32nd AIAA Fluid Dynamics Conference and Exhibit
24-26 June 2002, St. Louis, Missouri

AIAA 2002-3

AIAA Paper 2002-3110

Verification of a Compressible CFD Code using the Method of Manufactured Solutions

Christopher J. Roy,[†] Thomas M. Smith,[‡] and Curtis C. Ober[§]

Sandia National Laboratories*

P. O. Box 5800

Albuquerque, NM 87185

Abstract

The computational fluid dynamics code Premo is currently being developed to solve compressible flow problems with a finite-volume approach using unstructured meshes. This code employs both the Green-Gauss and the Least Squares gradient approximations in order to achieve a spatial accuracy that is formally second order for both convection and diffusion. In this paper, the Method of Manufactured Solutions is employed to generate exact solutions to the governing equations along with additional source terms. The exact solutions are then used to accurately evaluate the discretization error in the numerical solutions. The Manufactured Solutions approach is applied to the 2D inviscid Euler equations (both subsonic and supersonic), the 3D Euler equations (supersonic), and the 2D Navier-Stokes equations (both subsonic and supersonic). Through global discretization error analyses, the spatial order of accuracy is observed to be second order for all but one of the ten cases examined. With regards to the other nine cases, a high degree of confidence is achieved that the Premo code is free from coding mistakes in the spatial discretization for uniform meshes.

† Senior Member of Technical Staff, MS 0825, E-mail: cjroy@sandia.gov, Senior Member AIAA

‡ Senior Member of Technical Staff, MS 0316, E-mail: tmsmith@sandia.gov, Member AIAA

§ Principal Member of Technical Staff, MS 0316, E-mail: ccober@sandia.gov, Member AIAA

* Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Nomenclature

e	energy, m^2/s^2
f	source term
f_s	sine or cosine function
h	measure of grid spacing
k	thermal conductivity, $W/(m \cdot K)$
L	domain length, m
p	spatial order of accuracy, or pressure, N/m^2
Pr	Prandtl number ($Pr = 1.0$)
q	heat flux vector, $N/(m \cdot s)$
R	specific gas constant ($R = 287.\ Nm/(kg \cdot K)$)
r	grid refinement factor
$\vec{r}$	position vector, m
t	time, s
T	temperature, K
u, v, w	Cartesian velocity components, m/s
x, y, z	Cartesian spatial coordinates, m
γ	ratio of specific heats ($\gamma = 1.4$)
μ	absolute viscosity, $N \cdot s/m^2$
ρ	density, kg/m^3
τ	shear stress tensor, N/m^2
ϕ	general solution variable

Subscripts

exact	exact value
k	mesh level (k = 1, 2, 3, etc., fine to coarse)
m	mass equation
n	flowfield node index
x, y, z	x-, y-, or z-momentum equation
e	energy equation

Introduction

Modeling and Simulation (M&S) has enormous potential to impact the design, analysis, and optimization of engineering systems. Here M&S is viewed as the numerical solution to any set of partial differential equations which govern continuum mechanics or energy transport (e.g., structural dynamics, heat conduction, electrostatics, fluid dynamics, etc.). In order for M&S to fully achieve its potential, the engineering community must gain increased confidence that it can provide accurate predictions. Although the specific examples presented herein are for Computational Fluid Dynamics (CFD), the general concepts apply to any M&S code.

The sources of error in M&S can be categorized into two distinct areas,[1,2] physical modeling errors (validation-related) and mathematical errors (verification-related). The physical modeling errors arise due to shortcomings in the chosen model or when a model is applied outside of its intended range. An example of the latter is a turbulence model which provides surface heating results for an attached boundary layer flow within 10% accuracy. When this model is applied outside the range where it was calibrated, say for shock-induced separation, then the model may only give 50% accuracy on heating rates.

Mathematical, or verification-related errors, can arise from a number of sources including insufficient mesh resolution, improper time step, incomplete iterative convergence, round-off, and coding mistakes (i.e., coding bugs). In addition, the presence of coding errors, or mistakes, is an often overlooked source of error in M&S. Code developers often rely on expert judgement to determine when a code is producing the correct results. As M&S codes become more complex with numerous modeling options and multiphysics coupling, the reliance on expert judgement can be problematic. This paper will discuss a rigorous method for finding and eliminating coding mistakes known as the Method of Manufactured Solutions.[1,3,4]

Verification is defined as insuring that a model implementation matches the developer's conception.[2] Verification can be broken down into two distinct categories: code verification and solution verification. Code verification is a process performed to provide a high degree of certainty that a code is free from coding mistakes (i.e., coding bugs); however, a formal proof that a piece of software is "bug-free" is probably not forthcoming. If done rigorously, the code verification process needs to be done only once for each independent portion of the code.

There are four different ways to verify a code: the Method of Exact Solutions, the Method of Manufactured Solutions, comparison to benchmark numerical solutions, and code-to-code comparisons. The latter two are really confidence-building exercises, and should not take the place of rigorous code verification. In the Method of Exact Solutions, numerical solutions are compared to exact solutions, often with simplifications to the equations and/or the boundary conditions. In the Method of Manufactured Solutions, an analytical solution is chosen *a priori* and the governing equations are modified by the addition of analytical source terms.

Solution verification, on the other hand, could involve grid and time step refinement studies,[5,6] assessment of iterative convergence error,[7] errors due to singularities, errors due to discontinuities,[6] and errors due to far-field boundary location. Solution verification is a process that should be performed each and every time a predictive simulation is conducted.

The M&S code to be verified in the current work is the CFD code Premo. This code is being developed as part of the Department of Energy's Accelerated Strategic Computing Initiative (ASCI) to meet the needs of the Stockpile Stewardship Program. The Premo code is one of a number of mechanics and energy transport codes that serves as a module to the SIERRA multi-mechanics framework.[8] The SIERRA framework provides services for I/O, domain decomposition, massively parallel processing, mesh adaptivity, load balancing, code coupling, and a host of linear and nonlinear solvers. In the current paper, a wide range of options available in the Premo code are verified.

Numerical Formulation

Euler Equations

The 3D Euler equations in conservation form are

$$\frac{\partial(\rho)}{\partial t}+\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}+\frac{\partial(\rho w)}{\partial z}=f_m \tag{1}$$

$$\begin{aligned}
\frac{\partial(\rho u)}{\partial t}+\frac{\partial(\rho u^2+p)}{\partial x}+\frac{\partial(\rho uv)}{\partial y}+\frac{\partial(\rho uw)}{\partial z}&=f_x\\
\frac{\partial(\rho v)}{\partial t}+\frac{\partial(\rho vu)}{\partial x}+\frac{\partial(\rho v^2+p)}{\partial y}+\frac{\partial(\rho vw)}{\partial z}&=f_y\\
\frac{\partial(\rho w)}{\partial t}+\frac{\partial(\rho wu)}{\partial x}+\frac{\partial(\rho wv)}{\partial y}+\frac{\partial(\rho w^2+p)}{\partial z}&=f_z
\end{aligned} \tag{2}$$

$$\begin{aligned}
&\frac{\partial(\rho e_t)}{\partial t}+\frac{\partial(\rho ue_t+pu)}{\partial x}+\frac{\partial(\rho ve_t+pv)}{\partial y}\\
&\qquad+\frac{\partial(\rho we_t+pw)}{\partial z}=f_e
\end{aligned} \tag{3}$$

where the first term is the unsteady term, and the next three terms are the nonlinear convection terms in the *x*, *y*, and *z*, directions, respectively. For a calorically perfect gas, the Euler equations are closed with two auxiliary relations for energy

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

$$e_t = e + \frac{u^2 + v^2 + w^2}{2} \tag{5}$$

and with the ideal gas equation of state

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

For the solutions presented herein, the ratio of specific heats is $\gamma = 1.4$ and the specific gas constant is $R = 287.0$ *Nm/(kg·K)*.

