag{39}$$

$$\mathcal{B}_1 = \frac{\mathcal{A}_1}{\sqrt{\mathcal{A}_1^2 + \mathcal{A}_2^2}} \tag{40}$$

$$\mathcal{B}_2 = \frac{\mathcal{A}_2}{\sqrt{\mathcal{A}_1^2 + \mathcal{A}_2^2}} \tag{41}$$

$$u = \frac{1}{2}(-F)\mathcal{B}_1 a_{\mathrm{sl}} \tag{42}$$

$$v = \frac{1}{2}(-F)\mathcal{B}_2 a_{\mathrm{sl}} \tag{43}$$

$$w = -\frac{F_x u + F_y v}{F_z} \tag{44}$$

$$T = T_{\mathrm{sl}}\left(1 + \frac{1}{10}[1 + \sin(A_t z)]\right)F^2 \tag{45}$$

$$p = p_{\mathrm{sl}} - G p_{\mathrm{sl}}[1 + \cos(A_p x)\cos(B_p y)\cos(C_p z)] \tag{46}$$

$$\tilde{\nu} = -F\tilde{\nu}_\infty\left(1 + \frac{1}{2}\cos(A_\nu x)\cos(B_\nu y)\cos(C_\nu z)\right) \tag{47}$$

where:

$$a_{\mathrm{sl}} = 340 \text{ m/s} \tag{48}$$

$$p_{\mathrm{sl}} = 101.3 \text{ kPa} \tag{49}$$

$$T_{\mathrm{sl}} = 288 \text{ K} \tag{50}$$

$$\tilde{\nu}_\infty = 1.95 \times 10^7 \times \nu_{\mathrm{sl}} \tag{51}$$

$$\nu_{\mathrm{sl}} = 1.464 \times 10^{-5} \text{ m}^2/\text{s} \tag{52}$$

The coefficients used in the current work are $A_p = 3/4$, $B_p = 1/4$, $C_p = 1/2$, $A_t = 1$, $A_\nu = 3/4$, $B_\nu = 1/4$, and $C_\nu = 1/2$. As was the case in the definitions of F and G, this represents a reduction in most of the wave numbers from [3]. $\mathcal{B}_1$ and $\mathcal{B}_2$ are functions of F and G used to determine locations of inflow and outflow boundaries; a_{sl} is a reference speed of sound; p_{sl} is a reference pressure; T_{sl} is a reference temperature; ν_{sl} is a reference value for the kinematic viscosity. Changes to u, v, and A_t from [3,4] were made to control the locations of subsonic and supersonic regions.

For the effects of the Spalart–Allmaras transport equation to be significant on such a smooth solution, the reference value of $\tilde{\nu}$ was raised by seven orders of magnitude [Eq. (51)]. This is similar to the raising of the molecular viscosity in [4,10,11], which balanced the viscous and inviscid fluxes (a brief explanation is also given in Sec. VII.B). The intent of such term balancing is so that the numerical errors generated by each of the terms in the equations will be of approximately the same magnitude. If more physically realistic values are used in lieu of balancing the terms, then coding mistakes in small terms may be undetectable.

VII. Testing of Interior Equations

To break down the verification into manageable pieces, one can begin with the interior equations using only Dirichlet BCs. This allows the interior equation set to be verified first, so that any difference between the formal and observed orders of accuracy in the BC verification tests can be interpreted as a problem with the BC enforcement, rather than as a problem with the interior equation set. However, it is not necessary to verify every possible option for the interior equation set before progressing to the BCs. Examples of such options would be choice of auxiliary relations, such as the equation of state or transport property definitions. It is only necessary to verify in advance the same set of options which will be used in the BC tests.

A. Euler Equations

Bond et al. [3] showed tests of the Euler equations with Roe's [12] approximate Riemann solver using multiple gradient reconstruction techniques for extension to second order. Use of least-squares gradient reconstruction produced only first-order results. Because prior MMS verification [10,11] showed second-order behavior for least squares on uniform meshes, an explanation was sought for why first-order behavior was seen on nonuniform meshes. In [3], the authors erroneously concluded that this reduction in order was caused by improper choice of formulation: the use of equally weighted least squares (EWLS) as opposed to inverse-distance-weighted least squares (IDWLS). In fact, the correct explanation was simpler: a coding mistake was introduced in Premo during a revision between the two verification attempts. Because of this, a regression test (a test run frequently to ensure that code functionality is not degraded by future modifications) was added to the Premo test suite to prevent future recurrences. This is an important point for verification of codes in development. Once a particular set of options has been verified, regression tests should be implemented to prevent code modifications from changing the computational results obtained with this set of options.

After the coding mistake was corrected, tests for both equally weighted and inverse-distance-weighted gradient reconstruction were conducted. The results of these tests, contained in [4], showed behavior approaching second order for IDWLS. Although no evidence of first-order error was present for EWLS, the results were not sufficiently asymptotic to make a strong claim for second-order behavior.

The results of [3] showed behavior approaching second order when a particular approximate form of Green–Gauss (GG) gradient reconstruction was used. More recent results have shown that second-order behavior cannot be demonstrated with Premo's approximate form of GG gradient reconstruction on hexahedral element types when the meshes are sufficiently refined. With GG, the MMS results fail to iteratively converge when mesh resolution is increased beyond a certain threshold, rather than conclusively demonstrating a reduced order of accuracy. The specific issue with the approximate form of GG gradient reconstruction on hexahedral meshes is discussed in Sec. IX. This point calls into question the results of [3], and it is one of the reasons that the manufactured solution and associated meshes have been modified for the present work. Section X gives more details on this matter.

