AIAA 2004–2629

A Manufactured Solution for Verifying CFD Boundary Conditions

Ryan B. Bond, Patrick M. Knupp, and Curtis C. Ober
Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185

34th AIAA Fluid Dynamics Conference and Exhibit
28 June–1 July, 2004
Portland, Oregon

For permission to copy or republish, contact the copyright owner named on the £rst page
For AIAA-held copyright, write to AIAA Permissions Department,
1801 Alexander Bell Drive, Suite 500, Reston, VA 20191–4344

A Manufactured Solution for Verifying CFD Boundary Conditions

Ryan B. Bond,* Patrick M. Knupp,† and Curtis C. Ober‡
Sandia National Laboratories§
P. O. Box 5800
Albuquerque, NM 87185

Formal order of accuracy verification is necessary to ensure that software correctly and accurately solves a given set of equations. A manufactured solution has been derived and implemented that allows verification of not only the Euler, laminar Navier-Stokes, and Reynolds-Averaged Navier-Stokes (RANS) equation sets, but also some of their associated boundary conditions (BC's): slip, no-slip (adiabatic and isothermal), and outflow (subsonic and supersonic). Tests have been completed for the Euler equation set with both Green-Gauss and least-squares gradient reconstruction as well as the slip and supersonic outflow BC's. All tests were performed on non-uniform, hexahedral meshes. The Euler equations with Green-Gauss gradient reconstruction verified to second-order, as did the supersonic outflow BC. The Euler equations with least-squares gradient reconstruction using equal weighting verified only to first-order, but future tests will be performed with inverse distance weighting since unweighted least-squares on non-uniform meshes has been previously documented as having inaccuracies. The results of the slip BC tests identified a problem associated with the slip BC and the parallel decomposition of the domain. These initial tests demonstrated the ability of the method of manufactured solutions (MMS) to provide order verification of specialized CFD boundary conditions.

Introduction

The purpose of this work is to derive and implement a manufactured solution capable of testing computational fluid dynamics (CFD) codes' governing equation sets and boundary conditions on non-uniform meshes. Although the implementation is done for Premo,[1] the derivation respects the needs of general CFD codes and should be useful for their verification. Premo is a compressible flow code used to determine aerodynamic performance of arbitrary bodies, and it is built within the SIERRA framework[2] to allow multi-physics coupling (*e.g.*, ablation and aeroelasticity). It is a parallel, unstructured, edge-based, finite-volume code.

The method of manufactured solutions (MMS) provides formal order-of-accuracy verification of the implemented equation sets along with their various boundary conditions. Such verification is needed to ensure the correctness and accuracy of the software and to build confidence within the community of users. MMS derives its name from a solution which is manufactured for the purpose of verifying the code. It is similar to the method of exact solutions, where discrete solutions on a sequence of systematically refined meshes are compared to an exact solution. Manufactured solutions are used where no exact solutions exist or where exact solutions are not sufficiently general to fully test a code. A manufactured solution does not actually need to satisfy the original governing equation set. Instead, the equation set is modified through the addition of a source term so that the manufactured solution is an exact solution to the modified equation set. The exact form of the source term depends on the particular 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 in order to balance it. If lines for calculating the source term are added to the code, it can be used to solve the modified equation set instead of the original equation set. (The mechanism for this in Premo is a user-defined source routine.) Then, the discrete solutions produced by the code can be compared to the exact manufactured solution in order to determine the discretization error. A comparison of the discretization error on a series of systematically refined meshes will either verify that the observed order of accuracy matches the expected order of accuracy, or it will indicate that the observed order of accuracy lags the expected order of accuracy. The latter generally means that a bug in the code is reducing the observed order of accuracy, although it could also indicate that the expected order of accuracy is unattainable with

*Aerosciences and Compressible Fluid Mechanics Department, MS 0825, E-mail: rbbond@sandia.gov. Member, AIAA.

†Discrete Algorithms and Math Department, MS 1111, E-mail: pknupp@sandia.gov.

‡Computational Sciences Department, MS 0316, E-mail: ccober@sandia.gov. Senior Member, AIAA.

§Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

This paper is a work of the U.S. Government and is not subject to copyright protection in the United States.2004

the chosen formulation (bugs or no bugs) because the expected order was derived incorrectly.

The Reynolds-Averaged Navier-Stokes (RANS) equations are given as

$$\frac{\partial(\overline{\rho})}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{u_j})}{\partial x_j} = 0 \tag{1}$$

$$\frac{\partial(\overline{\rho}\widetilde{u_i})}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{u_i}\widetilde{u_j} + \overline{p}\delta_{ij} - \overline{\tau_{ij}})}{\partial x_j} = 0 \tag{2}$$

$$\frac{\partial(\overline{\rho}\widetilde{e_t})}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{u_j}\widetilde{e_t} + \overline{p}\widetilde{u_j} - \widetilde{u_i}\overline{\tau_{ij}} + \overline{q_j})}{\partial x_j} = 0, \tag{3}$$

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

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

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

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

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

The above relations reference the ratio of specific heats (γ) and the specific gas constant (R). For air, values for these quantities are 1.4 and 287.0 $\frac{\text{N m}}{\text{kg K}}$, respectively.