Navier-Stokes Equations

For viscous flows, the mass conservation equation given in Eq. (1) is still valid; however, the inviscid momentum equations are replaced by the Navier-Stokes equations, which may be written in 2D form as

$$\begin{aligned} \frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p - \tau_{xx})}{\partial x} \\ + \frac{\partial(\rho uv - \tau_{xy})}{\partial y} &= f_x \\ \frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho vu - \tau_{xy})}{\partial x} \\ + \frac{\partial(\rho v^2 + p - \tau_{yy})}{\partial y} &= f_y \end{aligned} \tag{7}$$

Including the viscous effects, the 2D energy conservation equation is now:

$$\begin{aligned} \frac{\partial(\rho e_t)}{\partial t} + \frac{\partial(\rho u e_t + pu - u\tau_{xx} - v\tau_{xy} + q_x)}{\partial x} \\ + \frac{\partial(\rho v e_t + pv - u\tau_{xy} - v\tau_{yy} + q_y)}{\partial y} &= f_e \end{aligned} \tag{8}$$

For the 2D Navier-Stokes equations, the shear stress tensor is

$$\begin{aligned} \tau_{xx} &= \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z}\right) \\ \tau_{yy} &= \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z}\right) \\ \tau_{xy} &= \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \end{aligned} \tag{9}$$

and the heat flux vector is given by:

$$\begin{aligned} q_x &= -k\frac{\partial T}{\partial x} \\ q_y &= -k\frac{\partial T}{\partial y} \end{aligned} \tag{10}$$

For the Navier-Stokes simulations presented herein, the absolute viscosity μ is chosen to be a constant, and the thermal conductivity k is determined from the viscosity by choosing the Prandtl number (here $Pr = 1$):

$$k = \frac{\gamma R}{\gamma - 1}\frac{\mu}{Pr} \tag{11}$$

Discretization

The spatial discretization employed in the Premo code is a node-centered finite-volume formulation. This discretization is implemented on unstructured meshes using an edge-based scheme which allows arbitrary element topologies, where an element is determined by connecting nearest-neighbor nodes. The surfaces of the control volume are found by connecting nodal edge midpoints and element centroids. The convective fluxes are evaluated with Roe's approximate Riemann solver.[10] Second-order spatial accuracy is achieved via MUSCL extrapolation[11] for the primitive variables to the control volume surface. This extrapolation takes the form

$$\begin{aligned} \phi^- &= \phi_L + \frac{1}{2}[\nabla\phi_L] \bullet \Delta\vec{r} \\ \phi^+ &= \phi_R - \frac{1}{2}[\nabla\phi_R] \bullet \Delta\vec{r} \end{aligned} \tag{12}$$

where L and R refer to the nodes to the left and right of the control surface, the - and + refer to the left and right Riemann states, respectively, and $\Delta\vec{r}$ is distance vector

$$\Delta\vec{r} = (x_R - x_L)\,\hat{i} + (y_R - y_L)\,\hat{j} + (z_R - z_L)\,\hat{k}.$$

For the first-order scheme, the gradient term shown in brackets in Eq. (12) is omitted. For the second-order scheme, this gradient is evaluated by either the Green-Gauss (G-G) or Least Squares (LSQ) gradient operator.[12] The gradient is also used in the evaluation of the viscous fluxes at the control-volume surface, resulting in a second-order discretization for the viscous terms. On uniform meshes, this scheme results in an upwind-biased Fromm's scheme for convection and central difference for diffusion. For the steady-state simulations discussed in this paper, the governing equations are integrated in time using a low-storage, four-stage Runge-

Kutta method.[13] See Ref. 9 for more details on the temporal and spatial discretization of the Premo code.

Boundary Conditions

The Euler simulations employ different boundary conditions depending on whether the flow is supersonic or subsonic. For subsonic flow, exact Dirichlet values for all primitive variables are set at both inflow and outflow boundaries based on the specified Manufactured Solution. For the node-centered finite-volume discretization, the Dirichlet boundary condition is enforced by simply setting the nodal boundary value. For supersonic flow cases, all variables are again set with exact Dirichlet values at the inflow boundaries, while for the outflow boundary nodes the weak form of the boundary condition is used where the nodal values are updated from the flux balance consistent with the Roe discretization. These nodal values are used for the entire control-volume surface along the outflow boundary.

For the Navier-Stokes simulations, the viscosity is set to a large value in order to ensure that the diffusive terms are on the same order as the convective terms. By setting the viscosity to a large value, the extrapolation boundary condition for supersonic outflow, which is based on 1D inviscid theory, is no longer valid. Therefore, for the Navier-Stokes simulations, the exact Dirichlet values for all primitive variables are specified on both inflow and outflow boundaries.

For the 2D cases presented, the discretization employs three nodes in the z-direction. A slip-wall boundary condition is used on the z_{min} and z_{max} faces, wherein the face-normal velocity is set to zero at the boundary node, and the nodal pressure at the boundary is found from the flux balance (i.e., the weak form). While this boundary condition for pressure is formally first order, it becomes second order for cases where the interior scheme gives the normal gradient ($\partial p/\partial n$) to be zero at the slip wall (with n normal to the boundary). For the current computations, this boundary condition is simply an artifice to compute 2D flows with a 3D code, and no gradients are formed in the z-direction.

Code Verification

Code verification is a way to build confidence that there are no coding mistakes or bugs in a simulation code. While no rigorous proof of code verification currently exists (and none may be forthcoming), the procedures discussed in this section can provide a high degree of confidence that a code is mistake-free. There is an ongoing debate as to whether code verification is something that must be done only once for a given set of code options,[1] or whether code verification is a process by which code verification evidence is gathered to provide increased confidence that the code is bug free. The authors ascribe to the view that, if done properly, code verification need be performed only once for a given set of code options.

Method of Exact Solutions

In the Method of Exact Solutions, specialized cases are identified where exact solutions exist to a given set of governing equations. For the Euler and Navier-Stokes sets of equations, there are only a limited number of exact solutions. Furthermore, these exact solutions may not exercise all of the terms in the governing equations. For example, in the flow between two infinite parallel plates, one moving relative to the other (Couette flow), the velocity profile is linear, hence the diffusion term, a second derivative of velocity, is identically zero and therefore would not be exercised.

Method of Manufactured Solutions

A more general approach to code verification is the Method of Manufactured Solutions.[1,3,4] In this case, the mathematical form of the solution is chosen *a priori*. The differential operator for the governing equations is applied to this chosen analytical solution to generate analytical source terms. These source terms are implemented within the code, and the modified governing equations (including the source terms) are then discretized and solved numerically and compared to the exact solution.

There are a variety of acceptance criteria for code verification.[3] In order of increasing rigor these criteria are:

- expert judgement
- error quantification
- consistency
- order of accuracy

The order of accuracy test is the most rigorous test and is therefore the recommended acceptance criteria. This test involves evaluating the numerical solution on a series of grids, and with various time steps for unsteady problems. The spatial and temporal discretization error are monitored to determine if the observed order of accuracy matches the formal order of accuracy (which can be determined by a truncation-error analysis of the discretized equations). The error will generally decrease as $1/r^p$, where r is the grid refinement factor (e.g., $r = \Delta x_{coarse}/\Delta x_{fine}$) and p is the order of accuracy. For example, if the numerical scheme is second order ($p = 2$) and the grid is doubled ($r = 2$), then the discretization error should decrease by a factor of four as the mesh is refined.