Results obtained with the current manufactured solution and sequence of meshes are given in Fig. 4. These results confirm second-order behavior for Roe's method using IDWLS gradient reconstruction. N is the number of nodes in each of the three mesh

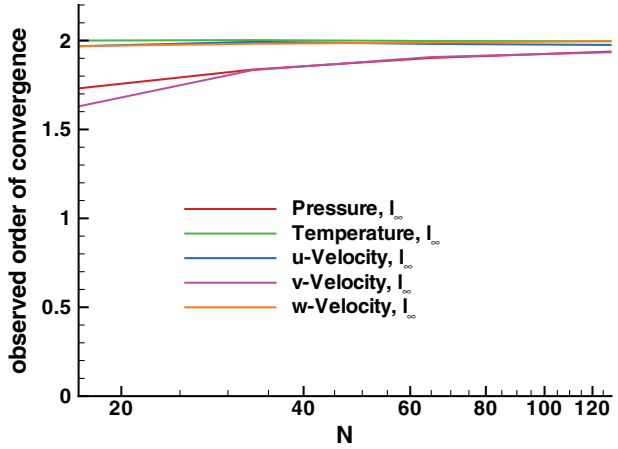

Fig. 4 Convergence order of l_∞ norms for the Euler equations with Roe's method.

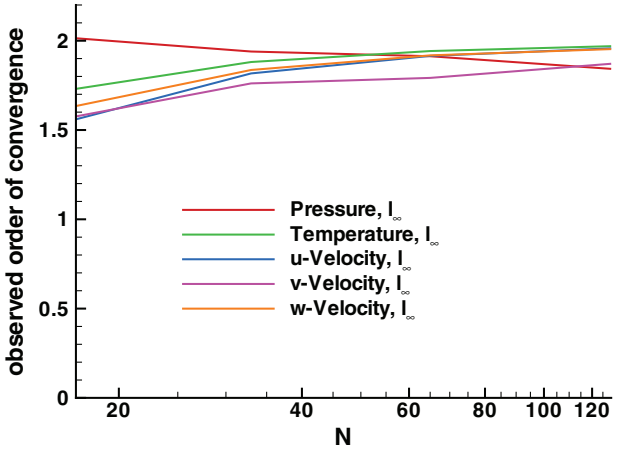

Fig. 5 Convergence order of l_∞ norms for the Navier–Stokes equations.

dimensions. For mesh refinement via doubling, the observed order of accuracy is given by the following equation:

$$\text{observed order} = \frac{\log(\|\epsilon_{N/2}\|/\|\epsilon_N\|)}{\log(2)} \tag{53}$$

where ϵ is the error (difference between discrete and exact solutions) for the flow variable of interest, and $\|\epsilon\|$ is any norm of that error. The convergence orders of the l_∞ norms are given for the errors in the flow variables p, T, u, v, and w. The l_∞ norm is used for the following reasons:

1) It tends to indicate nonconvergence on coarser meshes than the l_1 or l_2 norms.

2) The l_∞ norm generally requires finer meshes to produce convincingly asymptotic behavior for the observed order of accuracy, thus building more confidence that the asymptotic regime has in fact been reached if plots so indicate.

3) The l_∞ norm may find local sources of numerical error with order less than the global sources of numerical error if it converges to a value less than the converged values of the l_1 and l_2 norms ([4] gives examples where the values for the l_∞ norms differ from the l_1 and l_2 norms, and [13] gives a thorough theoretical discussion of why).

B. Navier–Stokes Equations

MMS verification of the Navier–Stokes equations can be performed on smaller meshes if values for the transport properties μ and k (the viscosity and thermal conductivity) cause the viscous fluxes and convective fluxes to be of similar magnitude. If the viscous fluxes are significantly smaller than the convective fluxes, coding mistakes present in viscous momentum or thermal flux calculations may not be observable by MMS tests except on extremely fine meshes.

Because Sutherland's law [$\mu = C_1 T^{1.5}/(T + C_2)$] and constant Prandtl number ($Pr = \mu C_p/k$, where C_p is the specific heat at constant pressure) were chosen as the options of highest priority for verification, the constants in Sutherland's law and the value of the constant Prandtl number were changed to allow the viscous momentum and thermal fluxes to be large enough that they were significant compared with the convective fluxes. The value used for C_1 was 100, whereas a typical value for air is 1.458×10^{-6}. For C_2, the typical value for air of 110.4 was used. This effectively raised the viscosity by eight orders of magnitude for the Navier–Stokes tests, making them highly sensitive to coding mistakes in the viscous flux calculations. A Prandtl number of one was used so that the thermal fluxes would be of the same order of magnitude as the viscous momentum fluxes.

Initial MMS tests using the Navier–Stokes equations revealed the presence of a coding mistake. Once this mistake was corrected, results from [4] showed second-order accuracy for the l_1 and l_2 norms of the errors in all of the variables, as well as the l_∞ norms of the errors of most of the variables. However, the order of the l_∞ norm of the p error appeared to be asymptotically approaching 1.5 when GG gradient reconstruction was used.

Results using IDWLS gradient reconstruction on the current sequence of meshes are shown in Fig. 5. These results demonstrate behavior in the vicinity of second order for all of the flow variables, in contrast to the results of [4]. This improvement is likely the result of switching from Premo's approximate form of GG gradient reconstruction to IDWLS gradient reconstruction. However, even with IDWLS gradients, the order of the l_∞ norm of the p error is trending downward on the finer meshes. Figure 6 does not show the same trend in the l_2 norms, indicating that any potential source of error of less than second order is confined to a local area that does not scale with the volume. In this specific case, it is most likely the influence of the gradient calculation on the viscous fluxes. The accuracy of the viscous fluxes can not exceed that of the gradients on which they depend, and the IDWLS gradient calculation drops to first order at the boundaries. The effect of this error term is very small, because the boundary gradient formulation provides a very small contribution to the viscous flux. The edge gradient used for the viscous flux has three components: half of the IDWLS gradient at one end point, half of the IDWLS gradient at the other end point, and a correction similar to the Computational Hydrodynamics for Advanced Design (CHAD) code [14] and also [15,16] which replace the gradient component along the edge with the directional derivative to avoid odd/even decoupling. For a boundary-terminated edge, the IDWLS gradient at the boundary node is only first order, but the CHAD correction dominates on smooth, hexahedral meshes, and so the influence of the first-order error upon the viscous flux is