The viscous stress tensor and heat flux vector are given by:

$$\overline{\tau_{ij}} = (\overline{\mu} + \mu_t)\left[\left(\frac{\partial\widetilde{u_i}}{\partial x_j} + \frac{\partial\widetilde{u_j}}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}\frac{\partial\widetilde{u_k}}{\partial x_k}\right] \tag{7}$$

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

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

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^{\frac{3}{2}} \frac{T_0 + S}{T + S}; \tag{9}$$

while 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[3] was used. In compressible form, the single transport equation for the model can be expressed as:

$$\frac{\partial(\overline{\rho}\widetilde{\nu})}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{u_j}\widetilde{\nu})}{\partial x_j} = D + S_P - S_D, \tag{12}$$

where

$$S_P = c_{b1}\overline{\rho}\widetilde{\nu}\widetilde{S} \tag{13}$$

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

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

The closure coefficients and auxiliary relations are given by

$$c_{b1} = 0.1355$$

$$c_{b2} = 0.622$$

$$c_{\nu 1} = 7.1$$

$$\sigma = 2/3$$

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

$$c_{\omega 2} = 0.3$$

$$c_{\omega 3} = 2$$

$$\kappa = 0.41$$

$$f_{\nu 1} = \frac{\chi^3}{\chi^3 + c_{\nu 1}^3}$$

$$f_{\nu 2} = 1 - \frac{\chi}{1 + \chi f_{\nu 1}}$$

$$f_\omega = g\left[\frac{1 + c_{\omega 3}^6}{g^6 + c_{\omega 3}^2}\right]^{\frac{1}{6}}$$

$$\chi = \frac{\widetilde{\nu}}{\nu}$$

$$g = r + c_{\omega 2}\left(r^6 - r\right)$$

$$r = \frac{\widetilde{\nu}}{\widetilde{S}\kappa^2 d^2}$$

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

The eddy viscosity is given by

$$\mu_t = \overline{\rho}\widetilde{\nu} f_{\nu 1}. \tag{16}$$

(Note that $\widetilde{\nu}$ is a working variable, not the Favre averaged kinematic viscosity.)

To manufacture a solution to Eq. (1), (2), (3), (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 boundary condition being tested. For this work, the variables p, T, u, v, w, and $\widetilde{\nu}$ were used. This manufactured solution may also be used for the laminar Navier-Stokes set or the Euler set by simply dropping the extraneous turbulence variable.

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

$$\phi = \phi_0 \left(1 + f(Ax)f(By)f(Cz)\right), \tag{17}$$

where A, B, and C are constants. The function f must be chosen so that this form satisfies the requirements of Knupp and Salari.[4] To prevent symmetry within the solution, $A \neq B \neq C \neq 0$. This form is useful in testing boundary conditions because it is a separated solution plus a constant.[5] The separated form makes deriving manufactured solutions with specific Neumann boundary conditions (such as a vanishing normal derivative) easier. This particular form is not sufficient for all of the boundary conditions of interest, but it provides a prototype which can be modified when necessary.

"Physical" versus "Numerical" Boundary Conditions

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 orientation 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. Conditions enforced on the boundaries of the domain are often categorized as "physical" or "numerical" boundary conditions.[6] The term "physical boundary condition" refers to a condition which is imposed on the solution at the boundary as a consequence of information propagating into the interior of the domain from the boundary. The term "numerical boundary condition" refers to a condition which holds true at the boundary as a consequence of information propagating to the boundary from the interior of the domain. The eigenvalues of a system of hyperbolic or mixed character equations determine the directions in which the information will propagate. For the 3-D 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 (this eigenvalue is repeated three times). The number of "physical boundary conditions" is equal to the number of negative eigenvalues. For a supersonic outflow condition, there are no negative eigenvalues; for a subsonic outflow or slip boundary, there is one; for a subsonic inflow, there are are four, and for a supersonic inflow, there are five.

It is important to remember, however, that "numerical boundary conditions" are not really boundary conditions at all in the mathematical sense. Instead, they are consequences of the governing equation set which hold as the boundary is approached from the interior. From an implementation standpoint, this distinction is usually lost, but setting boundary data according to "numerical boundary conditions" is just a special way of solving the governing equations at the boundary, not a way of imposing an additional constraint on the solution. An example would be the switch from a central to a one-sided difference for calculating a derivative at a boundary. This requires a portion of code only accessed for a boundary node/cell, but this specialized portion of code is not, in the mathematical sense, enforcing a boundary condition. On the other hand, "physical boundary conditions" are boundary conditions in the mathematical sense. As such, they can be expressed as Dirichlet, Neumann, or Robin boundary conditions (or their higher-order cousins). However, such an expression may not always be in the most convenient variable definition (*e.g.*, characteristic variables). The terms "physical boundary condition" and "numerical boundary condition" arise from a software engineering approach to CFD and reflect the need for specialized portions of code for enforcing the boundary conditions and governing equations on domain boundaries. Unfortunately, these terms are misleading from a mathematical perspective. For clarity in the remainder of this paper, "physical boundary conditions" will simply be referred to as "boundary conditions" since the existing mathematical definition of this phrase is sufficient. References to "numerical boundary conditions" will hereafter be avoided except when absolutely necessary for clarity.