Although the form of the Manufactured Solution is somewhat arbitrary, it should be chosen to be smooth, infinitely differentiable, and realizable (i.e., solutions should be avoided which have negative densities, pressures, temperatures, etc.). Solutions should also be chosen that are sufficiently general so as to exercise all terms in the governing equations. Adherence to these guidelines will help ensure that the formal order of accuracy is attainable on reasonably coarse meshes. Symbolic manipulation tools such as Mathematica™ or Maple™ can be used to apply the differential operator to the solution and generate source terms. These tools can then be used to generate the FORTRAN or C coding for the source term automatically. See Refs. 1, 3, and 4 for more information on the Method of Manufactured Solutions.

While the Method of Manufactured Solutions can be used to rigorously test the spatial and temporal discretization of a code, as well as the associated boundary conditions, this method will not test the efficiency or robustness of a given numerical solution method. In addition, the method cannot be used to test the stability of a given algorithm. Furthermore, options not exercised by the given Manufactured Solution are not verified.

The six steps required for implementing the Method of Manufactured solutions are:

Step 1) Choose the form of the governing equations
Step 2) Choose the form of the Manufactured Solution
Step 3) Apply the governing equations to the Manufactured Solution to generate analytical source terms
Step 4) Discretize the equations and solve on multiple mesh levels using analytical boundary conditions and source terms from the Manufactured Solution
Step 5) Evaluate the global discretization error in the numerical solutions
Step 6) Determine whether or not the observed order of accuracy matches the formal order of accuracy

If the comparison in Step 6 is favorable, then the coding options exercised are verified. If the comparison is unfavorable, then one generally examines the local discretization error, uses this information to debug the code, then returns to Step 4. The Manufactured Solution itself can be used to help debug the code by selectively turning off certain solution terms (e.g., viscous terms, a given spatial variation, etc.).

Order of Accuracy

The most rigorous test for code verification is the order of accuracy test. This test assesses whether or not the numerical method reproduce the formal order of accuracy in space and/or time. As previously discussed, the discretization error should drop as $1/r^p$, where in the current case the grid refinement factor is $r = 2$ and the nominal order of accuracy is $p = 2$; thus the error should drop by a factor of four on each successively refined mesh level. In order to examine the global discretization error behavior, both the L_2 and the L_∞ norms of the error are used

$$L_2\ \text{norm}_k = \left(\frac{\sum_{n=1}^{N} \left|\phi_{k,n} - \phi_{exact,n}\right|^2}{N} \right)^{1/2} \tag{13}$$

$$L_\infty\ \text{norm}_k = \max\left|\phi_{k,n} - \phi_{exact,n}\right|$$

where k refers to the discrete mesh level and n varies over all mesh nodes N in space (including both interior and boundary nodes) with the exception of the Dirichlet boundary nodes (for which the discretization error is identically zero). Since the L_∞ norm represents the maximum discretization error over the entire domain, obtaining the formal order of accuracy in this norm is generally more difficult than the other error norms. When these global error norms do not reproduce the formal order of accuracy, it is often helpful to examine the behavior of the local discretization error.

Results: Euler Equations

This section reports 2D and 3D Manufactured Solution results for the inviscid Euler equations. For the 2D simulations, two separate cases are examined: one where the flow is supersonic over the entire domain, and one where the flow is subsonic over the entire domain. These two cases are necessary to test the Roe[10] upwind spatial discretization, which uses different reconstruction methods depending on whether the flow is subsonic or supersonic. For the supersonic cases, the x_{max}, y_{max}, (and z_{max} for 3D) boundaries employed the supersonic outflow boundary condition.

2D Supersonic Case

The Method of Manufactured Solutions is first applied to the Euler equations given by Eqs. (1)-(3) along with the auxiliary relations given in Eqs. (4)-(6), which govern the conservation of mass, momentum, and energy for an inviscid (frictionless) fluid. With the governing equations specified, the next step is to choose the form of solution. The general form of the primitive solution variables is chosen as a function of sines and cosines

$$\phi(x, y, z) = \phi_0 + \phi_x f_s\left(\frac{a_{\phi x}\pi x}{L}\right) + \phi_y f_s\left(\frac{a_{\phi y}\pi y}{L}\right) + \phi_z f_s\left(\frac{a_{\phi z}\pi z}{L}\right) \tag{14}$$

where $\phi = \rho$, u, v, w, or p for density, u-velocity, v-velocity, w-velocity, or pressure, respectively, and the $f_s(\cdot)$ functions in Eq. (14) denote either the sine or cosine function. Note that in this case, ϕ_x, ϕ_y, and ϕ_z are constants (the subscripts do not denote differentiation). See Appendix A for the specific form of these primitive variables for all results presented herein (both Euler and Navier-Stokes). The chosen solutions are thus smoothly varying functions in space, while the temporal accuracy is not addressed in the current work. The constants used in the Manufactured Solutions for the 2D supersonic Euler case are given in Table A.1 of Appendix B. These solutions are presented graphically in Fig. 1. The solutions are smooth to allow the formal order of accuracy to be achieved on relatively coarse meshes.

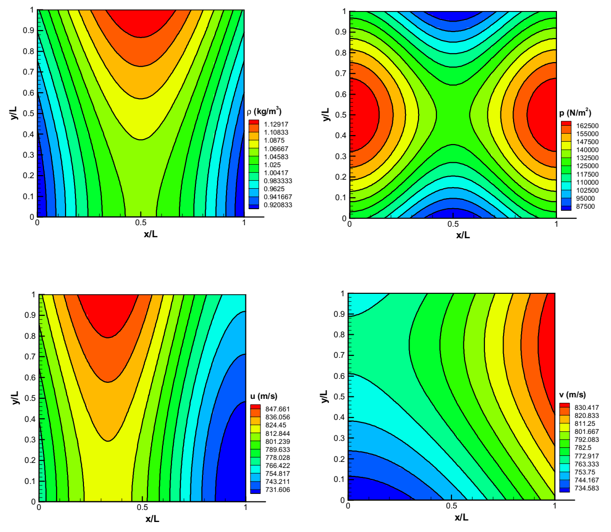

Fig. 1 2D Euler supersonic Manufactured Solution: density (top left), pressure (top right), u-velocity (bottom left), and v-velocity (bottom right).

The governing equations (Eqs. (2)-(6)) were then applied to the chosen solutions using the Mathematica™ symbolic manipulation software to generate FORTRAN code for the resulting source terms. The analytical source term for the 2D Euler mass conservation equation is given in Appendix 3. The source terms for each of the governing equations for this case are shown graphically in Fig. 2.