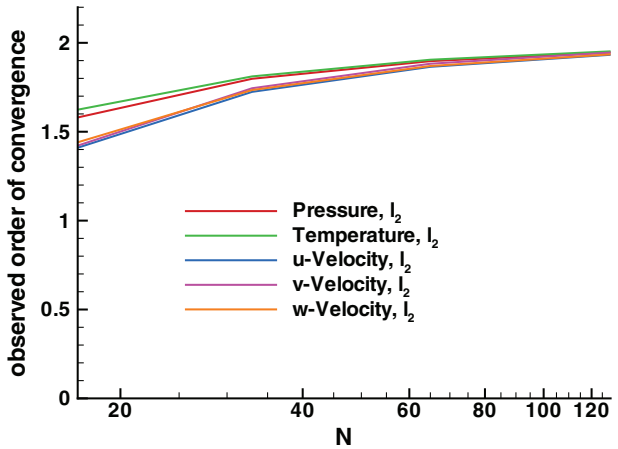

Fig. 6 Convergence order of l_2 norms for the Navier–Stokes equations.

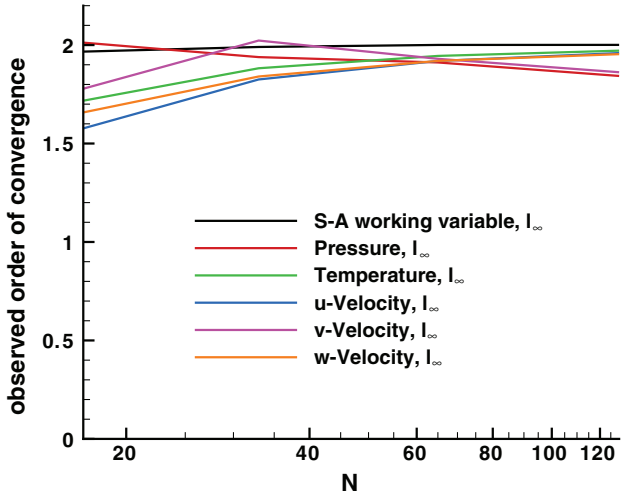

Fig. 7 Convergence order of l_∞ norms for the RANS equations.

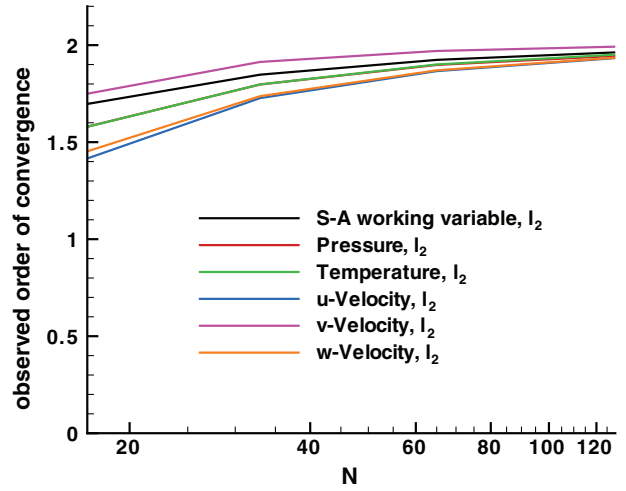

Fig. 8 Convergence order of l_2 norms for the RANS equations.

only detectable at extremely high resolution. Section IX gives more details on the gradient issue.

C. RANS Equations

For the RANS equation tests, C_1 in Sutherland's law was set to one to better balance the terms in the equations. Also, $Pr_t = 0.9$ was used for the turbulent Prandtl number, $\tilde{\nu}_\infty = 1.95 \times 10^7 \times \nu_{sl}$ was used as the reference value for the Spalart–Allmaras working variable, and $\kappa = 1 \times 10^6$ was used for the Kármán constant (the standard value for κ is 0.41).

Figure 7 shows the observed order of accuracy of the l_∞ norms of the errors in each of the flow variables for the RANS equations using IDWLS gradient reconstruction. These results show behavior in the vicinity of second order for the interior equations, although the orders of the l_∞ norms of the errors in p and v are trending downward. Figure 8 does not show the same downward trend for the l_2 norms. The most likely explanation involves the gradient calculations near the boundary, and it is discussed in Sec. VII.B.

VIII. Testing of Boundary Conditions

Once a set of governing equations has been verified on the interior of a domain, the same code options and domain can be used to test a given BC by using it on the appropriate surface instead of the Dirichlet BC. The BC can only be verified to an order equal to or less than the order observed using the same interior equation set with Dirichlet BCs applied. Note that some BCs in many CFD codes with second-order interior schemes (including Premo) should be expected to generate first-order error. Whether or not this error remains local to the boundary or propagates into the interior depends upon the mathematical character at the boundary [17]. If first-order error is generated locally at a boundary and does not propagate into the domain, then it is possible for the l_1 or l_2 norms of the error to show second-order convergence when the error calculation is performed over the entire domain [13]. This is true for the l_1 and l_2 norms because the boundary nodes compose a smaller fraction of the total number of nodes as the mesh is refined. On the other hand, if the l_∞ norm is used, or if the calculations for the l_1 or l_2 norms are restricted to the boundary region generating the first-order error, then the first-order error should be detected. Because of this, it is sometimes necessary to look at different error norms over different subsections of the domain to fully characterize the order behavior of a given BC implementation.