In the method of manufactured solutions, source term values on the interior and boundary are derived from the manufactured solution. This source term is produced via operation on the manufactured solution by the governing equation set. The source term then balances the system and makes the manufactured solution satisfy the modified equation set. At the boundary, operating on the manufactured solution with the governing equation set produces a source term which corrects for the local violation of the governing equation set by the manufactured solution (*i.e.*, any "numerical boundary conditions"), but not for any violation of the boundary conditions. If the manufactured solution satisfies the boundary conditions, then the source term generated by operating on the manufactured solution with the governing equations is sufficient to balance the equation. If the manufactured solution does not satisfy the boundary conditions, then

an additional boundary source (or flux) must be added locally to balance the equations at the boundary. The local boundary source can be calculated in the same fashion as the global MMS source if the boundary condition is expressed as an operator. If the boundary condition is expressed as a flux, then a secondary flux can be added which corrects for the boundary condition violation.

In many CFD codes, the interior source term is evaluated for all of the cells, including the boundary cells, but there is no special source term applied only at the boundary. For such codes, the manufactured solution needs to satisfy the boundary conditions. However, it is not necessary that the manufactured solution satisfy the governing equation set at the boundary (*i.e.*, any "numerical boundary conditions"). This is because any violation of the governing equation set by the manufactured solution will be corrected by the source term at the boundary, just as violations are corrected by the source term in the interior. The order of accuracy observed during testing with the manufactured solution will then be the minimum of the following:

1. on the interior, the order at which the solution to the discrete equation set converges to the solution to the continuum equation set,

2. on the boundary, the order at which the solution to the discrete equation set converges to the solution to the continuum equation set, and

3. the order at which the discrete enforcement of the boundary conditions (*i.e.*, what the code actually does) converges to the exact enforcement of the boundary conditions (*i.e.*, what the code is intended to approximate).

Note that item 2 is tested regardless of whether a "numerical boundary condition" is imposed by the code or whether the boundary discretization allows integration of the interior equations all the way to the boundary. The implementation of the code does not need to be considered in the design of the manufactured solution.

Solid Surface Boundary Conditions

For testing a solid surface condition, a surface must first be defined. Let $F = F(x, y, z)$ be a function from $\Re^3$ into $\Re$ such that $F = C$ defines a surface for any constant C. The normal to this surface is defined by

$$\hat{n} = \frac{\nabla F}{\|\nabla F\|}. \tag{18}$$

The manufactured solution can be constructed so that any variable ϕ satisfies either a Dirichlet (*i.e.*, ϕ constrained) or Neumann (*i.e.*, $\frac{\partial \phi}{\partial n}$ constrained) boundary condition along the surface. So that Eq. (18) will represent the outward facing normal, the domain of interest is taken to be on the side $F - C < 0$.

Conditions on Velocity

The two solid surface conditions discussed in this work are the slip condition ($\vec{v} \cdot \hat{n} = 0$) and the no-slip condition ($\vec{v} = 0$). Realistically, all solid surfaces for Euler flows are slip surfaces and all solid surfaces for continuum Navier-Stokes flows are no-slip surfaces, but practically speaking, it is frequently permissible and convenient to couple the slip condition with the Navier-Stokes equations. This is sometimes done for a surface which is far away from the region of interest (such as walls in a wind tunnel). It is also done when a boundary is defined along a stream surface rather than a solid surface. Because of this, the terms slip and no-slip are used in this paper to refer to the conditions imposed upon velocity only. Whether other boundary conditions need to be imposed depends upon the governing equation set. Thus, for an Euler flow, the only possible solid surface condition is the slip; whereas for a single species Navier-Stokes flow, the surface boundary condition is the velocity condition plus any remaining boundary conditions corresponding to incoming characteristic information.

The 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,

$$\vec{\mathcal{F}} \cdot \hat{n} = \begin{bmatrix} 0 \\ p\hat{n} \\ 0 \end{bmatrix}, \tag{19}$$

where $\vec{\mathcal{F}}$ is the flux hypervector and $\hat{n}$ is the unit normal vector of the surface. For an Euler flow, this condition is the actual boundary condition. 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) 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 condition on the pressure (or any variable other than v_n) because any discrepancies will be corrected by the source term.

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

$$\nabla F \cdot \vec{v} = F_x u + F_y v + F_z w = 0. \tag{20}$$

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:

$$w = -\frac{F_x u + F_y v}{F_z}. \tag{21}$$

This imposes an additional constraint on the function F: $F_z \neq 0$; $\forall (x, y, z) \in \Re^3$. This can be easily enforced

by defining $F(x,y,z) = \widetilde{F}(x,y) - z$. Thus, the surface $F = C_{\mathrm{s}}$ can be expressed as $z = \widetilde{F}(x,y) - C_{\mathrm{s}}$. With this surface definition, u and v can be left in the general form of Eq. (17) with no additional constraints.

The No-Slip Condition

The no-slip condition is actually slightly simpler, although a slight deviation from the form given in Eq. (17) is necessary. The following equation for u satisfies $u = 0$ on the surface $F = C_{\text{n-s}}$:

$$u = (C_{\text{n-s}} - F) u_0 \left[1 + f(Ax) f(By) f(Cz)\right]. \tag{22}$$

If v and w are of the same form as u, then the manufactured solution satisfies the no-slip condition for velocity along $F = C_{\text{n-s}}$. If, however, w is of the form of Eq. (21), then the manufactured solution will satisfy the no-slip condition on $F = C_{\text{n-s}}$ as well as the slip condition on any $F = C_{\mathrm{s}}$ where $C_{\mathrm{s}} \neq C_{\text{n-s}}$. This is convenient since it allows the same manufactured solution to be used for testing the slip and no-slip conditions.