The governing equations (Eqs. (2)-(6)) were discretized and solved numerically, including the analytical source terms. For a given control volume, the source terms were simply evaluated using the nodal values (i.e.,

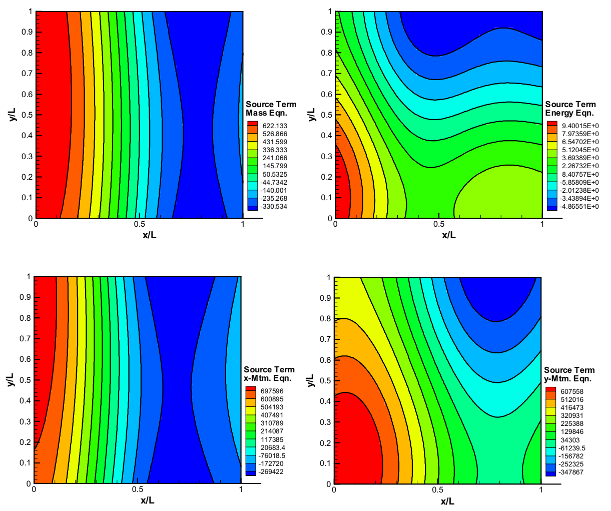

Fig. 2 Generated source terms for the 2D Euler supersonic case: mass (top left), energy (top right), x-momentum (bottom left), y-momentum (bottom right).

the values at the control-volume centroid). The numerical solutions were then compared to the Manufactured Solution to determine the discretization error in the solutions. The Premo code was used to solve the Euler equations using two different discretizations for determining the gradient: the Green-Gauss (G-G) and the Least-Squares (LSQ) approaches. The formal order of accuracy of both of these gradient schemes is second order in space, and the solutions were marched to a steady-state solution using the four-stage Runge-Kutta temporal integration. All solutions presented herein were integrated in time until the steady state residuals were converged to machine zero. Since the Premo code was run in double precision mode, the steady-state residuals were reduced by approximately 14 orders of magnitude from their initial levels. Figure 3 shows the steady-state residual history for one of the grid levels using the G-G gradient. Note that the steady-state residual is defined by plugging the current discrete solution into the discrete form of the steady-state equations (including the source terms).

Since the Manufactured Solution exists for all x, y, (and z for 3D), we are free to choose any sub-domain on which to solve the governing equations. For the 2D (3D) cases presented herein, the numerical solutions are obtained on the domain

$$0 \le x \le 1$$
$$0 \le y \le 1$$
$$(0 \le z \le 1)$$

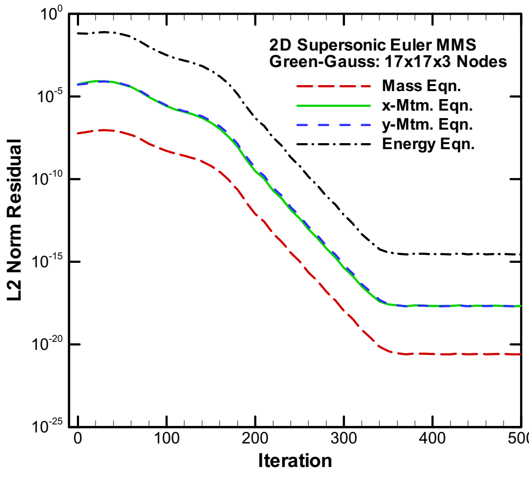

Fig. 3 L_2 norms of the steady-state residuals for the 2D Euler supersonic case with G-G gradient formulation.

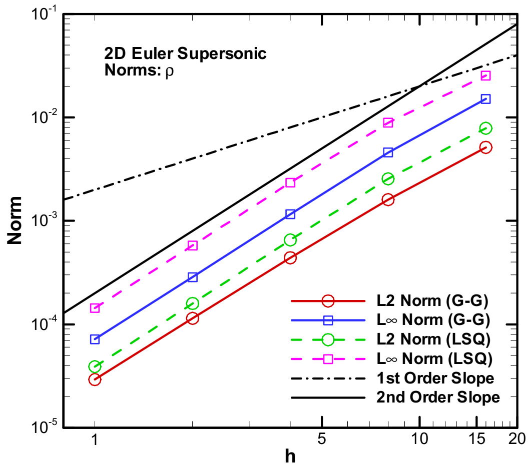

Fig. 4 Behavior of the density error norms as the mesh is refined for the 2D supersonic Euler case.

and the mesh sizes for 2D (3D) are given in Table 1, where

$$h_k = \frac{\Delta x_k}{\Delta x_1} = \frac{\Delta y_k}{\Delta y_1} = \left(\frac{\Delta z_k}{\Delta z_1}\right)$$

Note that since only uniform meshes were examined, the code can not be said to be verified for arbitrary meshes.

Table 1 Meshes for 2D (3D) Manufactured Solutions

Mesh	Mesh Nodes	Grid Spacing, h
Mesh 1	129×129×3	1
Mesh 2	65×65×3(65)	2
Mesh 3	33×33×3(33)	4
Mesh 4	17×17×3(17)	8
Mesh 5	9×9×3(9)	16

The error norms for this case are presented below in Fig. 4 for the conserved quantity density using both the G-G and LSQ gradients. The abscissa shows the measure of the grid spacing h on a log scale, with $h = 1$ being the finest mesh. Also shown on the plot are curves for the first- and second-order slopes for reference. Both the L_2 and L_∞ norms of the error drop by a factor of four with each mesh refinement, thus matching the second-order slope and verifying that the code is producing second-order accurate results.

The order of accuracy can be calculated given two discrete mesh levels (k and $k+1$) and the exact solution by

$$p_k = \ln\left(\frac{L_{k+1}}{L_k}\right) / \ln(r) \tag{15}$$

where "L" refers to one of the global error norms from Eq. (13), $k+1$ refers to the coarser mesh level, and r is the grid refinement factor. Results using Eq. (15) are shown graphically in Fig. 5 which again show that the solutions are second-order accurate as the mesh is refined. Once the order of accuracy is verified (assuming it is greater than zero), it is clear that the code is consistent in the sense that the numerical solution approaches the continuum solution (which was chosen in the beginning) as Δx and Δy approach zero. Although not shown, similar behavior was found for the other conserved variables (ρu, ρv, and ρe_t).

The Manufactured solutions can also be used to compute local discretization error in the numerical solutions. The discretization error in the finest grid solution using the G-G gradient is given in Fig. 6. The largest errors (±0.005%) occur at the bottom left corner between the two inflow planes. Although it is not clear why the error is relatively large error occurs at this corner, the results given in Fig. 5 demonstrate that this is an ordered error (i.e., on that is reduces with the proper order of accuracy as the mesh is refined). Since the flow is everywhere supersonic in both the x- and y-directions for this case, the mean flow direction is oriented at a 45 *deg* angle to the grid. The error at this corner appears to propagate along characteristic Mach lines through the domain.

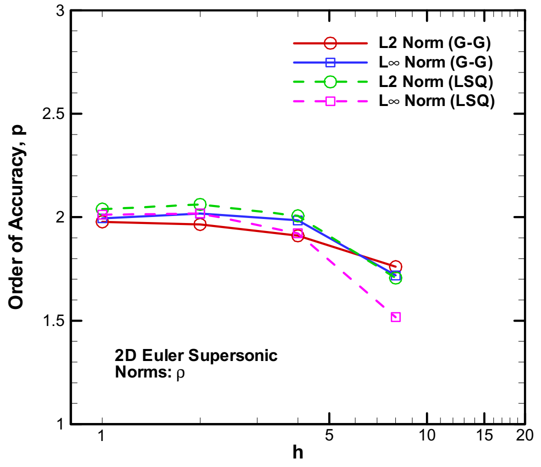

Fig. 5 Order of accuracy of the density error norms as the mesh is refined for the 2D supersonic Euler case.

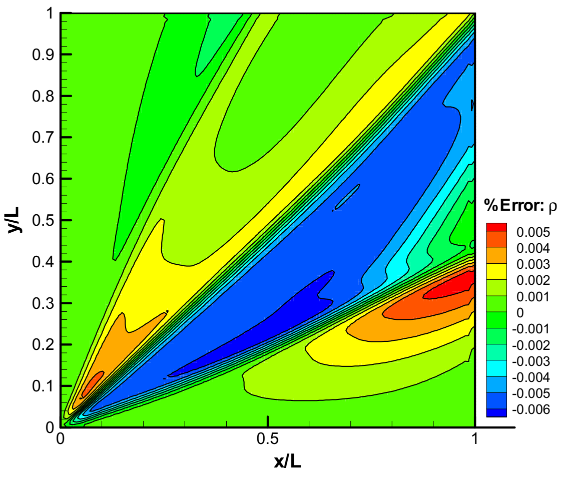

Fig. 6 Numerical error in the fine grid Green-Gauss solution for density for the 2D supersonic Euler case.

