# Code verification of boundary conditions for compressible and incompressible computational fluid dynamics codes[☆]

Aniruddha Choudhary[a,*], Christopher J. Roy[a], Edward A. Luke[b], Subrahmanya P. Veluri[c]

[a]*Aerospace and Ocean Engineering Department, Virginia Tech, Blacksburg, VA 24061, USA*
[b]*Computer Science and Engineering Department, Mississippi State University, Mississippi State, MS 39762, USA*
[c]*ANSYS Inc., Lebanon, NH 03766, USA*

**Abstract**

To establish confidence in the code, a rigorous assessment of boundary condition implementation is necessary. In this work, techniques are presented for performing code verification of boundary conditions commonly used in compressible and incompressible Computational Fluid Dynamics (CFD) codes. Using a compressible CFD code, this study assesses the subsonic inflow (isentropic and fixed-mass), subsonic outflow, supersonic outflow, no-slip wall (adiabatic and isothermal), and inviscid slip-wall. A novel approach is introduced to determine manufactured solutions for boundary condition verification when the velocity-field is constrained to be divergence-free during the simulation in an incompressible CFD code. The use of simplified curved surfaces is proposed for easier generation of manufactured solutions during the verification of certain boundary conditions involving many constraints. To perform rigorous code verification, general grids with mixed cell types at the verified boundary are used. It is found that the use of planar boundaries or only hexahedral cells at the verified boundary can mask sources of errors in the boundary condition implementation.

*Keywords:* Code verification, Method of manufactured solutions, Order of accuracy, Boundary conditions

## 1. Introduction

With increased use of computational tools for fluid dynamics simulations, it becomes important to perform verification of various aspects of a computational simulation such as conservation equations, closure models, and boundary conditions. Verification deals with the mathematics of the simulation and involves assessing the correctness of the computer code and numerical algorithms, as well as the accuracy of the numerical solution. There are two fundamental aspects to verification: code verification and solution verification. Code verification is the process of ensuring, to the degree possible, that there are no coding mistakes (bugs) or algorithm inconsistencies. Solution verification focuses on identifying and estimating various kinds of errors present in numerical simulations due to round-off, statistical sampling, iteration, and discretization error.

Different criteria for performing code verification, in order of increasing rigor, are: simple tests, code-to-code comparisons, discretization error quantification, convergence tests, and order of accuracy tests [1]. The order of accuracy test determines whether or not the discretization error is reduced at an expected rate given by the formal order of the implemented numerical scheme. Evaluation of discretization error requires knowing the exact solution for the governing equations which is generally not known for problems of practical interest. A technique called the method of manufactured solutions (MMS) can be used where a solution is "manufactured" and used as an exact solution [2]. This manufactured solution exactly solves the modified governing equations consisting of the MMS source terms, where the MMS source terms are

[☆]Distribution statement (to be decided)
[*]Corresponding author: aniruddhac@gmail.com, +1-540-239-4824

generated by substituting the manufactured solution in the original governing equations. The method of manufactured solutions was first applied for code verification by Roache and Steinberg [2] where it was used to verify a code for generation of 3D transformations for elliptic partial differential equations (PDEs). The books by Knupp and Salari [3], Roache [4], and Oberkampf and Roy [5] provide a comprehensive discussion of code verification, order of accuracy testing, and MMS.

The current work is focused on the code verification of boundary conditions (BCs) commonly used in compressible and incompressible computational fluid dynamics (CFD) simulations. Knupp and Salari [3] presented a detailed account of MMS-based code verification for incompressible and compressible Navier-Stokes codes. Their work addressed the verification of Dirichlet BC (where exact value from the manufactured solution is specified), Neumann BC (where exact value of the gradient from the manufactured solution is specified), and mixed BC (a combination of Dirichlet and Neumann BCs). In this method the user specifies the exact values from the manufactured solutions at the boundaries instead of allowing the boundary condition implementation of the code to evaluate the boundary solution. This method is useful to accomplish code verification of governing equations inside the computational domain, and simple BCs (Dirichlet, Neumann, or mixed). For specialized boundary conditions used in practical flow simulations such as various kinds of wall, inflow, and outflow BCs, the authors in [3] suggested a more rigorous approach. When a boundary condition requires that a solution variable must meet a specific criterion on that boundary then the manufactured solution must be constructed adhering to that criterion. For example, for a free-slip wall BC implementation where the normal velocity component at the wall boundary is required to be zero the manufactured solution must be constructed such that its normal velocity component is zero at that boundary.

Bond et al. [6] presented detailed methodology for MMS-based boundary condition code verification. Their analysis included slip-wall BC, no-slip wall BCs (adiabatic and isothermal), and outflow BCs (subsonic, supersonic, and mixed) over non-orthogonal computational domains with hexahedral cells. In [6], a finite volume, unstructured, compressible CFD code, SIERRA/Premo [7] was used to verify the slip-wall BC and outflow BCs (subsonic and supersonic) using Euler equations, and the no-slip wall BCs (adiabatic and isothermal) using Navier-Stokes and RANS equations. Veluri et al. [8] verified the no-slip wall BCs (adiabatic and isothermal), slip-wall BC, isentropic subsonic inflow BC, subsonic outflow BC, and supersonic outflow BC for an unstructured, compressible CFD code, Loci/CHEM [9], using the method of Bond et al. [6]. However, many of these tests employed manufactured solutions to work on meshes with planar boundary surfaces. In [8], the boundary conditions which were verified on curved surfaces (i.e., no-slip wall BC and slip-wall BC) used hexahedral cells at these surfaces. In [10], we presented the use of simple curved surfaces (a circular arc in 2D or a spherical cap in 3D) with hexahedral cells at the boundary for verification of inflow BCs (subsonic isentropic and subsonic fixed-mass) and outflow BCs (subsonic and supersonic). Recently, Folkner et al. [11] suggested an approach where the different boundary conditions are verified along with the flow governing equations using a single manufactured solution. Their approach involves a selection matrix-based unifying framework for implementation of different boundary conditions which is yet to be seen in widespread use.

In the current work we accomplish four objectives. First, we apply the method of [6] to different inflow, outflow, and wall BCs identifying the mathematical constraints for derivation of manufactured solutions for each boundary condition separately. Second, we demonstrate the use of boundary surfaces in the form of spherical caps to verify certain boundary conditions which makes the derivation of manufactured solutions under constraints much simpler than the use of general boundary surfaces. Third, we present the use of grids with boundary surfaces having cells of mixed types to verify boundary condition in a rigorous manner that could uncover implementation errors masked by using boundaries having cells of the same types. Fourth, we introduce two new methods to derive manufactured solutions under the constraint of divergence-free velocity field.

The paper is organized as follows: In Sec. 2, we present the governing equations and details of the flow solvers used for analysis. In Sec. 3, we give an overview of the methodology used for MMS-based code verification of boundary conditions. Sec. 4 presents different grid types and boundary surfaces used for rigorous code verification of boundary conditions. Sec. 5 presents results of this work including: boundary constraints, derivation of manufactured solutions, results of order of accuracy tests, some issues uncovered, and a discussion of our findings for each of the cases. Finally, we summarize our work in Sec. 6. Before proceeding further, we address what boundary conditions and what is it that we are attempting to verify with code verification procedures.