Additional Boundary Condition for the Navier-Stokes Equation Set

Because of the presence of thermal diffusion, the Navier-Stokes equation set requires the imposition of one additional boundary condition at a no-slip surface. This boundary condition is most often either a Dirichlet boundary condition (*i.e.*, T specified) or a Neumann boundary condition (*i.e.*, $\frac{\partial T}{\partial n}$ specified). The Neumann condition can also be expressed by specifying the heat flux. If Dirichlet and Neumann boundary conditions 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 boundary condition that the manufactured solution already satisfies. It can be expressed as either a Dirichlet or Neumann boundary condition. However, if the code has a special case, such as uniform temperature ($T = T_c$) or adiabatic boundary ($\frac{\partial T}{\partial n} = 0$), then the manufactured solution must satisfy the special case in order to verify its implementation.

Adiabatic Boundary

The temperature at an adiabatic ($\frac{\partial T}{\partial n} = 0$) surface is subject to a Neumann boundary condition. This constraint defines a partial differential equation (PDE) for T:

$$\nabla T \cdot \nabla F = F_x T_x + F_y T_y + F_z T_z = 0. \tag{23}$$

A manufactured solution to T which satisfies this PDE can be used for adiabatic tests. One such solution is given in a later section.

Isothermal Boundary

If an isothermal (*i.e.*, constant temperature) boundary condition is implemented as a strict Dirichlet boundary condition, then verification by the method of manufactured solutions is probably unnecessary. Simple inspection of the boundary values should indicate whether or not the condition is applied correctly. However, in the event that the constant temperature is maintained indirectly through the flux or residual vector formulations, some formal verification is necessary.

The isothermal boundary is formulated along a surface $F = C_{\mathrm{i}}$ 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 $F = C_{\mathrm{i}}$ and that T has sufficient spatial variation away from the surface. This can be done with the following equation:

$$T = T_c + (C_{\mathrm{i}} - F) T_0 \left(1 + f(Ax) f(By) f(Cz)\right). \tag{24}$$

Adiabatic and Isothermal

Because much of the work in verification by the method of manufactured solutions 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 boundary conditions. If the constraints mentioned in the previous sections are combined, the following solution is obtained:

$$T = T_c \left(1 + g(x,y,z) \left[C_{\mathrm{a/i}} - F\right]^n\right), \tag{25}$$

where $n > 1$ and $g(x,y,z) \geq 0$. Note that this is a departure from the general form in Eq. (17). The term $\left[C_{\mathrm{a/i}} - F\right]^n$ ensures that $T = T_c$ and $\frac{\partial T}{\partial n} = 0$ 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 $F(x,y,z) = \frac{1}{2}\cos(2x)\cos(3y) - z$, then g must be a function of z so that the leading error terms for the discretization of $\frac{\partial^2 T}{\partial z^2}$ are nontrivial. If $g(z) = 1 + \frac{1}{4}\sin(z)$, this condition is satisfied. Alternatively in this case, n could simply be increased until T has sufficient variance in z, but that has the effect of uniformly raising T, making it more difficult to locate supersonic regions to test supersonic inflow and outflow conditions. Also, raising n 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 more resolution (*i.e.*, bigger meshes and more expensive computations) to reach the asymptotic regime.

Turbulence Variable(s) for the RANS Equation Set

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

$$\widetilde{\nu} = (C_{\text{n-s}} - F) \widetilde{\nu}_0 \left(1 + f(Ax) f(By) f(Cy)\right). \tag{26}$$

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, so a different form must be used. Something similar to Eq. (24) would be sufficient.

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. Since the manufactured solution needs to match the boundary conditions, the eigenvalues must also conform to the signs dictating which boundary conditions are matched (for an inflow surface, $v_n < 0$, and for an outflow surface, $v_n > 0$). Each of these surfaces may be either subsonic ($M_n < 1$), supersonic ($M_n > 1$), or a mixture of the two. Since the previously defined function F used for defining the slip and no-slip surfaces can not also be used for defining an inflow or outflow surface, a new function must be considered. Let $G = G(x, y, z)$ be a function from $\Re^3$ into $\Re$ such that $G = C$ defines a surface for any constant C. The definition of this function and the value of C are both arbitrary. 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.

Subsonic Outflow Condition

Since the previously derived manufactured solutions for u, v, w, and T leave only one degree of freedom for an Euler solution, an inflow condition (having four or five incoming characteristics, depending on the Mach number) can not be tested. However, an outflow condition (having one or no incoming characteristics, depending on the Mach number) can be tested. The pressure profile can be fixed so that it is constant along a general surface $G(x, y, z) = C_{\mathrm{o}}$. This condition is similar to the isothermal condition at the solid boundary, so an equation for pressure similar to Eq. (24) can be used.

$$p = p_c + (C_{\mathrm{o}} - G)\, p_0 \left(1 + f(Ax)f(By)f(Cz)\right) \quad (27)$$

It is also necessary that $v_n > 0$ and $M_n < 1$. However, these conditions are not very restrictive, so they can be combined with the conditions used to define earlier versions of the velocity profiles. Since the solution is intended to be used for testing all previously mentioned BC's in addition to the outflow condition, w is still defined by Eq. (21). The resulting normal velocity is

$$v_n = \frac{\vec{v} \cdot \nabla G}{\|\nabla G\|} = \frac{\left(G_x - G_z \frac{F_x}{F_z}\right) u + \left(G_y - G_z \frac{F_y}{F_z}\right) v}{\|\nabla G\|}. \quad (28)$$