2D Subsonic Case

A 2D subsonic Euler Manufactured Solution was generated, and the constants used for this case are presented in Table A.2 of Appendix B. The mean velocities in each of the two coordinate directions were chosen to be approximately one-tenth of the speed of sound. With the exception of the velocity magnitudes, the Manufactured Solution in this case is quite similar to the solution used for the 2D supersonic Euler case.

Figure 7 shows the behavior of the *y*-momentum (ρ*v*) discretization error norms for both the G-G and LSQ gradient methods. Again, the error norms match the second-order slope on nearly all grid levels. The order of accuracy as calculated from Eq. (15) is shown in Fig. 8, confirming that the code is producing second-order results. The conserved quantity momentum is used in this case rather than the velocity, since the governing equations are solved in terms of the conserved variables. The error norms for the other conserved variables gave similar behavior.

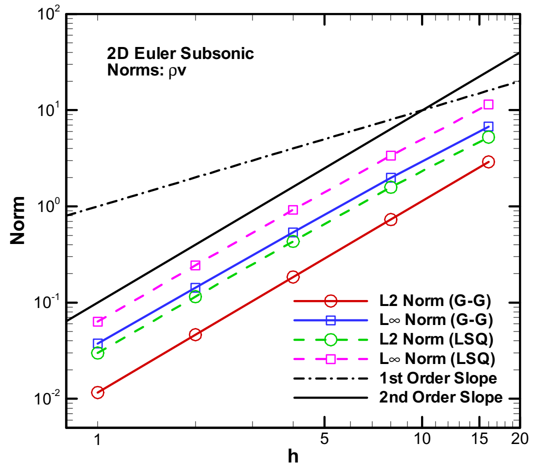

Fig. 7 Behavior of the *y*-momentum (ρ*v*) error norms as the mesh is refined for the 2D subsonic Euler case.

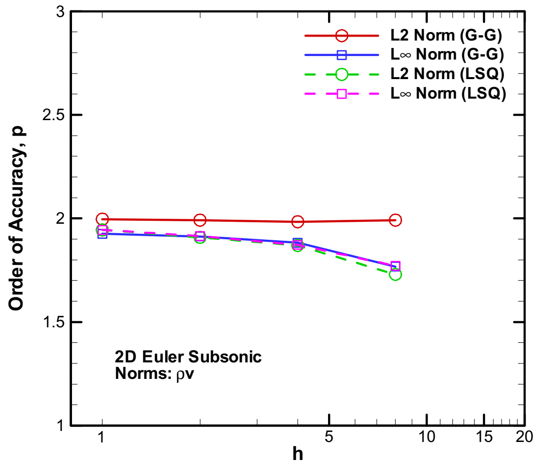

Fig. 8 Order of accuracy of the *y*-momentum (ρ*v*) error norms as the mesh is refined for the 2D subsonic Euler case.

3D Supersonic Case

Due to the limited efficiency of the explicit Runge-Kutta temporal integration for obtaining steady-state solutions, a subsonic Euler solution in 3D was not feasible. However, a supersonic 3D Euler Manufactured Solution was generated using the constants given in Table A.3 of Appendix C. The chosen Manufactured Solution for the w-velocity is shown graphically in Fig. 9, which shows the smooth variation of the w-velocity at the outflow boundaries.

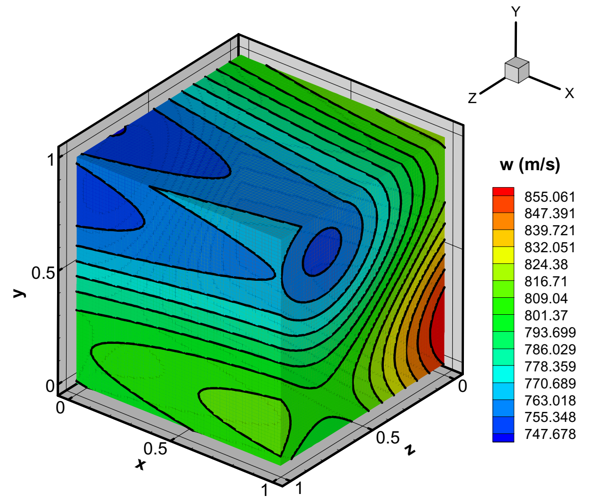

Fig. 9 3D supersonic Euler Manufactured Solution for w-velocity.

The behavior of the global discretization error norms for the w-momentum (ρw) are given in Fig. 10. Due to the larger computational cost involved with the 3D calculations, the finest mesh that could be run was Mesh 2 with 65×65×65 nodes (274,625 nodes total). While there is some first-order behavior evident on the coarser meshes, second-order accuracy is achieved as the mesh is refined. The order of accuracy is shown graphically (using Eq. (15)) in Fig. 11, confirming that the finer grids approach second-order accuracy. Although not shown, the other conserved variables also converged to second order.

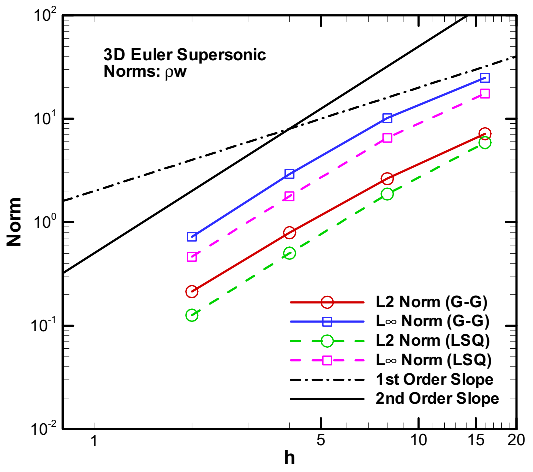

Fig. 10 Behavior of the z-momentum (ρw) error norms as the mesh is refined for the 3D supersonic Euler case.

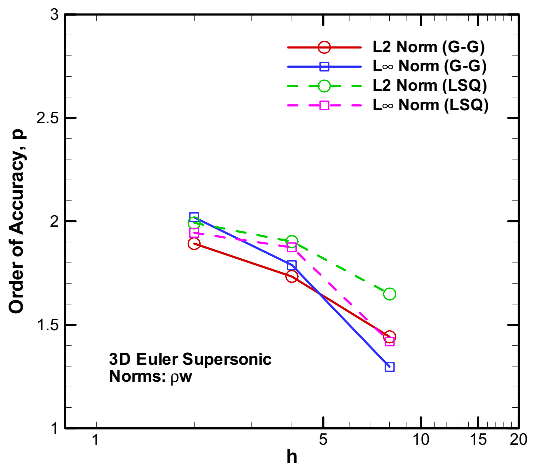

Fig. 11 Order of accuracy of the z-momentum (ρw) error norms as the mesh is refined for the 3D supersonic Euler case.

Results: Navier-Stokes Equations

Results are presented in this section for the Navier-Stokes equations (Eqs. (1),(7), and (8)), along with the auxiliary relationships given in Eqs. (4)-(6) and (9)-(11). Due to the increased computational costs associated with solving viscous, compressible flow with the explicit solver, only 2D results are presented.