### *1.1. A note on boundary conditions in CFD*

Physically, boundary conditions at the domain boundary determine the interaction between the physical phenomenon in the domain interior and the domain exterior. Mathematically, boundary conditions are mathematical relations that, together with the initial conditions, can ensure (at least locally) that the initial boundary value problem (IBVP) for Navier-Stokes and/or Euler equations is well-posed (i.e., a stable, unique solution exists). Simple extrapolation of variables at the domain boundary is not sufficient as it does not account for the physical phenomenon and the mathematical nature of the governing equations at the boundary [12]. Thompson [13, 14] defined a time-dependent boundary condition for a time-dependent IBVP as a precise mathematical relation that can be used to determine $\partial\vec{U}/\partial t$, where $\vec{U}$ is the conserved variable vector, along the domain boundary and is not obtainable solely from the governing equation of $\vec{U}$ in the domain interior. For 1D Euler equations, a compatible set of boundary conditions can be derived that would ensure a unique determination of $\partial\vec{U}/\partial t$ at the domain boundary. Navier-Stokes equations have mixed hyperbolic-parabolic character and the assessment of well-posedness is not straightforward. In general, the determination of compatible boundary conditions for IBVPs (here, Euler and Navier-Stokes equations) does not guarantee an existence of a stable, unique solution. See [15] for a summary of boundary conditions that may ensure a well-posed problem.

Efficient and robust techniques based upon mathematical theory and application based evidence are widely used to determine boundary conditions in CFD. For example, at the wall, one generally imposes no-slip conditions on velocity for Navier-Stokes equations along with either isothermal or adiabatic conditions for the state variables. For a complete description of different boundary conditions, the interested readers are referred to the pioneering works by Yee [16], Yee, Beam and Warming [17], and the book by Hirsch [12]. In the present work, the FUN3D technical note [18] and the Loci/CHEM user's guide [19] are referred to for an overview of the implementation techniques of different boundary conditions.

The various aspects to be considered while selecting, implementing, and assessing a boundary condition can be summarized as follows [20]:

1. Physical: A boundary condition must satisfy the physical phenomenon. For example, for 1D Euler equations, at a subsonic inflow, the physics of the boundary imposes, (a) $\vec{v}\cdot\vec{n} = 0$, and (b) $|\vec{v}\cdot\vec{n}| < c$, where $\vec{v}$ is the velocity vector, $\vec{n}$ is the outward normal at the boundary, and $c$ is the local speed of sound.
2. Mathematical: A boundary condition should be mathematically well-posed. For example, for 1D Euler equations, at a subsonic inflow, an analysis using compatibility conditions (see [21] for derivation) shows that (a) two variables must be defined, and (b) information on density must be directly or indirectly obtainable for the problem to be well-posed. This suggests a use of either a fixed-mass inflow (mass flow rate and density specified) or an isentropic inflow (total pressure and total temperature specified) formulation amongst others. For 2D, and 3D flows, information on flow directionality must either be provided or assumed at the subsonic inflow.
3. Numerical: Includes numerical approximation of the mathematical condition (e.g.,one-sided extrapolation), and additional numerical requirements. Stability and accuracy of the boundary scheme must also be assessed.
4. Application: With all above aspects, the BC must ultimately be satisfactorily efficient and general to handle a wide range of application-based flow problems.

For cell-centered finite-volume methods, the boundary flux is usually developed using either a standard weak formulation or a scheme-consistent weak formulation. In a standard (or a scheme-inconsistent) formulation the boundary flux is directly computed at the boundary making the boundary scheme different than the finite-volume scheme applied at the interior faces. In a scheme-consistent formulation, the boundary condition is applied at the ghost cells (i.e., computational cells just outside the edge of the physical domain) which is used along with the interior scheme to compute the boundary flux [22]. Whether a standard or a scheme-consistent formulation has been implemented, in as far as boundary condition code verification is concerned, the goal is to verify its implementation in the code as intended by the developer. This requires us to identify the relevant physical, mathematical, and numerical constraints of a boundary condition while constructing the manufactured solutions. For example, a one-sided extrapolation of a solution variable to the boundary is usually formulated based upon a finite-difference approximation of a derivative condition such as zero gradient. In this case, the zero gradient condition becomes one of the constraints under which the manufactured solution must be derived. Together, the physical, mathematical, and numerical constraints of a boundary condition are referred to as boundary constraints and are the primary objects during our analysis.

## 2. Codes and governing equations

### *2.1. Compressible solver: Loci/CHEM*

For the discussion of subsonic inflow, outflow, and wall boundary conditions using compressible flows, we use Loci/CHEM [9] which is a finite-volume, compressible CFD code for generalized 3D unstructured grids and employs a density-based solver capable of solving turbulent, multiphase, multispecies, chemically reacting 3D flows. Loci/CHEM is developed upon the Loci framework, a rule-based programming framework developed for large-scale, massively parallel, multidisciplinary simulations [23].

The governing equations for CHEM include the 3D unsteady Favre-averaged Navier-Stokes equations (referred to herein as baseline governing equations) and several turbulence models. The steady-state form of equations is obtained from the baseline governing equations by setting the time derivative terms to zero. The Euler equations are obtained by removing the viscous terms, while laminar Navier-Stokes equations are obtained by specifying constant viscosity and neglecting turbulence models in the baseline governing equations. The 3D, laminar Navier-Stokes equations along with the constitutive relations as verified in this work are given in Eqs. 1-6 using Cartesian tensor notations, where $i$, $j$, and $k$ correspond to the components in the three coordinate directions, $x$, $y$ and $z$ (other symbols have usual thermodynamic connotations).

Continuity:

$$\frac{\partial \rho u_j}{\partial x_j} = 0 \tag{1}$$

Momentum:

$$\frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \tag{2}$$

Energy:

$$\frac{\partial \rho e_T u_j}{\partial x_j} = -\frac{\partial q_j}{\partial x_j} - \frac{\partial p u_j}{\partial x_j} + \frac{\partial \tau_{jk} u_k}{\partial x_j} \tag{3}$$

Stress tensor:

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3}\mu \left( \frac{\partial u_i}{\partial x_i} \right) \delta_{ij} \tag{4}$$

Laminar heat flux:

$$q_j = -\frac{\mu}{Pr} c_p \frac{\partial T}{\partial x_j} \tag{5}$$

Thermodynamic relations:

$$h_T = e + \frac{u_i u_i}{2} + \frac{p}{\rho}, \quad e = c_v T, \quad p = \rho R T \tag{6}$$

Formally second order upwind flux-differencing schemes of the Roe family are employed with flux reconstruction at the cell center using weighted least-square gradient technique. Loci/CHEM is capable of handling generalized grids, i.e., grids composed of arbitrary polyhedra, including tetrahedra, prisms, pyramids, and hexahedra, making adaptive mesh refinement feasible. Loci/CHEM has been comprehensively verified to be second order for baseline governing equations, unsteady time-stepping accuracy, and some boundary conditions [8, 24–26].

### *2.2. Incompressible solver: MFIX/TFM*

For the verification of no-slip wall BC, slip wall BC, and pressure outflow BC using incompressible flows, we use MFIX (Multiphase Flow with Interphase eXchanges) [27] which is an open-source, general-purpose, hydrodynamic code that can simulate heat transfer and chemical reactions for fluids containing multiple solid phases. It is generally used for flows common in energy conversion and chemical reactor processes such as bubbling, circulating, and spouted fluidized beds. Both the continuum and discrete approaches to model the multiphase interactions are available in MFIX. In a continuum approach, the different phases are mathematically described as interpenetrating continua. The continuum approach is also referred to as the Eulerian-Eulerian method or the two-fluid method (TFM). Methods where the carrier (or surrounding) phase is treated as a continuum and the dispersed phase is treated as discrete entities (i.e., particles or parcels of particles) are called Lagrangian-Eulerian methods or Continuum Discrete Methods.