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. If $\mathcal{A}_1 = G_x - G_z \frac{F_x}{F_z}$, $\mathcal{A}_2 = G_y - G_z \frac{F_y}{F_z}$, $\mathcal{B}_1 = \frac{\mathcal{A}_1}{\sqrt{\mathcal{A}_1^2 + \mathcal{A}_2^2}}$, and $\mathcal{B}_2 = \frac{\mathcal{A}_2}{\sqrt{\mathcal{A}_1^2 + \mathcal{A}_2^2}}$, then u and v can be defined as

$$u = (C_{\text{n-s}} - F)\, \mathcal{B}_1 u_0 \left(1 + f(Ax)f(By)f(Cz)\right) \quad (29)$$

$$v = (C_{\text{n-s}} - F)\, \mathcal{B}_2 v_0 \ \left(1 + f(Ax)f(By)f(Cz)\right). \quad (30)$$

This guarantees that the surface is an outflow surface ($v_n > 0$) if both $(1 + f(Ax)f(By)f(Cz))$ and $(C_{\text{n-s}} - F)$ 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. Assuming the terms $\mathcal{B}_1$ and $\mathcal{B}_2$ have sufficient variance in x, y, and z, u and v can be simplified to

$$u = (C_{\text{n-s}} - F)\, \mathcal{B}_1 u_0 \quad (31)$$

$$v = (C_{\text{n-s}} - F)\, \mathcal{B}_2 v_0. \quad (32)$$

Supersonic Outflow

Provided that the u_0 and v_0 from Eq. (31) and (32) and n and T_0 from Eq. (25) are defined appropriately, there will exist regions where $M_n > 1$ and regions where $M_n < 1$. Thus, different regions of the solution can be used to test implementation of supersonic and subsonic outflow enforcement at the surface $G = C_{\mathrm{o}}$. For the supersonic case, there is no boundary condition enforced, so there is no constraint upon p.

The Computational Domains

Functions F and G are used to define some of the boundaries of the computational domain. A third function, H is also needed to construct the domain, although no boundary conditions are tested on H-constant surfaces. The no-slip surface is the $F = 0$ surface, and the exit surface is the $G = 0$ surface. All other F-constant surfaces are slip surfaces. F, G, and H are defined as follows:

$$F = \frac{1}{2}\cos(2x)\cos(3y) - z \quad (33)$$

$$G = x - \frac{1}{2}\cos(y)\sin(2z) - \frac{\pi}{4} \quad (34)$$

$$H = -y. \quad (35)$$

A simple definition of H is permissible since no boundary conditions will be tested on H-constant surfaces. A six-sided domain can be defined by setting maximum and minimum values for F, G, and H. Figure 1 shows two domains. The smaller domain (shown in green) was used for testing the slip condition along its F_{max} boundary, the supersonic outflow condition along its $G = 0$ boundary, and the Euler equations on its interior. The larger domain (shown in red) will be

used for future testing of the no-slip condition along its $F = 0$ boundary, the subsonic outflow condition along its $G = 0$ boundary, and all available equation sets on its interior. The mesh on the larger (red) domain has points clustered toward the no-slip surface in an effort to allow the tests to be performed on reasonably-sized meshes.

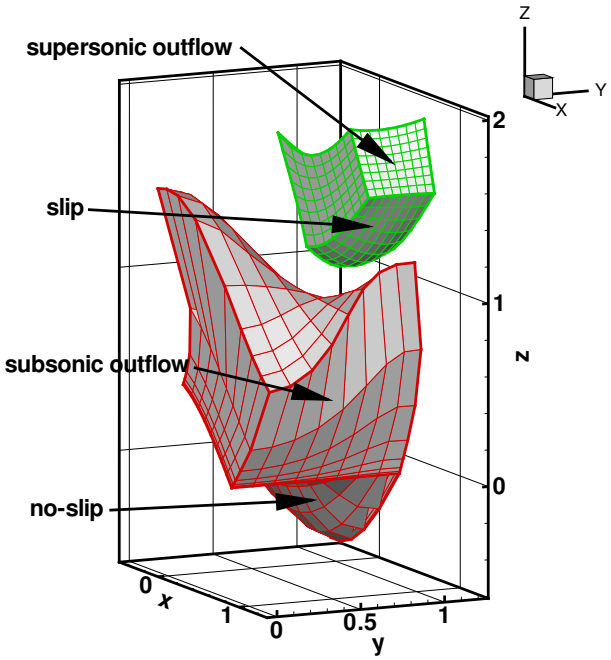

Fig. 1 Computational domains for current and future MMS testing

The Manufactured Solution

Once F, G, and H are defined, the manufactured solution is defined as follows:

$$\mathcal{A}_1 = G_x - G_z \frac{F_x}{F_z} \tag{36}$$

$$\mathcal{A}_2 = G_y - G_z \frac{F_y}{F_z} \tag{37}$$

$$\mathcal{B}_1 = \frac{\mathcal{A}_1}{\sqrt{\mathcal{A}_1^2 + \mathcal{A}_2^2}} \tag{38}$$

$$\mathcal{B}_2 = \frac{\mathcal{A}_2}{\sqrt{\mathcal{A}_1^2 + \mathcal{A}_2^2}} \tag{39}$$