2D Supersonic Case

The constants used in the Manufactured Solution for the 2D supersonic Navier-Stokes case are given in Table A.4 of Appendix B. In order to ensure that the viscous terms were of the same order of magnitude as the convective terms, the absolute viscosity was chosen as $\mu = 20\ N{\cdot}s/m^2$. By balancing these two terms, the possibility of a "false positive" on the order of accuracy test is minimized. If instead, only high Reynolds number solutions were examined (where the diffusive terms were much smaller than the convective terms), then extremely fine meshes would be needed in order to detect discretization mistakes in the viscous terms. The ratio of the viscous contribution to the governing equations (terms involving τ or q) to the convective contribution (terms appearing in the Euler equations) was calculated. This ratio is shown graphically in Fig. 12 for the y-momentum equation, and varies from values near 0.1 to large values in regions were the convective terms go to zero.

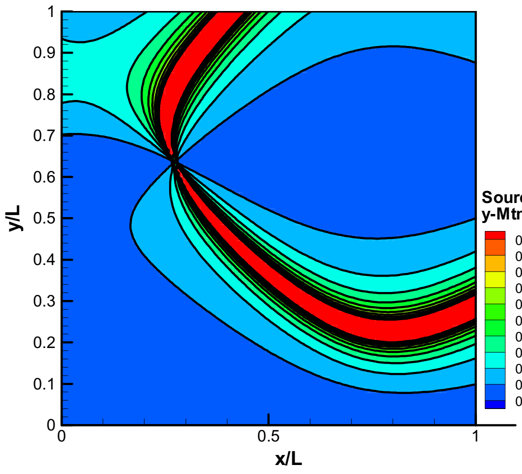

Fig. 12 2D supersonic Navier-Stokes Manufactured Solution: ratio of diffusion to convection terms in the *y*-momentum equation.

The behavior of the global error norms for the *x*-momentum equation are given in Fig. 13. The slopes match the second-order slope in nearly all cases. The order of accuracy from Eq. (15) is shown graphically in Fig. 14. All of the curves approach second order as the mesh is refined. The other conserved variables displayed similar behavior (not shown).

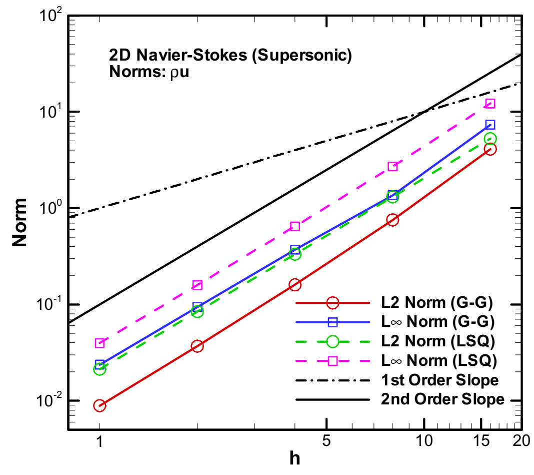

Fig. 13 Behavior of the *x*-momentum (ρ*u*) error norms as the mesh is refined for the 2D supersonic Navier-Stokes case.

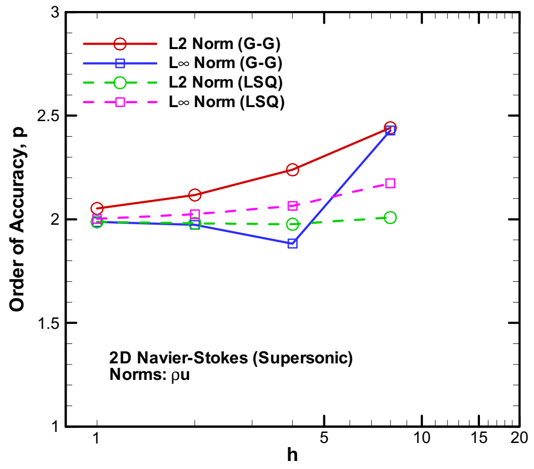

Fig. 14 Order of accuracy of the *x*-momentum (ρ*u*) error norms as the mesh is refined for the 2D supersonic Navier-Stokes case.

2D Subsonic Case

The final Manufactured Solution was generated for the 2D Navier-Stokes equations assuming the flow was subsonic over the entire domain. The constants used in this Manufactured Solution are given in Table A.5 of Appendix B. These solutions are shown graphically in Fig. 15. In order to again ensure that the convective and diffusive terms in the governing equations are fully exercised, the ratio of diffusive to convective terms for the energy equation was computed and is shown in Fig. 16.

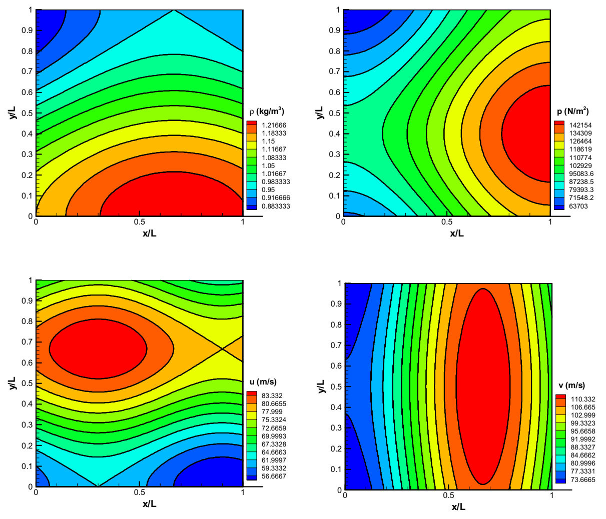

Fig. 15 2D subsonic Navier-Stokes Manufactured Solution: density (top left), pressure (top right), *u*-velocity (bottom left), and *v*-velocity (bottom right).

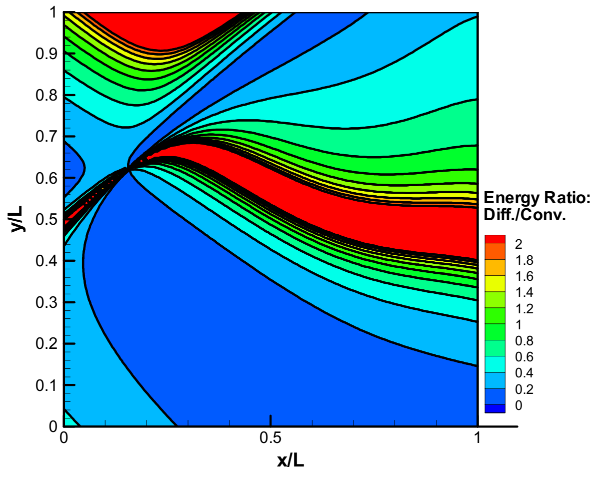

Fig. 16 2D subsonic Navier-Stokes Manufactured Solution: ratio of diffusion to convection terms for the energy equation.

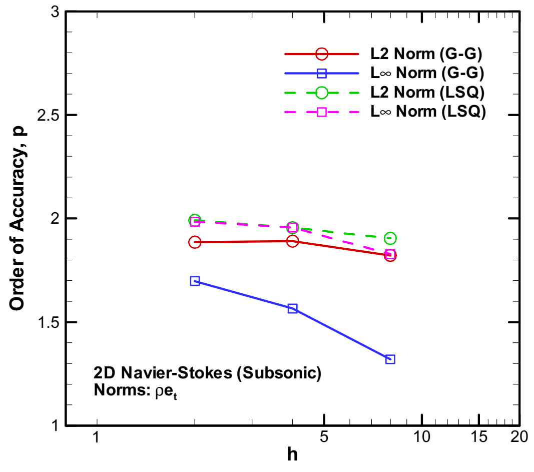

Fig. 18 Order of accuracy of the energy (ρe_t) error norms as the mesh is refined for the 2D subsonic Navier-Stokes case.