In the current discussion of boundary condition verification techniques, we are concerned with the two-fluid model with the assumption of steady-state, single-phase flow. In this sense, the MFIX governing equations reduce simply to the incompressible Navier-Stokes equations where the continuity equation is given by Eq. 1 and the momentum conservation equation is given by Eq. 2 but with the density term, $\rho$, treated as a constant. This is a description of only the relevant governing equations for single-phase, incompressible flows as verified in the current work. For a detailed description of multiphase governing equations and models in MFIX, see [28].

MFIX uses formally second-order centered and upwinding methods for spatial discretization featuring a staggered-grid approach where momentum variables (e.g., fluid and solid velocities) reside at the face centers while scalar variables (e.g., pressure, temperature, volume-fraction) reside at the cell centers. MFIX uses a modified SIMPLE-based algorithm and employs a pressure projection method imposing a divergence-free velocity for incompressible flow [29]. The two-fluid model of MFIX has been verified to be second order accurate for the single-phase and two-phase baseline governing equations on 3D, stretched Cartesian meshes using the centered discretization scheme [30].

## 3. Methodology

The order of accuracy test for code verification determines whether the observed order of accuracy for the numerical scheme matches the formal order of accuracy as the mesh is refined into the asymptotic range. The formal order is obtained from a truncation error analysis of the discrete equations [1]. The observed order is determined directly from the simulations by observing the rate at which the norms (e.g., $L_1, L_2, L_\infty$) of the solution discretization error decrease as the grid is systematically refined. The asymptotic range is that range of discretization sizes (e.g., $\Delta x$, $\Delta t$) where the higher order terms of the truncation and discretization error are negligible compared to the lowest order term (i.e., the term which defines the formal order of the scheme). The observed order of accuracy will generally fail to match the formal order when there is a coding mistake (bug), any algorithm inconsistency, a discontinuity in the solution, large iterative or round-off errors, or if the solution is not in the asymptotic range.

To calculate the observed order of accuracy, the numerical solution must be calculated at multiple grid levels. The number of grid levels required depends on whether the solution is in the asymptotic range and usually varies from one problem to another. Systematic mesh refinement [5] is defined as uniform and consistent refinement over the spatial domain. Uniform refinement ensures that the mesh has been refined along all coordinate directions equally over the entire domain and consistent refinement ensures that the mesh quality either stays constant or improves with mesh refinement.

Systematic mesh refinement for structured grids is usually performed by selecting the finest grid and then uniformly coarsening the grid along all the coordinate directions by the factor selected for grid refinement. Alternatively, systematically refined mesh levels can be obtained by selecting a consistent set of mesh transformation equations from an evenly spaced computational space $(\xi, \eta, \zeta)$ to the physical space $(x, y, z)$, performing mesh refinement on the computational domain, and then applying the mesh transformation to the refined computational domain. If the transformation functions used are continuous then such a refinement is systematic. This latter process is used in the current study to generate hybrid unstructured systematic meshes by starting from a family of systematically refined structured meshes, then splitting structured cells from parts of the domain into multiple unstructured cells. This technique was previously implemented in generating sets of systematically-refined unstructured grids by Veluri et al. [8].

A few compressible CFD codes (e.g, Premo [6, 31], WIND [31], and Loci/CHEM [8, 26]) and an incompressible code (MFIX [30]) have been comprehensively verified using MMS applying the systematic mesh refinement approach. Herbert and Luke [25] used an alternative, statistical approach to MMS which employs grid shrinking and successfully verified the Loci/CHEM CFD code for multi-species, laminar Navier-Stokes equations using both statistical and traditional MMS. Thomas et al. [32] used a similar approach of computational windows to isolate and verify specific elements of the computational scheme applied to the FUN3D CFD code. These are only a few examples of MMS applied to CFD codes (see Refs. [3–5] for more examples).

The procedure for applying MMS with order of accuracy test is summarized as follows [1, 5]:

1. Choose the form of the governing equations.
2. Choose the form of the manufactured solution.
3. Derive the modified governing equations.
4. Solve the discrete form of the modified governing equations on multiple, systematically-refined meshes.
5. Evaluate the global discretization error in the numerical solution.
6. Apply the order of accuracy test to determine if the observed order of accuracy matches the formal order.

MMS is based upon the philosophy that code verification deals with mathematics of a given problem and hence arbitrary functions (with certain continuity and other requirements) can be selected as exact solutions. Manufactured solutions need not be physically realistic, but should be selected based upon certain criteria outlined below [1, 5]:

1. Manufactured solutions should be smooth, analytic functions of spatial (or temporal, in case of time accuracy verification) coordinates.
2. The derivatives of a manufactured solution should not vanish within the domain, including the cross-derivative terms (if they appear in the governing equations).
3. Care should be taken that one term does not dominate the other terms (e.g., a manufactured solution for a Navier-Stokes code verification should be such that the convective and the diffusive terms are of the same order of magnitude).
4. The manufactured solutions should be realizable within the code (e.g., non-positive temperature values in the MS cannot be accepted if the code uses square root of temperature to calculate the speed of sound).

The manufactured solution for each boundary condition is generated by modifying the baseline manufactured solution to satisfy the boundary constraints. The baseline manufactured solution selected is a combination of sine and cosine functions and takes the following general form

$$\begin{aligned}\phi\left(x,y,z\right) = \phi_0 + \phi_x f_{\phi x}\left(\frac{a_{\phi x}\pi x}{L}\right) + \phi_y f_{\phi y}\left(\frac{a_{\phi y}\pi y}{L}\right) + \phi_z f_{\phi z}\left(\frac{a_{\phi z}\pi z}{L}\right) + \\ \phi_{xy} f_{\phi xy}\left(\frac{a_{\phi xy}\pi xy}{L^2}\right) + \phi_{yz} f_{\phi yz}\left(\frac{a_{\phi yz}\pi yz}{L^2}\right) + \phi_{zx} f_{\phi zx}\left(\frac{a_{\phi zx}\pi zx}{L^2}\right)\end{aligned} \tag{7}$$

where $L$ represents a characteristic length, $\phi = [\rho, u, v, w, p]^T$ represents manufactured solutions for any of the primitive variables, and the functions, $f_{\phi x}(\cdot)$, $f_{\phi y}(\cdot)$, etc. represent sine or cosine functions. The sinusoidal functions used in the current work are given in Table A.1. The constants, $\phi_0$, $\phi_x$, $a_{\phi x}$, etc. are selected based upon the criteria for selecting manufactured solutions mentioned before. Since trigonometric functions have been used, it is ensured that there is reasonable periodicity of solutions and their derivatives within the domain. (See Ref. [3–5] for more examples of continuous manufactured solutions.) The characteristic length parameter can be used to modify the periodicity of manufactured solutions and their derivatives within the domain. In the current study, we select the characteristic length as equal to the domain length. The manufactured solution constants used for compressible flow and incompressible flow cases are given in Table A.2 and Table A.3, respectively. The 2D manufactured solutions can be obtained from the 3D manufactured solutions (Eq. 7) by removing the terms involving the third coordinate direction.