$$u = \frac{3}{4}\left(-F\right)\mathcal{B}_1 a_{\mathrm{sl}} \tag{40}$$

$$v = \frac{3}{4}\left(-F\right)\mathcal{B}_2 a_{\mathrm{sl}} \tag{41}$$

$$w = -\frac{F_x u + F_y v}{F_z} \tag{42}$$

$$T = T_{\mathrm{sl}}\left(1 + \frac{1}{4}\left(1 + \sin(z)\right)\right) F^2 \tag{43}$$

$$p = p_{\mathrm{sl}} - G p_{\mathrm{sl}}\left(1 + \cos(3x)\cos(y) cos(2z)\right) \tag{44}$$

$$\widetilde{\nu} = -F\widetilde{\nu}_\infty\left(1 + \frac{1}{2}\cos(2x)\cos(3y)\cos(4z)\right), \tag{45}$$

where:

$$a_{\mathrm{sl}} = 340\mathrm{m/s} \tag{46}$$

$$p_{\mathrm{sl}} = 101.3\mathrm{kPa} \tag{47}$$

$$T_{\mathrm{sl}} = 288\mathrm{K} \tag{48}$$

$$\widetilde{\nu}_\infty = 1.95\nu_{\mathrm{sl}} \tag{49}$$

$$\nu_{\mathrm{sl}} = 1.464 \times 10^{-5}\mathrm{m}^2/\mathrm{s}. \tag{50}$$

Testing of Interior Equations

Before any boundary condition is tested, it is a good idea to test the interior equation set with which it is coupled using only Dirichlet boundary conditions. (In this case, the term "Dirichlet" is used in its mathematical sense to mean that the value of the dependent variable is specified at the boundary. In Premo, the Dirichlet option sets the dependent variable to a constant at the boundary. The more general case of a time or spatially varying Dirichlet condition is handled via a user-defined boundary condition option.) This allows the interior equation set to be verified so that any subsequent deviation from the observed order of accuracy in the boundary condition tests can be rightfully interpreted as a problem with the boundary condition enforcement, rather than 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 boundary conditions.

Because of the dependence of RANS and laminar Navier-Stokes calculations on inviscid fluxes, the Euler equation set was verified first. Also, the relatively low cost of Euler calculations provided a good starting point. The Euler equation set was tested for both the Green-Gauss and least-squares methods for gradient reconstruction on unstructured meshes. Future work will involve testing of the laminar Navier-Stokes and RANS equation sets.

Figure 2 shows the observed order of accuracy of the code for each mesh level. N is the number of nodes in each of the three mesh dimensions. For mesh refinement via doubling, the observed order of accuracy is given by the following equation:

$$\text{observed order} = \frac{\log\left(\frac{\|\epsilon_{N/2}\|}{\|\epsilon_N\|}\right)}{\log(2)}, \tag{51}$$

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 L_∞, L_2, and L_1 norms are given for the errors in the flow variables ρ and w. The behavior or the ρ error is representative of the error in p, T, u, and v. These values indicate that the second-order convergence is being asymptotically approached for these code options. The L_∞ norm of w, however, lags the others and is not monotone. The

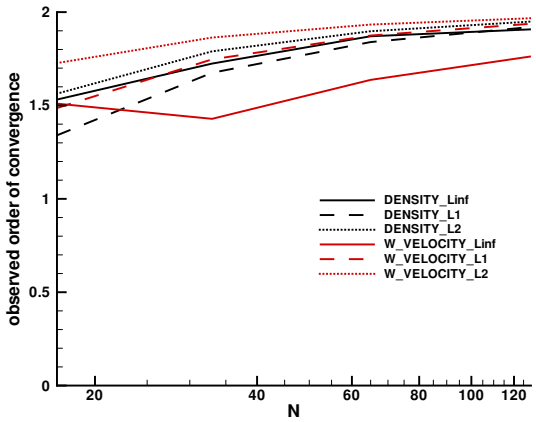

Fig. 2 Convergence order for the Euler equations with Green-Gauss

manufactured solution for w contains higher frequencies because of the number of products of sinusoidal functions dictated by Eq. (21). Higher frequencies require tighter mesh spacing to resolve, so more mesh levels are required to reach the asymptotic regime. Because of this, one more mesh level will be run in the future in order to observe the behavior of this norm. However, as seen in Fig. 3, least-squares gradient reconstruction only demonstrates first-order convergence at best. Roy[7] observed second-order accuracy for the Euler equations using least-squares gradient reconstruction with Premo, but his results were obtained on uniform, Cartesian meshes. The least-squares formulation used for the current tests was unweighted, and the authors believe that the first-order behavior is a result of inaccurate gradient calculation along curved boundaries and in skewed or higher aspect ratio cells. Inaccuracy in unweighted least-squares gradient reconstruction on non-uniform meshes was examined in-depth by Mavriplis.[8] He found that inverse distance weighting produced a much more accurate result. While inverse distance weighted reconstruction is available in Premo, it was not used in this study. Future verification tests with least-squares gradient reconstruction will examine the inverse distance weighted case.

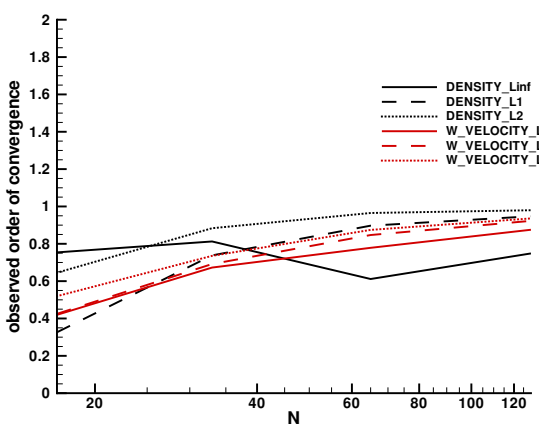

Fig. 3 Convergence order for the Euler equations with least-squares

Testing of Boundary Conditions

Since the Euler equation set implementation was verified, its associated boundary conditions were examined next. Because the Euler equations verified to second-order only with Green-Gauss gradient reconstruction, each of the boundary conditions was tested using Green-Gauss. That way, any first-order error generated by the boundary conditions could be easily detected.