Global error norms for the conserved variable energy (ρe_t) are given in Fig. 17. For the Least Squares gradient, these norms match the second-order slope as the mesh is refined; however, the norms for the Green-Gauss gradient do not match the second-order slope, especially on the coarser meshes. These results for spatial convergence are confirmed by examining the order of accuracy, shown in Fig. 18. The results with the Green-Gauss gradient require further investigation. Although not shown, a similar behavior was found for the other conserved variables.

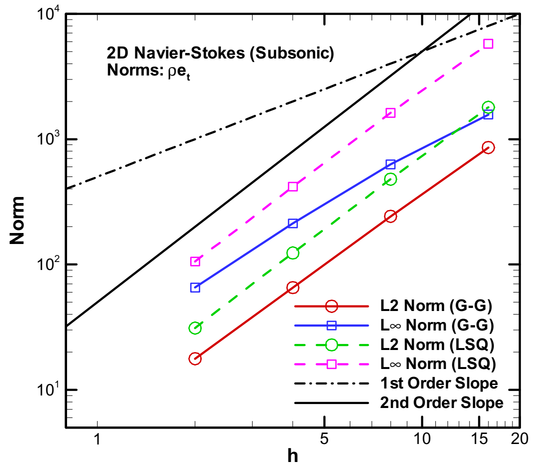

Fig. 17 Behavior of the energy (ρe_t) error norms as the mesh is refined for the 2D subsonic Navier-Stokes case.

The local discretization error in the energy is presented in Fig. 19. The magnitude of this error is largest at the top-left (+0.06%) and bottom-right (-0.02%) corners of the domain. This error near the boundaries is likely caused by the over-specification of the boundary conditions with the exact Dirichlet values. Recall that, in a 1D inviscid sense, a subsonic inflow requires the specification of two properties and the extrapolation of one quantity from within the domain, while a subsonic outflow boundary requires the specification of one quantity and the extrapolation of two quantities from within the domain.

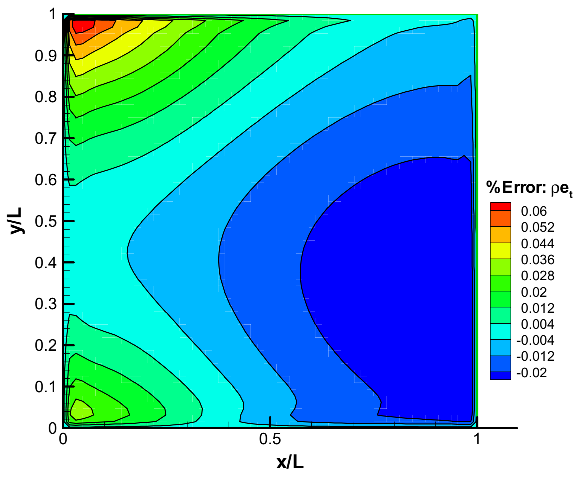

Fig. 19 Numerical error in the fine grid Least Squares solution for energy (ρe_t) for the 2D subsonic Navier-Stokes case.

Conclusions

The Method of Manufactured Solutions has been applied to the new, unstructured compressible fluid dynamics code Premo. A number of cases were examined including both the Euler (2D supersonic, 2D subsonic, and 3D supersonic) and the Navier-Stokes (2D supersonic and 2D subsonic) equations as well as the Green-Gauss and the Least Squares gradient approximations. By solving the Manufactured Solutions on a number of different grid levels, the spatial order of accuracy was ascertained by comparing the numerical solutions to the exact Manufactured Solutions. In nine cases out of the ten examined, the Premo code demonstrated second-order spatial accuracy as the mesh was refined. This By demonstrating that the formal order of accuracy was achieved, the Premo code was verified, thus providing confidence that there are no mistakes in the spatial discretization for uniform meshes. For all subsonic cases, and for the supersonic Navier-Stokes case, the specification of inflow and outflow boundary values for each of the primitive variables with exact Dirichlet values resulted in over-specified boundary conditions; however, the order of accuracy was not adversely affected, except possibly for the subsonic Navier-Stokes case with the Green-Gauss gradient.

The Method of Manufactured Solutions was found to be an invaluable tool for finding coding mistakes. In one case, a logic error allowed the gradient term in the MUSCL extrapolation to be zeroed out when the flux limiter (not used in the current work) was set to zero. This error resulted in first-order behavior of the spatial discretization error which was easily found by the Manufactured Solutions. As another example, an "IF DEF" for a variable was used to turn on an option for a constant angular velocity. This was accidentally set to zero (rather than being left undefined), which triggered the constant angular velocity terms. This coding mistake was found since non-ordered errors were generated at the slip wall boundaries in 2D.

A number of coding options were verified by the Method of Manufactured Solutions. The options verified include: inviscid (Euler) and viscous (Navier-Stokes), the Roe upwind scheme (both subsonic and supersonic), the Green-Gauss and Least Squares gradients, and boundary conditions for Dirichlet values, slip-walls, and supersonic outflow. Options not verified in the current study include: solver efficiency and stability (these are not verifiable with the method), nonuniform meshes, temporal accuracy (the Runge-Kutta scheme is formally second-order accurate, however the chosen solutions were not functions of time), variable transport properties μ and k, and the viscous cross-derivatives (e.g., $\partial/\partial x(\mu \partial u/\partial y) = 0$ for the chosen solutions).

References

1. Roache, P. J., *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, New Mexico, 1998.
2. *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*, AIAA G-077-1998, p. 3.
3. Salari, K., and Knupp, P., "Code Verification by the Method of Manufactured Solutions," SAND 2000-1444, Sandia National Laboratories, Albuquerque, NM, June 2000.
4. Knupp, P., and Salari, K., *Verification of Computer Codes in Computational Science and Engineering*, CRC Press, in press.
5. Roache, P. J., "Perspective: A Method for Uniform Reporting of Grid Refinement Studies," *ASME Journal of Fluids Engineering*, Vol. 116, Sept. 1994, pp. 405-413.
6. Roy, C. J., "Grid Convergence Error Analysis for Mixed-Order Numerical Schemes," AIAA Paper 2001-2606, June 2001.
7. Roy, C. J., and Blottner, F. G., "Assessment of One- and Two-Equation Turbulence Models for Hypersonic Transitional Flows," *Journal of Spacecraft and Rockets*, Vol. 38, No. 5, September-October 2001, pp. 699-710.
8. Edwards, H. C., and Stewart, J. R., "SIERRA: A Software Environment for Developing Complex Multiphysics Applications," Proceedings of the First MIT Conference on Computational Fluid and Solid Mechanics, Elsevier Scientific, K. J. Bathe Ed., June 2001.
9. Smith, T. M., Ober, C. C., and Lorber, A. A., "SIERRA/Premo - A New General Purpose Compressible Flow Simulation Code," AIAA Paper 2002-3292, June 2002.
10. Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," Journal of Computational Physics, Vol. 43, pp. 357-372.
11. van Leer, B., "Towards the Ultimate Conservative Difference Scheme III. Upstream-Centered Finite-Difference Schemes for Ideal Compressible Flow," *Journal of Computational Physics*, Vol. 23, pp. 263-275.
12. Haselbacher, A., and Blazek, J., "Accurate and Efficient Discretization of Navier-Stokes Equations on Mixed Grids," AIAA Journal, Vol. 38, No. 11, 2000, pp. 2094-2102.
13. van der Houwen, P. J., "Explicit Runge-Kutta Formulas with Increased Stability Boundaries," *Numerische Mathematik*, Vol. 20, 1972, pp. 149-164.