The general approach adapted in this work to obtain manufactured solutions for the purpose of code verification of boundary conditions is based upon the approach developed by Bond et al. [6]. The method primarily involves multiplying a baseline standard manufactured solution as described in Eq. 7 by a function representing the boundary surface. This procedure is explained here with the help of a simple example in 2D. A baseline 2D manufactured solution for the steady-state form of governing equations can be written from Eq. 7 as follows

$$\phi\left(x,\ y\right) = \phi_0 + \phi_x f_{\phi x}\left(\frac{a_{\phi x}\pi x}{L}\right) + \phi_y f_{\phi y}\left(\frac{a_{\phi y}\pi y}{L}\right) + \phi_{xy} f_{\phi xy}\left(\frac{a_{\phi xy}\pi xy}{L^2}\right) \tag{8}$$

Suppose the goal is to derive a manufactured solution for $\phi$ such that it satisfies the following conditions at a given boundary expressed as $F\left(x,y\right) = C$ in the 2D domain, where $C$ is some scalar constant:

1. $\phi = \phi_0$, and
2. the derivatives of $\phi$ normal to the boundary up to order $m-1$ are zero.

The manufactured solution satisfying these conditions is obtained by multiplying the sinusoidal parts of the manufactured solution from Eq. 8 with $\left(C - F\left(x,y\right)\right)^m$ as shown in Eq. 9

$$\begin{aligned}\phi_{BC}\left(x,y\right) = \phi_0 + \left(C - F\left(x,y\right)\right)^m \\ \left(\phi_x f_{\phi x}\left(\frac{a_{\phi x}\pi x}{L}\right) + \phi_y f_{\phi y}\left(\frac{a_{\phi y}\pi y}{L}\right) + \phi_{xy} f_{\phi xy}\left(\frac{a_{\phi xy}\pi xy}{L^2}\right)\right)\end{aligned} \tag{9}$$

In further discussion and derivations of boundary condition manufactured solutions, we specify the baseline manufactured solution of Eq. 7 with the following notation

$$\phi = \phi_0 + \phi_1 \tag{10}$$

where $\phi_0$ is the base term, and $\phi_1$ is the variable term consisting of sinusoidal functions. The base term, $\phi_0$, can be a constant or a variable function depending on the requirements of the boundary condition being verified.

The MMS-based code verification approach discussed here can be applied to both compressible and incompressible flow codes given the selected manufactured solutions meet the criteria described earlier (i.e., smooth, analytic, continuous, balanced, and realizable). Note that, there is no separate evolution equation for pressure and temperature in the Navier-Stokes equations. This is addressed differently in compressible flows than in incompressible flows. For compressible flows, the continuity equation is a transport equation for density which is directly related to the fluid pressure in the flow via the equation of state. For incompressible flows, the density is constant and the continuity equation is simply a constraint on the velocity field which is expressed as the divergence free condition. MFIX employs a SIMPLE-based algorithm [33, 34] to solve for pressure using a pressure-correction (or pressure-projection) method. Assuming an initial pressure field, the momentum equations are solved to obtain an intermediate velocity field. A pressure-correction equation is then formulated and solved under the assumption of continuity (divergence-free flow) and momentum conservation. The pressure-correction solution is then used to correct the initial pressure field and the intermediate velocity field. This process is iteratively repeated until all conservation equations are satisfied within a given tolerance. Though possible, it would require modifications at several steps in the MFIX code to allow for a general manufactured solution to be used for code verification. Modification of important steps of the algorithm is not advised during a code verification exercise. In other words, a manufactured solution without a divergence-free velocity field is not effectively realizable within this code violating one of the criteria for selection of manufactured solutions.

To derive the manufactured solutions for 3D flows satisfying the divergence-free condition, we propose a new curl-based approach. Since the divergence of the curl of a 3D vector field is identically zero, one can select the manufactured solution for the 3D velocity field, $\vec{V}$, as

$$\vec{V} = \nabla \times \vec{H} \tag{11}$$

where $\vec{H} = \{u(x,y,z), v(x,y,z), w(x,y,z)\}^T$ is a general 3D vector field consisting of functions of the form described in Eq. 7. The pressure manufactured solution is selected directly using Eq. 7. Furthermore, for verification of boundary conditions with incompressible flows, the manufactured solutions must be constructed with boundary constraints as well as the divergence-free constraint which is discusses in Sec.5.

## 4. Mesh types

For rigorous boundary condition code verification, the most general grid on which a boundary condition should be verified must have

1. a surface function for the boundary with sufficient curvature,
2. cells with variable quality metrics (skewness, stretching, etc.), and
3. cells of different types (hexahedral, tetrahedral, prismatic cells) at the boundary.

For compressible flow verification using Loci/CHEM, the 3D hybrid grids consist of hexahedral cells, tetrahedral cells, prismatic cells, and curved surfaces as boundaries (with skewness and stretching). Fig. 1(a) shows a general hybrid mesh used in this work for which the mesh generation equations are given as

$$\begin{aligned} &x = 0.03\sin(2\pi\eta) + 0.03\sin(2\pi\zeta) + 0.04\,\eta + 0.05\,\zeta + \xi, \\ &y = -0.03\sin(2\pi\xi) - 0.1\xi + \eta, \quad \text{and} \quad z = -0.03\sin(2\pi\xi) - 0.1\xi + \zeta \end{aligned} \tag{12}$$

where $\xi$, $\eta$, and $\zeta$ are coordinates on an evenly spaced grid of unit dimensions. The computational domain for all the grids discussed in this work assumes a domain reference length of $L = 1$ m in each coordinate direction. The inflow and wall BCs are tested on the $S_{\xi 0}$ surface as defined in Eq. 13 which is obtained by substituting $\xi = 0$ into Eq. 12. Whereas, the outflow BCs are tested on the $S_{\xi 1}$ surface as defined in Eq. 14 which is obtained by substituting $\xi = 1$ into Eq. 12.

$$S_{\xi 0}(x,y,z) \equiv x - 0.03\sin(2\pi y) - 0.03\sin(2\pi z) - 0.04y + 0.05z = 0 \tag{13}$$

$$\begin{aligned} S_{\xi 1}(x,y,z) \equiv x &- 0.03\sin(2\pi(y+0.1)) - 0.03\sin(2\pi(z+0.1)) \\ &- 0.04(y+0.1) - 0.05(y+0.1) - 1 = 0 \end{aligned} \tag{14}$$

The numerical constants selected in Eq. 12 ensure sufficient curvature for the surface to be tested. For example, the difference in maximum and minimum elevation over the 3D wave representing $S_{\xi 0}$ as shown in Fig. 1(b) is about 9% of the domain length. Grids with even more complexity (e.g., larger cell quality