The Slip Condition

Figure 4 shows the observed order of accuracy versus mesh size for the slip condition. Although the L_1 norms appear to be converging second-order accurately, the L_2 norms appear to be converging to less than second-order, and the L_∞ norms do not appear to be converging at all. This seems paradoxical, since the convergence of one norm in the limit of $N \to \infty$ implies that all norms should converge. However, this sample represents only the behavior in the range $9 \leq N \leq 129$. Convergence problems tend to show up on coarser meshes in the L_∞ norm while the the L_2 and especially the L_1 norms tend to mask the lack of convergence until the maximum local error (the only component of L_∞) dominates in the L_2 and L_1 computations. Figure 5 shows the error norms versus mesh size for the slip condition. The L_∞ norms level off, while the L_2 and L_1 norms continue to converge. It can be reasonably expected that further increases in N would soon show the L_2 norm leveling off and eventually also show the L_1 norm leveling off. If so, that would indicate the presence of an unordered component of the error which would need to be removed via a change in boundary discretization or by identifying

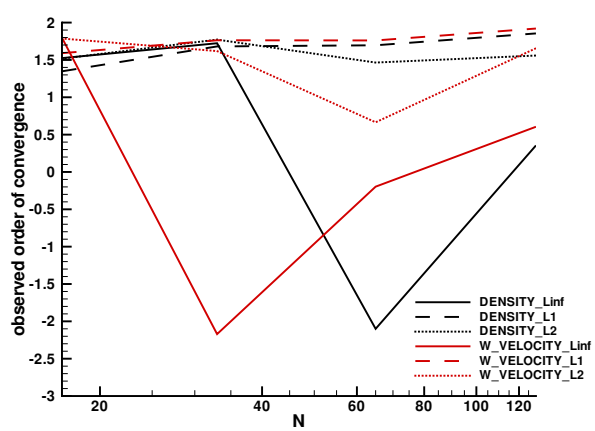

Fig. 4 Convergence order for the slip condition

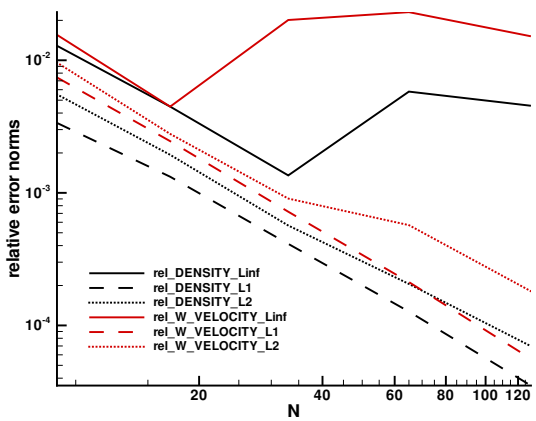

Fig. 5 Relative error norms versus mesh size for the Slip BC

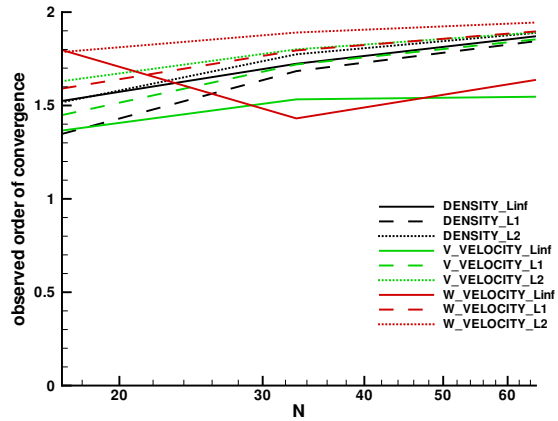

Fig. 7 Convergence order for the slip condition on 2 processors

possible coding mistakes with the current boundary discretization.

One major advantage of verification via manufactured or exact solutions is that the actual error can be examined in the domain. Figure 6 shows the error on the $N = 65$ mesh. The domain is rotated so that the slip surface can be clearly seen. The maximum error in this solution (and also in the $N = 33$ and $N = 129$ solutions) is located where the slip surface and subdomain boundaries intersect (the mesh is divided by a domain decomposition program prior to the start of parallel calculations). The coarser mesh

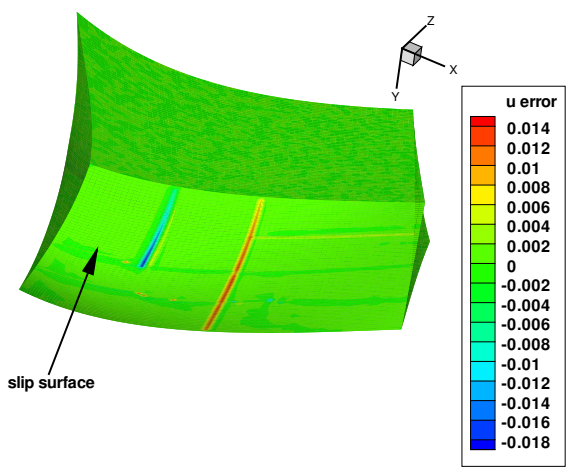

Fig. 6 Error in u on the $N = 65$ mesh

solutions ($N \leq 17$) were produced using only two processors, and the surface dividing the two subdomains did not intersect with the slip surface. However, for the finer meshes ($N \geq 33$), more processors were used, and subdomain boundaries did intersect the slip surface for these cases. This behavior was identified as evidence of a possible problem with the slip condition enforcment at subdomain boundaries.