Appendix A: Analytical Manufactured Solution

The general form chosen for the Manufactured Solution, for both Euler and Navier-Stokes implementations, is given as follows:

$$\begin{aligned}
\rho(x,y,z) &= \rho_0 + \rho_x \sin\left(\frac{a_{\rho x}\pi x}{L}\right) + \rho_y \cos\left(\frac{a_{\rho y}\pi y}{L}\right) + \rho_z \sin\left(\frac{a_{\rho z}\pi z}{L}\right) \\
u(x,y,z) &= u_0 + u_x \sin\left(\frac{a_{ux}\pi x}{L}\right) + u_y \cos\left(\frac{a_{uy}\pi y}{L}\right) + u_z \cos\left(\frac{a_{uz}\pi z}{L}\right) \\
v(x,y,z) &= v_0 + v_x \cos\left(\frac{a_{vx}\pi x}{L}\right) + v_y \sin\left(\frac{a_{vy}\pi y}{L}\right) + v_z \sin\left(\frac{a_{vz}\pi z}{L}\right) \\
w(x,y,z) &= w_0 + w_x \sin\left(\frac{a_{wx}\pi x}{L}\right) + w_y \sin\left(\frac{a_{wy}\pi y}{L}\right) + w_z \cos\left(\frac{a_{wz}\pi z}{L}\right) \\
p(x,y,z) &= p_0 + p_x \cos\left(\frac{a_{px}\pi x}{L}\right) + p_y \sin\left(\frac{a_{py}\pi y}{L}\right) + p_z \cos\left(\frac{a_{pz}\pi z}{L}\right)
\end{aligned} \tag{A.1}$$

Appendix B: Manufactured Solution Constants

Constants employed include $L = 1\ m$, $\gamma = 1.4$, and $R = 287.0\ Nm/(kg{\cdot}K)$. For the supersonic Navier-Stokes calculations, additional constants include the absolute viscosity $\mu = 20.\ N{\cdot}s/m^2$ and the Prandtl number $Pr = 1.0$, while for the subsonic Navier-Stokes calculations, the viscosity was chosen as $\mu = 10.\ N{\cdot}s/m^2$. Note, the ϕ constants all have the same dimensions as the primitive variable (listed in the first column), and the a constants are dimensionless.

Table A.1 Constants for 2D Euler Supersonic Manufactured Solution

Equation, ϕ	ϕ_0	ϕ_x	ϕ_y	ϕ_z	$a_{\phi x}$	$a_{\phi y}$	$a_{\phi z}$
ρ *(kg/m³)*	1.	0.15	-0.1	0.	1.	0.5	0.
u (m/s)	800.	50.	-30.	0.	1.5	0.6	0.
v (m/s)	800.	-75.	40.	0.	0.5	2./3.	0.
w (m/s)	0.	0.	0.	0.	0.	0.	0.
p (N/m²)	$1.\times10^5$	0.2×10^5	0.5×10^5	0.	2.	1.	0.

Table A.2 Constants for 2D Euler Subsonic Manufactured Solution

Equation, ϕ	ϕ_0	ϕ_x	ϕ_y	ϕ_z	$a_{\phi x}$	$a_{\phi y}$	$a_{\phi z}$
ρ *(kg/m³)*	1.	0.15	-0.1	0.	1.	0.5	0.
u (m/s)	70.	5.	-7.	0.	1.5	0.6	0.
v (m/s)	90.	-15.	8.5	0.	0.5	2./3.	0.
w (m/s)	0.	0.	0.	0.	0.	0.	0.
p (N/m²)	$1.\times10^5$	0.2×10^5	0.5×10^5	0.	2.	1.	0.

Table A.3 Constants for 3D Euler Supersonic Manufactured Solution

Equation, ϕ	ϕ_0	ϕ_x	ϕ_y	ϕ_z	$a_{\phi x}$	$a_{\phi y}$	$a_{\phi z}$
ρ *(kg/m^3)*	1.	0.15	-0.1	-0.12	1.	0.5	1.5
u (m/s)	800.	50.	-30.	-18.	1.5	0.6	0.5
v (m/s)	800.	-75.	40.	-30.	0.5	2./3.	1.25
w (m/s)	800.	15.	-25.	35.	1./3.	1.5	1.
p (N/m^2)	$1.\times10^5$	0.2×10^5	0.5×10^5	-0.35×10^5	2.	1.	1./3.

Table A.4 Constants for 2D Navier-Stokes Supersonic Manufactured Solution

Equation, ϕ	ϕ_0	ϕ_x	ϕ_y	ϕ_z	$a_{\phi x}$	$a_{\phi y}$	$a_{\phi z}$
ρ *(kg/m^3)*	1.	0.15	-0.1	0.	1.	0.5	0.
u (m/s)	800.	50.	-30.	0.	1.5	0.6	0.
v (m/s)	800.	-75.	40.	0.	0.5	1.5	0.
w (m/s)	0.	0.	0.	0.	0.	0.	0.
p (N/m^2)	$1.\times10^5$	0.2×10^5	0.5×10^5	0.	2./3.	1.	0.

Table A.5 Constants for 2D Navier-Stokes Subsonic Manufactured Solution

Equation, ϕ	ϕ_0	ϕ_x	ϕ_y	ϕ_z	$a_{\phi x}$	$a_{\phi y}$	$a_{\phi z}$
ρ *(kg/m^3)*	1.	0.1	0.15	0.	0.75	1.0	0.
u (m/s)	70.	4.	-12.	0.	5./3.	1.5	0.
v (m/s)	90.	-20.	4.	0.	1.5	1.0	0.
w (m/s)	0.	0.	0.	0.	0.	0.	0.
p (N/m^2)	$1.\times10^5$	-0.3×10^5	0.2×10^5	0.	1.0	1.25	0.

Appendix C: Example Source Term

A sample source term is given below for the 2D mass conservation equation (Eq. (1)). The source terms is the result of analytically differentiating the general Manufactured Solution given in Appendix A for ρ, u, and v.

$$\begin{aligned}\text{Mass Equation Source} = {} & \frac{a_{ux}\pi u_x}{L}\cos\left(\frac{a_{ux}\pi x}{L}\right)\left[\rho_0 + \rho_x\sin\left(\frac{a_{\rho x}\pi x}{L}\right) + \rho_y\cos\left(\frac{a_{\rho y}\pi y}{L}\right)\right] \\ & + \frac{a_{vy}\pi v_y}{L}\cos\left(\frac{a_{vy}\pi y}{L}\right)\left[\rho_0 + \rho_x\sin\left(\frac{a_{\rho x}\pi x}{L}\right) + \rho_y\cos\left(\frac{a_{\rho y}\pi y}{L}\right)\right] \\ & + \frac{a_{\rho x}\pi\rho_x}{L}\cos\left(\frac{a_{\rho x}\pi x}{L}\right)\left[u_0 + u_x\sin\left(\frac{a_{ux}\pi x}{L}\right) + u_y\cos\left(\frac{a_{uy}\pi y}{L}\right)\right] \\ & - \frac{a_{\rho y}\pi\rho_y}{L}\sin\left(\frac{a_{\rho y}\pi y}{L}\right)\left[v_0 + v_x\cos\left(\frac{a_{vx}\pi x}{L}\right) + v_y\sin\left(\frac{a_{vy}\pi y}{L}\right)\right]\end{aligned} \tag{A.2}$$