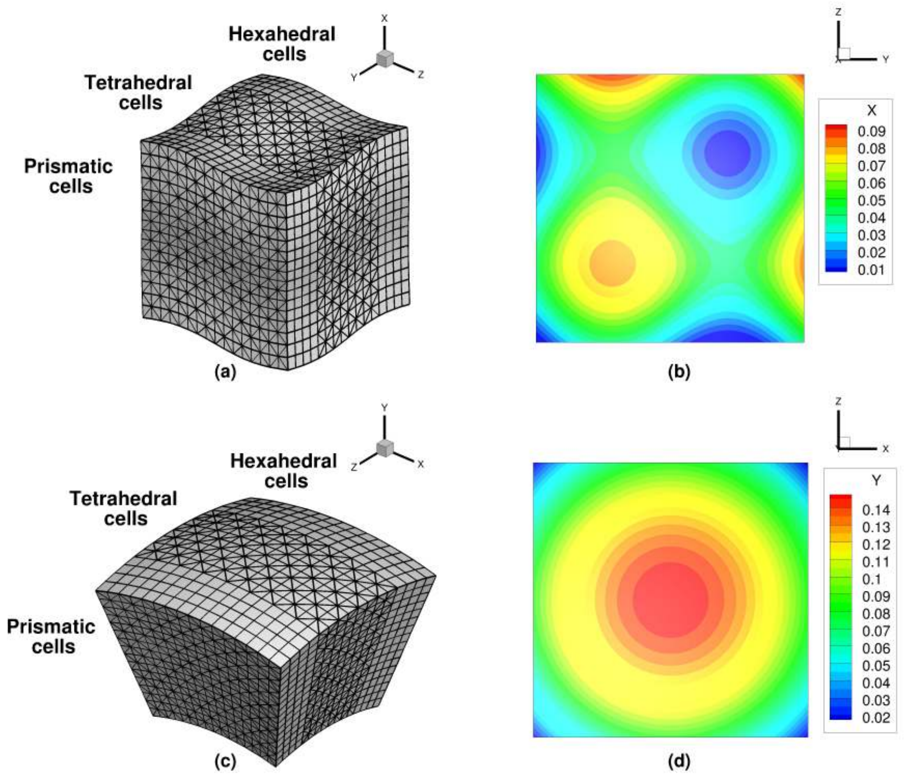

Figure 1: Grids used for boundary condition verification with the compressible solver: (a) general 3D mesh with hybrid boundaries at $\xi = 0$ and $\xi = 1$, (b) $S_{\xi 0}$ boundary for the general 3D mesh, (c) annular 3D mesh with hybrid boundaries at $r = r_{min}$ and $r = r_{max}$, and (d) $S_{r0}$ boundary for the annular 3D mesh. Note: $\xi$ corresponds to $x$ in subfigures (a)(b) while $r$ corresponds to $y$ in subfigures (c)(d).

variations) are possible, but may lead to iterative non-convergence and/or require unfeasibly large number of mesh nodes to achieve asymptotic convergence.

The boundary conditions addressed here require multiple constraints to be satisfied by the manufactured solution at the curved surface. A general curved boundary as shown in Fig. 1(b) makes it difficult to correctly and intuitively devise a manufactured solution for some of the boundary conditions (discussed in Sec. 5). An annular hybrid grid where the verified boundary surfaces follow equations of a spherical cap as shown in Fig. 1(c) is used for easy selection of manufactured solutions. The mesh nodes are generated using the following set of equations

$$\begin{aligned} &x = 0.5 + r\sin(\theta), \quad z = 0.5 + r\sin(\gamma), \quad \text{and} \\ &y = -1.8 + \sqrt{(x-0.5)^2 + (z-0.5)^2 + r^2}, \end{aligned} \tag{15}$$

where $\theta$ and $\gamma$ each vary uniformly in the range $[-\pi/12, \pi/12]$, and $r$ varies uniformly in the range $[1.95, 2.95]$. For the annular hybrid mesh, the inflow and wall BCs are tested on the $S_{r0}$ surface as defined in Eq. 16 while the outflow BCs are tested on the $S_{r1}$ surface as defined in Eq. 17. The subscripts "$r0$" and "$r1$" in Eqs. 16 and 17 correspond to the substitution of $r = r_{min} = 1.95$ and $r = r_{max} = 2.95$ into Eq. 15, respectively. It is evident that these spherical cap surfaces are parts of a sphere centered at $(x, y, z) = (0.5, -1.8, 0.5)$. The difference in maximum and minimum elevation over the spherical cap as shown in Fig. 1(d) is about 13% of the domain length.

$$S_{r0}(x,y,z) \equiv \sqrt{(x-0.5)^2 + (y+1.8)^2 + (z-0.5)^2} - 1.95 = 0 \tag{16}$$

$$S_{r1}(x,y,z) \equiv \sqrt{(x-0.5)^2 + (y+1.8)^2 + (z-0.5)^2} - 2.95 = 0 \tag{17}$$

The mesh nodes generated from Eq. 12 for the general grid and from Eq. 15 for the annular grid are then appended with appropriate nodal connectivity information such that mixed type cells are present on the boundaries under consideration. In this process, no new nodes are added, 50% of the hexahedral cell are each subdivided into five tetrahedral cells each, and 25% of the hexahedral cells are each subdivided into two prismatic cells each. This process is also used to generate grids with all hexahedral, all tetrahedral, or all prismatic cells where necessary.

MFIX employs a structured, staggered-grid solver and is equipped to work with 2D and 3D Cartesian grids with stretched cells. The MFIX algorithm under test is currently not equipped to work with grids having curved, skewed or rotated cells. For incompressible, divergence-free flow verification using MFIX, the 2D and 3D stretched Cartesian grids selected are shown in Fig. 2(a) and Fig. 2(b), respectively. The mesh nodes are generated using an internal layer grid equation as

$$\begin{aligned} &x = \xi + 2.5(0.4-\xi)(1-\xi)\xi, \quad y = \eta + 2.5(0.4-\eta)(1-\eta)\eta \quad \text{and} \\ &z = \zeta + 2.5(0.4-\zeta)(1-\zeta)\zeta, \end{aligned} \tag{18}$$

where $\xi$, $\eta$, and $\zeta$ are coordinates on an evenly spaced grid of unit dimensions. The maximum stretching factor (i.e., cell size ratio of two consecutive cells) ranges from 1.72 on the coarsest mesh (9x9x9) to 1.04 on the finest mesh (129x129x129) ensuring sufficient variation in the cell quality at the boundaries. The planar boundary surfaces on these grids are simply given as

$$\begin{aligned} &S_{x0}(x,y,z) \equiv x = 0, \quad S_{y0}(x,y,z) \equiv y = 0, \quad S_{z0}(x,y,z) \equiv z = 0 \\ &S_{x1}(x,y,z) \equiv x = 1, \quad S_{y1}(x,y,z) \equiv y = 1, \quad \text{and} \quad S_{z1}(x,y,z) \equiv z = 1. \end{aligned} \tag{19}$$

In this work, results for only the most complex grid type tested are presented. Simplified grid types are discussed only where necessary. A successful verification on a complex grid type (e.g., 3D general hybrid grid with curved boundaries) implies successful verification on the corresponding simpler grid types (e.g., 3D Cartesian grid with planar boundaries).

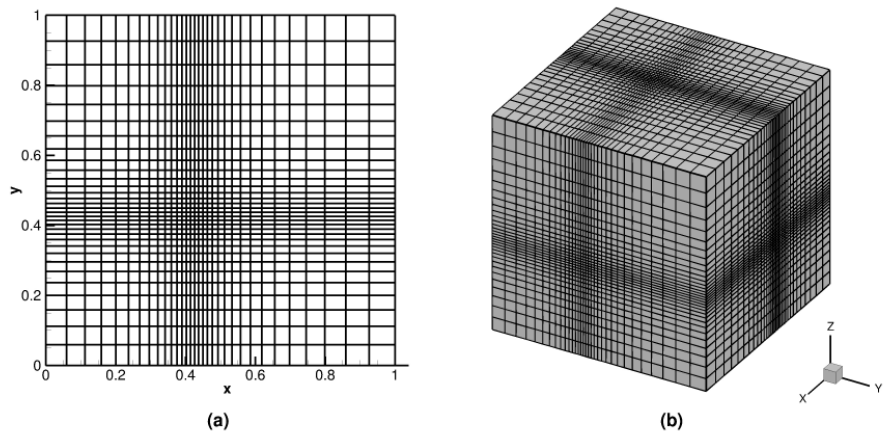

Figure 2: Grids used for BC verification with the incompressible solver: (a) 2D stretched Cartesian mesh, and (b) 3D stretched Cartesian mesh.