The Euler BCs testable by the current manufactured solution are the slip condition and outflow condition with the normal flow subsonic, supersonic, or mixed. The original slip condition implementation was tested in [3,4]. Results with a new implementation are contained in Sec. VIII.A. The original outflow condition implementation was tested for fully supersonic outflow in [3]. It was tested for transonic conditions and near-stagnation conditions in [4]. Results with a new implementation are contained in Sec. VIII.B.

A. Slip Results

Problems with the slip condition were observed in [3] wherever interprocessor boundaries intersected with the slip surface. (Some of the cases in each sequence of runs were done in parallel to verify that parallel coding mistakes were not overlooked.) In that particular case, a unit normal function valid for planar surfaces was used for the curved slip surface. A new function was written which was valid for curved surfaces as well. After this new function was properly referenced by the slip condition implementation, tests were repeated, and the results are given in [4]. Unfortunately, positive order of accuracy (i.e., consistency) was still not observed in the tests. It was eventually concluded that the problem was not caused by coding mistakes but instead was caused by a problem with the weak enforcement of the slip condition. This may be an example of an algorithm that is correctly implemented, but which has insufficient accuracy because of an inherent weakness. To resolve this issue, a new formulation was derived and implemented. The details of this formulation are presented in the Appendix. The basic difference is that the fully discretized conservation equations are changed into characteristic form for each boundary node. Hence the equations associated with incoming information are easily identified and discarded. The resulting reduced system of equations is closed by adding an appropriate number of BCs.

Figure 9 shows the observed order of convergence for the Euler equations with the slip condition enforced by the new implementation. Behavior approaching second order is observed, indicating that the newly implemented characteristic formulation eliminates the issues associated with the weak enforcement of the slip condition.

B. Outflow Results

Bond et al. [3–5] showed second-order behavior for the weakly enforced outflow condition when the outflow was supersonic ($M_n > 1$). However, the observed order of accuracy was nonasymptotic for tests with $M_n < 1$, even at high resolution, and less than two for some of the flow variables. A new enforcement strategy was implemented to remedy this problem. It is similar to the characteristic enforcement of the slip condition and is described in the Appendix.

Figure 10 shows the observed order of convergence for the Euler equations with the outflow condition enforced by the new implementation. This case has transonic outflow, with part of the outflow supersonic and part subsonic. Behavior approaching second order is observed, indicating that the newly implemented

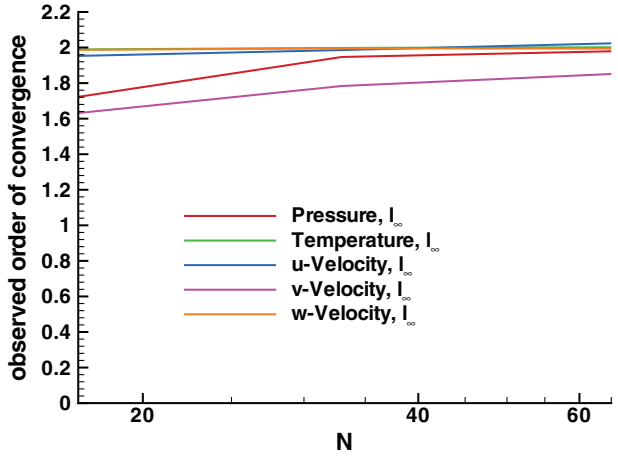

Fig. 9 Convergence order of l_∞ norms for the slip condition.

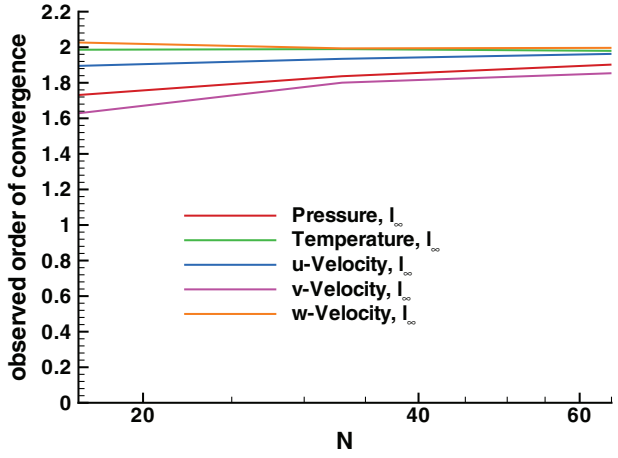

Fig. 10 Convergence order of l_∞ norms for the outflow boundary at transonic conditions.

characteristic formulation eliminates the issues associated with the weak enforcement at the outflow surface when $M_n < 1$.

C. No-Slip Results

The Navier–Stokes BCs testable by the current manufactured solution are the adiabatic and isothermal no-slip conditions. Results for tests of these cases are shown in [4,5]. The conclusions of [4] are that some first-order error is generated by these BCs and that this error does not propagate into the interior in detectable amounts. Updated results which cast doubt upon prior conclusions are discussed in [5]. No new results are shown here because the no-slip enforcement strategy in Premo has not been updated since the results in [5] were obtained.