To determine if the slip condition converges second-order accurately when no subdomain boundaries intersect with the slip surface, the $N = 33$ and $N = 65$ solutions were recomputed on only two processors (the $N = 129$ mesh was too large). Figure 7 shows the observed convergence order for these cases. Most of the norms (including the L_∞ norm of the w error) behave similarly to the Euler, Green-Gauss case with Dirichlet boundary conditions. The exception is the L_∞ norm of the v error, which is shown in this figure because it, like the L_∞ norm of the w error for all of the cases, behaves differently than the norms of the other variables' error. It is not (yet) asymptotically approaching 2 in this range of mesh sizes. More refined meshes ($N = 129$ and $N = 257$) are needed to confirm the convergence order, but it can be concluded that the problem observed in Fig. 4 is not present for this case.

The Supersonic Outflow Condition

Since no information propagates into the domain through a supersonic outflow boundary (all the eigenvalues are positive), the solution at this boundary is actually constrained only by the interior equations (no boundary conditions exist, in the mathematical sense). Figure 8 shows the observed order of accuracy for the Euler equations coupled with the supersonic outflow condition. Correct handling of a fully supersonic outflow condition requires that the only constraints upon the solution come from the governing equations, and this appears to be demonstrated since second-order accuracy is suggested. Finer meshes are needed for confirmation, especially for the error in w.

Conclusions

A manufactured solution was derived which is capable of testing the Euler, laminar Navier-Stokes, and RANS equations on non-uniform meshes along with the slip and no-slip (both adiabatic and isothermal) conditions and outflow (both subsonic and supersonic) boundary. Tests have been carried out for the Euler

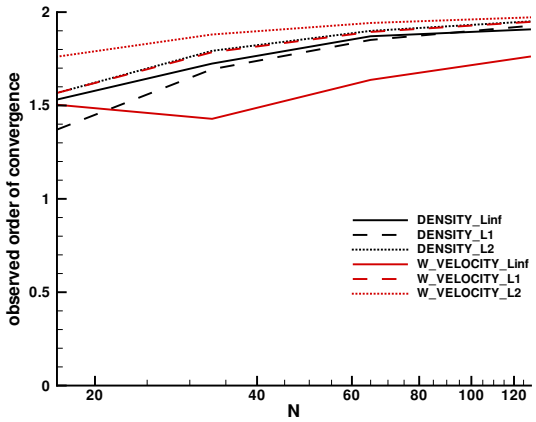

Fig. 8 Convergence order for the supersonic outflow condition

equations using both Green-Gauss and least-squares gradient reconstruction, the slip condition, and the supersonic outflow boundary. The code verified to the expected order of accuracy (second-order) for the Euler equations using Green-Gauss and the supersonic outflow condition. Least-squares gradient reconstruction showed first-order behavior, but the formulation used was unweighted. The authors believe that the first-order behavior is a natural consequence of the formulation, rather than an indication of a bug. Premo contains an inverse-distance weighted formulation, so this formulation will be examined in future tests. The insight gained from the analysis of the least-squares gradient reconstruction underscores the value of performing verification on non-uniform meshes. In previous work[7] on uniform, Cartesian meshes, second-order convergence for the Euler equations was observed with both Green-Gauss and least-squares gradient reconstruction. Tests of the slip condition identified a problem with the interaction between the slip condition enforcement and the parallel decomposition and communication. Second-order accuracy is suggested for the slip condition when no subdomain boundaries intersect the slip surface, but finer meshes are needed to confirm this.

This work demonstrated the value of using the method of manufactured solutions for verifying the boundary conditions of CFD codes. The derived manufactured solution is applicable not only to Premo, but also to general CFD codes. Through the implementation of this manufactured solution in Premo, a bug was found in the parallel enforcement of the slip condition which would have otherwise been difficult to detect. Also, the possible sensitivity to weighting in least-squares gradient reconstruction was identified. If a sequence of solutions using the inverse distance weighting shows second-order convergence, then the default weighting within Premo will be changed from unweighted to inverse distance weighting. (Both options currently exist in the code; the only issue is which should be the default.)

Acknowledgments

The authors would like to thank Alfred Lorber and Thomas Smith for their help in implementing the manufactured solutions in Premo. They, along with coauthor Curtis Ober, are Premo developers.

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*, Elsevier Scientific, K. J. Bathe, ed., June 2001.

[3]Spalart P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," *La Recherche Aerospatiale*, No. 1, 1994, pp. 5–21.

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

[5]Strauss, W. A., *Partial Differential Equations: an Introduction*, John Wiley and Sons, Inc., New York, 1992, pp. 82–86.

[6]Hirsch, C., *Numerical Computation of Internal and External Flows Volume 2: Computational Methods for Inviscid and Viscous Flows*, John Wiley and Sons, Chichester, 1990, p. 173.

[7]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.

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