## 5. Results and discussion

### *5.1. Baseline governing equations*

Code verification is performed for each boundary condition separately by allowing the boundary condition algorithm implemented in the code to determine the values at the concerned boundary. The solution variables on all other boundaries are specified using the Dirichlet values from the selected manufactured solution. Thus, code verification of the baseline governing equations using Dirichlet specification of all solution variable at all boundaries is a necessary step before proceeding to boundary condition code verification. This ensures that the code has been verified in the interior of the domain and that any errors observed during boundary condition verification result from issues in the boundary condition being tested.

The compressible flow baseline governing equations presented in Sec.2.1 are verified on 3D general hybrid and annular hybrid grids using subsonic and supersonic manufactured solutions. Baseline manufactured solutions are selected and code verification is performed with Dirichlet specification of all the solution variables at all domain boundaries. The implementation of Dirichlet BC in Loci/CHEM sets the outer state from the user-specified conditions, determines the inner state using interior state and characteristic information, and then combines the two states using Roe's flux scheme to formulate the boundary flux. Fig. 3(a) shows a representative plot for the baseline density manufactured solution. The order of accuracy approaches the formal (second) order of accuracy of the code for both $L_2$ and $L_\infty$ norms of the discretization error as shown in Fig. 3(b) and Fig 3(c), respectively. Similarly, for supersonic manufactured solutions all variables are established to be second order accurate for 3D general hybrid and annular hybrid meshes.

The incompressible flow baseline governing equations discussed in Sec. 2.2 are verified on 3D stretched Cartesian grids using manufactured solutions with the divergence-free velocity field of the form described in Eq. 11 and pressure manufactured solution of the form described in Eq. 7. The resulting x-, y-, z-velocity, and pressure manufactured solutions are shown in Fig. 4(a)-(d). The constants selected for the manufactured solutions demonstrate sufficient variation in all solution variables over the domain. The observed order approaches the formal order for both $L_2$ and $L_\infty$ norms of discretization error as shown in Fig. 4(e)(f).

For boundary condition verification, we focus on the strictest of the global error norms, namely the $L_\infty$ norms, as $L_1$ or $L_2$ error norms will be dominated by positive results from the rest of the domain possibly masking any issues at the boundaries within feasible number of mesh levels. Unless mentioned otherwise, all order verification results presented in this work use solutions on five mesh levels with 9×9×9, 17×17×17, 33×33×33, 65×65×65, and 129×129×129 mesh nodes.

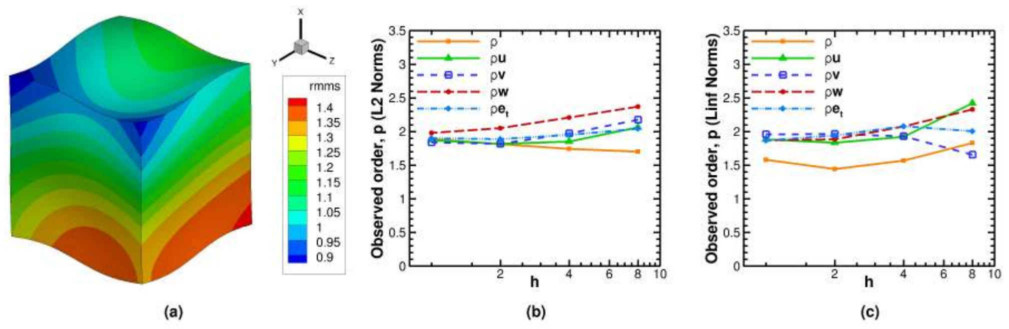

Figure 3: Baseline compressible flow code verification on the general 3D hybrid mesh: (a) density manufactured solution, (b) order of accuracy with $L_2$ norms of the discretization error, and (c) order of accuracy with $L_\infty$ norms of the discretization error.

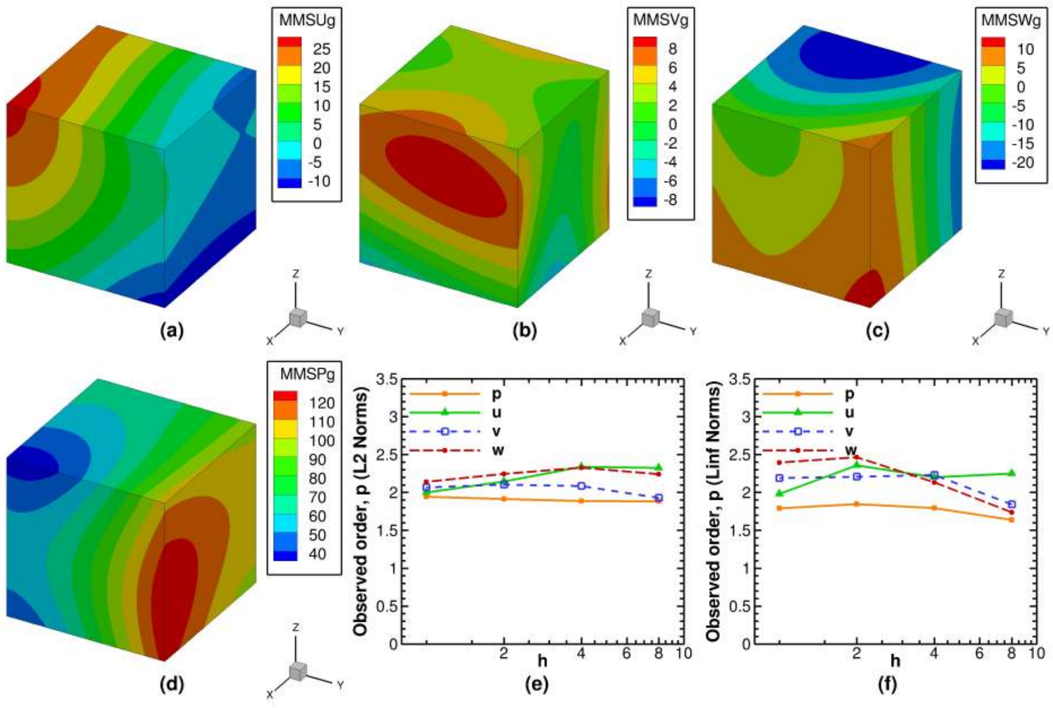

Figure 4: Baseline incompressible flow code verification on 3D stretched Cartesian mesh: (a) x-velocity, (b) y-velocity, (c) z-velocity, and (d) pressure manufactured solutions; order of accuracy using (e) $L_2$ and (f) $L_\infty$ norms of the discretization error.