IX. Testing of Gradient Calculations

Robustness issues with calculations of viscous boundary layers on hexahedral grids with high aspect ratios prompted an in-depth study of the gradient calculations. The questions raised were whether or not the gradient reconstruction options in Premo were performing at their desired orders of accuracy, whether or not accuracy was maintained for increasing curvature and aspect ratio, and what the magnitudes of the gradient errors were. For all of these tests, the manufactured solution was given to Premo as an initial condition, and the numerical gradients of that solution were written to a file. Figure 11 shows the order of accuracy of the gradient calculations using IDWLS. Fifty-four curves (6 variables × 3 components × 3 norms) are on this plot, making it difficult to distinguish them, but the intent is to demonstrate that all of the l_1, l_2, and l_∞ norms asymptotically approach 2.0, 1.5, and 1.0, respectively. This indicates that the interior gradient

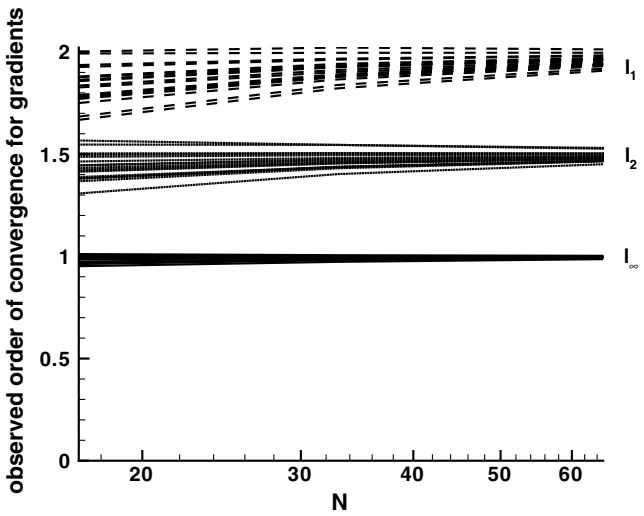

Fig. 11 Convergence order for the gradient calculation using IDWLS.

calculations are second order and that the boundary gradient calculations are first order. (The 1.5 convergence order for the l_2 norms does not indicate order 1.5 error; rather, it is the result of how the weights of interior and boundary nodes sum as the mesh is refined.) This is the expected order behavior for this algorithm. Even though virtual edges [16] are used to support the IDWLS stencil at boundaries, the error is still first order at all nodes on boundary faces, boundary edges, and boundary corners. This observed order behavior is preserved as mesh curvature and aspect ratio are increased, although finer meshes are necessary to reach the asymptotic regime.

The performance of the IDWLS gradient calculations was initially thought to be satisfactory, based on these order analyses. However, inspection of the magnitude of the error on the boundaries indicated serious problems, especially on corners or edges of the domain, where the stencil is one-sided in two or three dimensions. Even for the smooth manufactured solution at high resolution, the gradients were seen to have high relative error and occasionally even the wrong sign at boundaries.

Figure 12 shows the same 54 curves for the gradient calculations using a particular approximate form of GG. Obviously, severe problems are indicated in the order behavior. Premo's implementation of GG gradient calculations is only strictly valid for tetrahedral element types because it neglects some additional support necessary for non-Cartesian, hexahedral element types. Because of this approximation, poor order performance is not

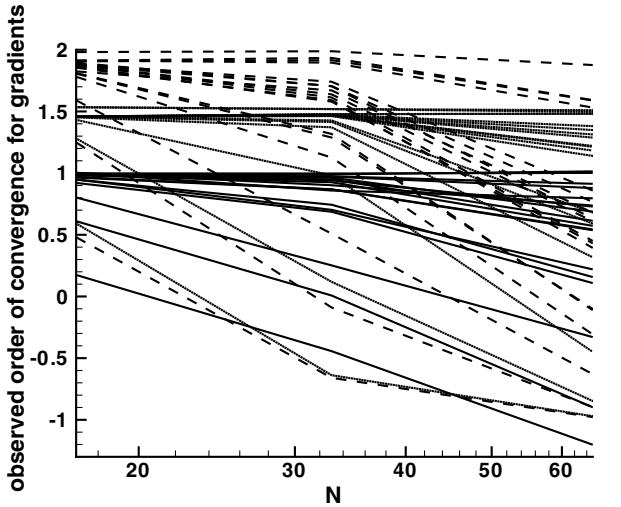

Fig. 12 Convergence order for the gradient calculation using an approximate form of GG.

surprising; however, the severity of this problem (negative observed order is evident in some of the curves) even on smooth meshes was underestimated. The use of this approximation to the GG gradient calculation is believed to be the source of several of the anomalies documented in [4]. Mavriplis [18], Aftosmis et al. [19], and Barth and Jesperson [20] compare the formulations of these gradient reconstruction schemes and their performance on more complex mesh topologies.

The gradient calculations in Premo are currently being revised in an attempt to provide increased accuracy and robustness. Viscous flux calculations are particularly sensitive to the accuracy of the gradients. Also, overprediction of the gradients artificially increases turbulent production, leading to a catastrophic loss of robustness.

X. Demonstrating Asymptotic Behavior

The authors believe that asymptotic behavior of observed order of accuracy should be demonstrated in all order-of-accuracy verification exercises. A common but questionable practice is to plot error norms vs mesh size on a logarithmic scale and then compare them with a line (parallel, in the "eyeball" norm) representing a particular integer order of accuracy. Quantitative order-of-accuracy plots or tables are far more likely to highlight problems with a code's order behavior. Another questionable practice is to quote only l_1 or l_2 norms, even though the l_∞ norms are much more sensitive to problems.

Bond et al. [3,4], and the current work, show an evolution toward lower curvature manufactured solutions and smaller domains (in the physical space) in an effort to produce more convincingly asymptotic results at reasonable computational cost. However, this evolution has demonstrated that results claimed to represent second-order behavior in [3,4] did not have sufficient resolution to uncover sources of error of less than second order (e.g., second-order accuracy was claimed for some cases where Premo's approximate implementation of GG gradients was used). This indicated that some criterion other than asymptotic order behavior needed to be specified. As demonstrated in the following example, the relative error should be noted as part of the order-verification exercise.