## *5.2. Compressible flow*

### *5.2.1. Subsonic inflow boundary conditions*

The two subsonic inflow boundary conditions discussed here are isentropic inflow and fixed-mass inflow. These boundary conditions are verified for laminar Navier-Stokes equations using the compressible flow solver. The isentropic inflow BC is a subsonic inflow boundary condition which requires input specification of: (a) total pressure, $p_t$, and total temperature, $T_t$, as constants at the boundary and (b) a direction vector for the flow (which is assumed to be normal if flow directionality is not specified). The mathematical formulation for the isentropic inflow BC [18, 19] assumes that the flow through the boundary is isentropic as well as adiabatic. Based upon the specified $p_t$ and $T_t$ values, and assumption of adiabatic flow (i.e., total enthalpy is conserved), a quadratic equation is formulated to evaluate the speed of sound, and consequently flow velocity, Mach number, pressure, and temperature to determine the right (domain interior) state at the boundary. The left (domain exterior) state is determined using the interior conditions.

One possible approach for obtaining the manufactured solution for the isentropic inflow BC is to select the variables such that

1. $\rho$ and $p$ are constant at the boundary,
2. the velocity magnitude ($\sqrt{u^2+v^2+w^2}$ in 3D) is constant at the boundary,
3. the gradients of $\rho$ and $p$ normal to the boundary are zero,
4. $u$, $v$, and $w$ are such that velocity vector is normal to the boundary,
5. the gradients of $u$, $v$, and $w$ normal to the boundary are zero.

Conditions 1 and 2 ensure that constant $p_t$ and $T_t$ requirements are met, condition 3 is necessary to ensure that the flow through the boundary is adiabatic and isentropic, and condition 4 ensures a normal inflow velocity (assuming normal flow during this test). Condition 5 is required because the implementation of this boundary condition under the physical conditions of isentropic, adiabatic flow assumes zero gradients for velocity variables near the boundary.

Deriving a manufactured solution from these boundary constraints for a general surface is extremely complicated. It can be shown that the simplest possible manufactured solution on a general curved surface satisfies conditions 1-4 while violates condition 5. The use of such a manufactured solution on a general curved boundary can be shown to give erroneous results. For example, consider a 2D grid where the equation for the isentropic inflow boundary curve is given as

$$F_{BC}\left(x,y\right) \equiv y - x\sin\left(\frac{5\pi}{180}\right) - 0.05\sin\left(2\pi x\right) = 0. \tag{20}$$

A simple manufactured solution giving adiabatic, isentropic and normal flow to the curved boundary of Eq. 20 is given as (in SI units)

$$\rho = 1, \quad p = 1\times 10^5, \quad u = \frac{c_0 \partial_x\left(F_{BC}\right)}{\Psi}, \quad \text{and } v = \frac{c_0 \partial_y\left(F_{BC}\right)}{\Psi} \tag{21}$$

where, $c_0 = 80$ m/s is the magnitude of velocity at the boundary, and $\Psi = \sqrt{\left[\partial_x\left(F_{BC}\right)\right]^2 + \left[\partial_y\left(F_{BC}\right)\right]^2}$. The velocity streamlines for the manufactured solution of Eq. 21 are shown in Fig. 5(a) which has non-zero gradients of the velocity variables (i.e., $u$ and $v$) normal to the boundary. Consequently, the numerically converged solution results in an incorrect solution for velocity as shown in Fig. 5(b) and Fig. 5(c) where the velocity is normal yet non-isentropic at the curved boundary. Incorrect boundary solution results in an incorrect solution for the entire flow over the domain.

To address this issue, a set of annular grids such that the concerned boundary is a spherical cap as shown in Fig. 1(d) is used for the isentropic inflow BC verification. This allows us to determine the base terms of manufactured solutions for the $x$-, $y$-, and $z$-velocity components by multiplying the respective direction cosines of the normal vector to the surface with the constant velocity magnitude at the inflow boundary. For example, the $x$-direction cosine for the concerned surface, $S_{r0}$, is multiplied to the velocity magnitude,

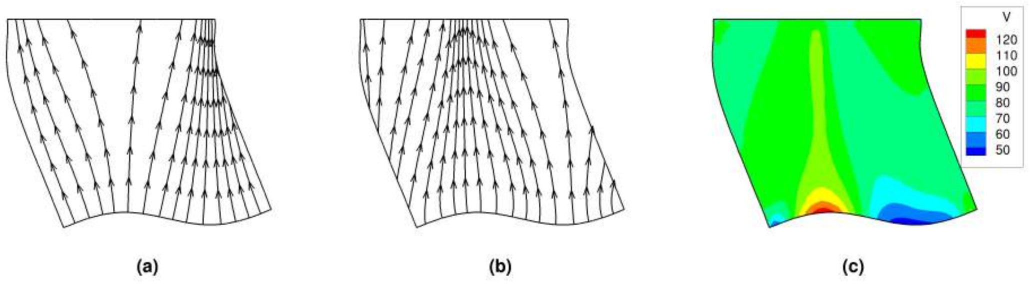

Figure 5: (a) Velocity streamlines for selected manufactured solution, (b) Velocity streamlines for iteratively converged numerical solution, and (c) velocity magnitude for the numerically converged solution, when isentropic inflow BC is used with the 2D manufactured solution of Eq. (22).

$c_0$, to get the base term as $u_0 = c_0 l_x$, where $l_x$ is the $x$-direction cosine of the normal vector to the surface, $S_{r0}$, defined as

$$l_x = \frac{\partial_x (S_{r0})}{\sqrt{[\partial_x (S_{r0})]^2 + [\partial_y (S_{r0})]^2 + [\partial_z (S_{r0})]^2}} \tag{22}$$

The variable term of the manufactured solution is obtained as discussed in Eq. 9, by multiplying the variable part of the baseline manufactured solution (e.g., $u_1$ for $x$-velocity) with the surface function raised to the second power. The combination of base term and the variable term thus obtained meets all 5 conditions required for the manufactured solution to test the isentropic inflow BC. Finally, the manufactured solution for verification of the isentropic inflow BC on 3D annular grids can be written as

$$\begin{aligned} &\rho = \rho_0 + (S_{r0})^2 \rho_1, \quad p = p_0 + (S_{r0})^2 p_1, \quad u = \frac{c_0 \partial_x (S_{r0})}{\Psi} + (S_{r0})^2 u_1, \\ &v = \frac{c_0 \partial_y (S_{r0})}{\Psi} + (S_{r0})^2 v_1, \quad w = \frac{c_0 \partial_z (S_{r0})}{\Psi} + (S_{r0})^2 w_1, \\ &\text{where } \Psi = \sqrt{[\partial_x (S_{r0})]^2 + [\partial_y (S_{r0})]^2 + [\partial_z (S_{r0})]^2} \end{aligned} \tag{23}$$

where velocity magnitude of $c_0 = 80$ m/s ensures subsonic flow. For the manufactured solution of Eq. 23, Figs. 6(a) and 6(b) show that there is small variation in state variables at the inflow boundary, while Fig. 6(c) shows the normal direction of the inflow velocity vector. The observed order of accuracy plot in Fig. 6(e) shows that this boundary condition is verified to be second-order accurate on 3D grids with hexahedral cells at the boundary.