As the manufactured solution and meshes evolved from those in [3,4] to the current work, the relative errors of some of the variables began to approach the single-precision limit (approximately 10^{-7}) and level off. Rather than stemming from a coding mistake in Premo, this problem was a result of the coupling between the g77 compiler and Mathematica®, the symbolic manipulation software used to generate the code for calculating the MMS source terms. The FORTRAN code output by Mathematica was compiled with g77 before linking with the remainder of the C++ Premo source. Mathematica writes constants out in single-precision format (e.g., 1.0) rather than double-precision format (e.g., 1.0d0), and the typecasting in g77 does not fill the four least significant bytes of a double-precision variable with zeros when a single-precision constant is used in an assignment statement. This resulted in a single-precision error in the routine for calculating the MMS source, even though the C++ Premo source was accurate to double precision. Once this problem was identified, a remedy allowed the MMS tests to be performed in double precision. Although no coding mistake regarding the handling of double precision floating point values was actually present in Premo, if there had been one, it would never have been detected if the mesh resolution of the MMS tests was not sufficient to allow relative errors of the flow variables to be below the single-precision limit.

All of the cases presented in this article have relative errors in all flow variables smaller than the single-precision limit, at least on the $N \geq 65$ meshes. The authors now use this as an additional constraint for claiming that a code has been verified to a given order and recommend this requirement for use by others as well (or something similar, depending upon the desired precision of the code). Otherwise, any claims to a given precision have not been substantiated, and any claims to a given order of accuracy are suspect for high values of relative error, even if the computed order of accuracy appears to be asymptotically approaching a given value (e.g., the GG results of [3]). This is especially important when the l_∞ norm is used, because it may not monotonically approach the observed order or get within a small tolerance of an integer-valued observed order except on extremely fine meshes. The sometimes erratic behavior of the l_∞ norm is a result of the location of the maximum error moving around as the mesh is refined. This makes it an unpopular choice for order verification, but it is preferred by the authors for the reasons discussed in Sec. VII.A.

XI. Conclusions

A three-dimensional manufactured solution has been derived that is capable of testing the Euler, Navier–Stokes, and RANS equations on nonuniform meshes along with the slip and no-slip (adiabatic and isothermal) conditions and outflow (subsonic, supersonic, and mixed) boundary. For Premo, use of this manufactured solution has identified coding mistakes that prevented initial results from confirming the formal order of accuracy in some cases. Use of this manufactured solution has also identified or clarified a number of formulation weaknesses with BCs and gradient reconstruction methods.

The derived manufactured solution is applicable not only to Premo, but also to general CFD codes. Additionally, the methodology for deriving manufactured solutions for specialized BCs used in CFD codes has been shown useful, and a similar approach could be used to derive a manufactured solution for other BCs (e.g., inflow boundaries). Unlike the previous versions of the manufactured solution and sequence of meshes given in [3,4], the current manufactured solution and sequence of meshes are mature enough for other code developers to use for order-of-accuracy verification. It is hoped that such use will not only allow other code developers to find and correct coding mistakes in their own software, but also evaluate different formulations of BCs. Formal order-of-accuracy verification is often heralded for its usefulness in identifying coding mistakes, but its ability to identify weaknesses in numerical formulations and algorithms themselves is as, if not more, significant. The generality provided by using manufactured solutions for formal order-of-accuracy verification allows very rigorous testing of BC formulations. Many BC formulations found in the literature contain approximations of unspecified order of accuracy that are not obvious or well documented, and manufactured solutions derived specifically to evaluate BCs can be used to eliminate some formulations from consideration in the code design process.

Appendix: Characteristic Boundary Conditions

For inviscid applications, the governing equations are hyperbolic. For such a system of equations, information propagates at certain wave speeds along characteristic directions (e.g., [21]). It is well established that the method of characteristic projection leads to a set of approximately nonreflecting BCs that is well posed for hyperbolic systems. The motivation and detailed analysis for this procedure is available in the literature (e.g., [22–24]). In this section, we present implementation details that are unique to a three-dimensional, unstructured, implicit finite volume code. Our approach closely follows the development of Merkle [25], which was performed for structured grids.

On the boundaries, the control volume integration is closed by calculating the surface integral using the mathematical flux, as opposed to the numerical, dissipative flux. Let the resulting, linearized, fully discretized conservation equations for a boundary node be written as

$$J\delta \boldsymbol{U} = -\boldsymbol{r}(\boldsymbol{U}) \tag{A1}$$

where J is the linearized coefficient matrix (e.g., a Jacobian in a Newton iteration scheme), $\delta \boldsymbol{U}$ is the incremental change in the solution state variable, and $\boldsymbol{r}(\boldsymbol{U})$ is the residual. Next, using the chain rule, we introduce a transformation of dependent variable $\hat{\boldsymbol{U}}(\boldsymbol{U})$,

$$\delta \boldsymbol{U} = \frac{\partial \boldsymbol{U}}{\partial \hat{\boldsymbol{U}}} \delta \hat{\boldsymbol{U}} \tag{A2}$$

Let

$$M \equiv \frac{\partial \boldsymbol{U}}{\partial \hat{\boldsymbol{U}}}$$

Now, Eq. (A1) may be transformed into the new variables $\hat{\boldsymbol{U}}$ if it is premultiplied by the matrix M^{-1}, assuming M is invertible. Thus we may write the conservation equations for a boundary node as

$$M^{-1} J \delta \boldsymbol{U} = -M^{-1} \boldsymbol{r}(\boldsymbol{U}) \tag{A3}$$

We now define M to be the modal matrix of the convective flux vector oriented in the direction of the local outward normal vector, namely $\hat{\boldsymbol{n}}$ as defined in Eq. (22). In other words, M diagonalizes the matrix $A_{\hat{\boldsymbol{n}}}$, where

$$A_{\hat{\boldsymbol{n}}} = \frac{\partial \boldsymbol{\mathcal{F}}_{\hat{\boldsymbol{n}}}}{\partial \boldsymbol{U}} \tag{A4}$$

It is well known that this linear combination of flux Jacobians has the eigenvalues

$$\lambda_1 = v_n - a \tag{A5}$$

$$\lambda_{2,3,4} = v_n \tag{A6}$$

$$\lambda_5 = v_n + a \tag{A7}$$

Moreover, by virtue of the hyperbolic property of the Euler equations, the matrix also possesses a full set of linearly independent eigenvectors. Hence M is invertible, and M^{-1} may be written as

$$\begin{pmatrix} \frac{1}{2}\left(v_n a + \frac{1}{2}\hat{\gamma} q^2\right) a & -\frac{1}{2}(a n_x + u\hat{\gamma}) & -\frac{1}{2}(a n_y + v\hat{\gamma}) & -\frac{1}{2}(a n_z + w\hat{\gamma}) & \frac{1}{2}\hat{\gamma} \\ a^2 n_x + a(w n_y - v n_z) - \frac{1}{2} n_x \hat{\gamma} q^2 & \hat{\gamma} u n_x & a n_z + n_x \hat{\gamma} v & -a n_y + n_x \hat{\gamma} w & -\hat{\gamma} n_x \\ a^2 n_y + a(u n_z - w n_x) - \frac{1}{2} n_y \hat{\gamma} q^2 & -a n_z + \hat{\gamma} n_y u & \hat{\gamma} n_y v & a n_x + \hat{\gamma} n_y w & -n_y \hat{\gamma} \\ a^2 n_z + a(v n_x - u n_y) - \frac{1}{2} n_z \hat{\gamma} q^2 & a n_y + n_z \hat{\gamma} u & -a n_x + n_z \hat{\gamma} v & \hat{\gamma} n_z w & -n_z \hat{\gamma} \\ \frac{1}{2}\left(-a v_n + \frac{1}{2}\hat{\gamma} q^2\right) & \frac{1}{2}(a n_x - \hat{\gamma} u) & \frac{1}{2}(a n_y - \hat{\gamma} v) & \frac{1}{2}(a n_z - \hat{\gamma} w) & \frac{1}{2}\hat{\gamma} \end{pmatrix} \tag{A8}$$

where $\hat{\gamma} = \gamma - 1$.

Now that the system of Eqs. (A3) is written in terms of characteristic variables, each component may be identified as incoming or outgoing simply by the sign of its associated eigenvalue λ_i, $i = 1, \ldots, 5$. If $\lambda_i < 0$, then the ith component of the system (A3) is associated with incoming information, and it should be replaced with a BC. To handle this change in a systematic way, we introduce a diagonal selector matrix S, each element s_i of which is defined according to

$$s_i = \begin{cases} 1 & \text{if } \lambda_i \geq 0 \\ 0 & \text{if } \lambda_i < 0 \end{cases}$$

Now we may write the reduced system of outgoing conservation equations in characteristic form as

$$SM^{-1} J \delta \boldsymbol{U} = -SM^{-1} \boldsymbol{r}(\boldsymbol{U}) \tag{A9}$$

The reduced system (A9) must be augmented by BC equations to recover five equations in five unknowns. Let these conditions be represented by the vector

$$\boldsymbol{\mathcal{B}}(\boldsymbol{U}) = \boldsymbol{0} \tag{A10}$$

Upon application of the chain rule to Eq. (A10), we obtain

$$\frac{\partial \boldsymbol{\mathcal{B}}}{\partial \boldsymbol{U}} \delta \boldsymbol{U} = \boldsymbol{0} \tag{A11}$$

Equation (A11) may be applied at a boundary node if the BC does not change in time. In practice, one may accommodate time-varying BCs via the relation

$$\frac{\partial \boldsymbol{\mathcal{B}}}{\partial \boldsymbol{U}} \delta \boldsymbol{U} = \delta \boldsymbol{b} \tag{A12}$$

where $\delta \boldsymbol{b}$ is the local difference between value of the quantity that is specified at the boundary and the current value.

By construction, the system of equations in Eq. (A12) has nonzero entries in complementary positions compared with the system (A9). We may therefore add them together to obtain a single system of equations for a boundary node which may be written compactly as

$$\left[\frac{\partial \boldsymbol{\mathcal{B}}}{\partial \boldsymbol{U}} + SM^{-1} J\right] \delta \boldsymbol{U} = \delta \boldsymbol{b} - SM^{-1} \boldsymbol{r}(\boldsymbol{U}) \tag{A13}$$

To solidify ideas, consider a subsonic inflow. In this case, only $\lambda_1 < 0$, and we specify the back pressure p_b. Therefore, $S = \mathrm{diag}(0, 1, 1, 1, 1)$, and

$$\boldsymbol{\mathcal{B}}(\boldsymbol{U}) = \begin{pmatrix} p - p_b \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{A14}$$

Upon differentiation, Eq. (A14) becomes

$$\frac{\partial \boldsymbol{\mathcal{B}}}{\partial \boldsymbol{U}} = \begin{pmatrix} \frac{\hat{\gamma} v^2}{2} & -\hat{\gamma} u & -\hat{\gamma} v & -\hat{\gamma} w & \hat{\gamma} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{A15}$$

The situation is similar at a slip condition, where $v_n = 0$. Kentzer [26] observed that although λ_2, λ_3, and λ_4 vanish, the fact that only $\lambda_1 < 0$ implies that only a single BC is needed. This BC is, of course, $v_n = 0$. Therefore

$$\boldsymbol{\mathcal{B}}(\boldsymbol{U}) = \begin{pmatrix} v_n \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{A16}$$

Upon differentiation, Eq. (A16) becomes

$$\frac{\partial \boldsymbol{\mathcal{B}}}{\partial \boldsymbol{U}} = \begin{pmatrix} -v_n/\rho & n_x/\rho & n_y/\rho & n_z/\rho & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{A17}$$

We remark that when the fluid viscosity is nonnegligible, the system of equations becomes incompletely parabolic and the inviscid theory is no longer formally applicable. However, in practice, the same method of enforcing the BCs may, and often is, still used at inflow/outflow boundaries, especially if the boundary is located in a region where the viscous stresses are relatively unimportant.

Acknowledgments

This work is supported by Sandia National Laboratories and the U.S. Department of Energy. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed–Martin company, for the U.S. Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000. The authors would like to thank Alfred Lorber and Thomas Smith for their help in implementing the manufactured solutions in Premo.

References

[1] Smith, T. M., Ober, C. C., and Lorber, A. A., "SIERRA/Premo: An Unstructured Mesh Compressible Flow Simulation Code," AIAA Paper 2002-3292, June 2002.

[2] Edwards, H. C., and Stewart, J. R., "SIERRA : A Software Environment for Developing Complex Multiphysics Applications," *Proceedings of First MIT Conference on Computational Fluid and Solid Mechanics*, edited by K. J. Bathe, Elsevier, New York, June 2001.

[3] Bond, R. B., Knupp, P. M., and Ober, C. C., "Manufactured Solution for Verifying CFD Boundary Conditions," AIAA Paper 2004-2629, June 2004.

[4] Bond, R. B., Knupp, P. M., and Ober, C. C., "Manufactured Solution for Verifying CFD Boundary Conditions, Part 2," AIAA Paper 2005-0088, Jan. 2005.

[5] Bond, R. B., Ober, C. C., Knupp, P. M., "Manufactured Solution for Verifying CFD Boundary Conditions, Part 3," AIAA Paper 2006-3722, June 2006.

[6] Knupp, P., and Salari, K., *Verification of Computer Codes in Computational Science and Engineering*, Chapman and Hall/CRC, Boca Raton, FL, 2003.

[7] Roache, P. J., *Verification and Validation in Computational Science and Engineering*, Hermosa, Albuquerque, NM, 1998.

[8] Spalart, P. R., and Allmaras, S. R., "One-Equation Turbulence Model for Aerodynamic Flows," *La Recherche Aerospatiale: Bulletin Bimestriel de l'Office National d'Etudes et de Recherches Aerospatiales*, No. 1, 1994, pp. 5–21.

[9] Strauss, W. A., *Partial Differential Equations: an Introduction*, Wiley, New York, 1992, pp. 82–86.

[10] Roy, C. J., Smith, T. M., and Ober, C. C., "Verification of a Compressible CFD Code Using the Method of Manufactured Solutions," AIAA Paper 2002-3110, June 2002.

[11] Roy, C. J., Nelson, C. C., Smith, T. M., and Ober, C. C., "Verification of Euler/Navier–Stokes Codes Using the Method of Manufactured Solutions," *International Journal for Numerical Methods in Fluids*, Vol. 44, No. 6, 2004, pp. 599–620.

[12] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, Oct. 1981, pp. 357–372.

[13] Gustafsson, B., "Convergence Rate for Difference Approximations to Mixed Initial Boundary Value Problems," *Mathematics of Computation* Vol. 29, No. 130, 1975, pp. 396–406.

[14] O'Rourke, P. J., Zhang, S., and Sahota, M. S., "Parallel, Unstructured-Mesh Methodology for Device-Scale Combustion Calculations," *Oil and Gas Science and Technology Review*, Vol. 54, No. 2, 1999, pp. 169–173.

[15] Weiss, J. M., Maruszewski, J. P., and Smith, W. A., "Implicit Solution of Preconditioned Navier–Stokes Equations Using Algebraic Multigrid," *AIAA Journal*, Vol. 37, No. 1, 1999, pp. 29–36.

[16] Haselbacher, A., and Blazek, J., "On the Accurate and Efficient Discretization of the Navier–Stokes Equations on Mixed Grids," *AIAA Journal*, Vol. 38, No. 116, 2000, pp. 2094–2102; also AIAA Paper 99-3363, 1999.

[17] Roy, C. J., "Grid Convergence Error Analysis for Mixed-Order Numerical Schemes," *AIAA Journal*, Vol. 41, No. 4, 2003, pp. 595–604.

[18] Mavriplis, D. J., "Revisiting the Least-Squares Procedure for Gradient Reconstruction on Unstructured Meshes," AIAA Paper 2003-3986, June 2003.

[19] Aftosmis, M., Gaitonde, D., and Tavares, T. S., "Behavior of Linear Reconstruction Techniques on Unstructured Meshes," *AIAA Journal*, Vol. 33, No. 11, Nov. 1995, pp. 2038–2049.

[20] Barth, T. J., and Jesperson, D. C., "Design and Application of Upwind Schemes on Unstructured Meshes," AIAA Paper 89-0366, Jan. 1989.

[21] LeVeque, R. J., *Numerical Methods for Conservation Laws*, Birkhäuser–Verlag, Boston, 1992.

[22] Engquist, B., and Majda, A., "Absorbing Boundary Conditions for the Numerical Simulation of Waves," *Mathematics of Computation*, Vol. 31, No. 139, 1977, pp. 629–651.

[23] Hedstrom, G. W., "Nonreflecting Boundary Conditions for Nonlinear Hyperbolic Systems," *Journal of Computational Physics*, Vol. 30, No. 2, 1979, pp. 222–237.

[24] Giles, M. B., "Nonreflecting Boundary Conditions for Euler Equation Calculations, *AIAA Journal*, Vol. 28, No. 12, 1990, pp. 2050–2058.

[25] Merkle, C. L., "Time Iterative Methods in Computational Fluid Dynamics," *Lecture Notes*, Purdue Univ. Press, West Lafayette, IN, 2006.

[26] Kentzer, C. P., "Discretization of Boundary Conditions on Moving Discontinuities," *Lecture Notes in Physics*, Vol. 8, Springer–Verlag, New York, 1970, pp. 108–113.

Z. Wang
Associate Editor