Similar to the isentropic inflow BC, the fixed-mass inflow BC is applicable only to subsonic inflow conditions and requires the specification of (a) mass flux, $\dot{m}/A$, through the boundary, (b) total temperature, $T_t$, at the boundary, and (c) a direction vector for the inflow at the boundary (assumed normal to the boundary here). The mathematical formulation is similar to that for isentropic inflow BC and assumes adiabatic, isentropic flow through the boundary surface. Consequently, the manufactured solution for fixed-mass inflow BC is same as shown in Eq. 23 for isentropic inflow BC and 3D annular grids with hexahedral cells are used for verification. During the simulation, fixed-mass inflow BC is specified using constant values of mass flux, $\dot{m}/A = 80$ kg/m$^2$/s, and total temperature, $T_t = 350.65$ K, calculated using the selected manufactured solutions. The flow is ensured to be subsonic by selecting the velocity magnitude, $c_0 = 80$ m/s. Initial verification of the fixed-mass BC for Loci/CHEM gave negative results. This was found to be because of an incorrect implementation of the boundary condition which was preserving total enthalpy instead of total temperature at the boundary. Although not shown here for brevity, the corrected version of the code was then verified to be second order accurate on 3D annular grids.

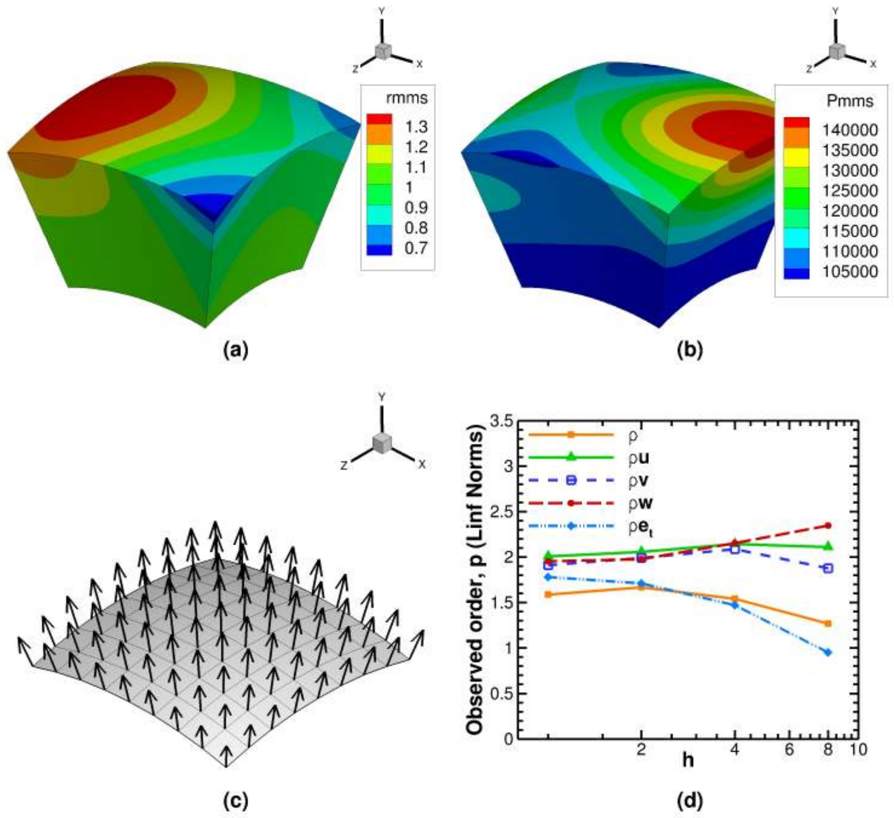

Figure 6: Isentropic inflow manufactured solution on 3D annular grid with hexahedral cell boundary: (a) density, (b) pressure, (c) velocity vectors, and (d) observed order of accuracy using $L_\infty$ norms of discretization error.

Both the subsonic inflow boundary conditions discussed in this section are verified to be second order accurate using laminar Navier-Stokes equations on annular grids with only hexahedral cells at the inflow boundary. Cases with hybrid boundaries failed the order test for these boundary conditions when laminar Navier-Stokes equations were solved while showed iterative non-convergence for Euler equations indicating that there are some unknown issues in using hybrid cells at the boundaries while ensuring isentropic, adiabatic, subsonic inflow. When Euler equations were used along with only hexahedral cells at the inflow boundary, the discretization errors were readily found to decrease at second order with mesh refinement for the same manufactured solutions presented in Eq. 23.

### *5.2.2. Outflow boundary conditions*

The two outflow boundary conditions addressed here are the subsonic outflow and the supersonic extrapolation-based outflow. Both these boundary conditions are tested for laminar Navier-Stokes equations using the compressible flow solver on 3D general grids with hybrid boundaries shown in Fig. 1(a).

The subsonic outflow boundary condition imposes a user-specified constant static pressure at the outflow. The implementation of this boundary condition involves extrapolating the velocity and temperature from the interior which is mathematically imposed using a zero derivative, usually zero gradient or zero curvature, condition on these variables. The velocity extrapolation in Loci/CHEM is done with zero velocity gradient at the boundary which is code specific. A CFD code may assume zero velocity curvature (i.e., second derivative of velocity variables are zero) at the subsonic outflow boundary in which case the manufactured solutions presented here will require appropriate modification. Density is updated using the extrapolated temperature and the specified static pressure under the assumption that flow is adiabatic and isentropic through the outflow boundary. The outflow velocity vector need not be normal to the boundary but must be subsonic and flowing out of the domain.

One possible approach to select the manufactured solutions based upon the subsonic outflow boundary constraints is as follows:

1. the gradients of $u$, $v$, and $w$ normal to the boundary are zero,
2. $\rho$ and $p$ are constant at the boundary,
3. the gradients of $\rho$ and $p$ normal to the boundary are zero,
4. $u$, $v$, and $w$ are selected such that flow is subsonic and outward at the boundary.

Condition 1 is based upon the zero gradient implementation in the code which also ensures isentropic, adiabatic flow at the boundary. Conditions 2 and 3 impose the requirements of isentropic and adiabatic flow through the boundary on state variables. To ensure a subsonic outflow, the base terms of the manufactured solutions for $u$, $v$, and $w$ components of the velocity vector are selected as constants. Such a selection of functions ensures that all solution variables are constant at the boundaries and that their derivatives normal to the boundaries are zero. Finally, the manufactured solution for the verification of the subsonic outflow BC is given as

$$\begin{aligned} &\rho = \rho_0 + (S_{\xi 1})^2 \rho_1, \quad p = p0 + (S_{\xi 1})^2 p_1, \quad u = u0 + (S_{\xi 1})^2 u_1, \\ &v = v0 + (S_{\xi 1})^2 v_1, \quad w = w0 + (S_{\xi 1})^2 w_1. \end{aligned} \tag{24}$$

For boundaries with hexahedral cells using Euler equations, all variables are verified to be second order accurate as shown in Fig. 7(a)(b). For hybrid boundaries using laminar Navier-Stokes equations, it is found that the observed orders of accuracy for mass and energy variables are less than second order and around 1.5 as shown in Fig. 7(d). Examining the errors in density solution, as shown in Fig. 7(c), and pressure solution (not shown here), it is found that large errors are present at the outflow boundary for density and pressure solution, and at the interfaces between the cells of different kinds. The source of the this error is currently unknown. We think that the problem could be related to a first order approximation relating the face center locations and the cell center locations of tetrahedral and prismatic cells.

The supersonic extrapolation-based BC is used to establish zero gradients on velocity, pressure and temperature variables at the boundary for supersonic outflow conditions. There are no other constraints or user specifications required for this boundary condition except that flow should be supersonic and going

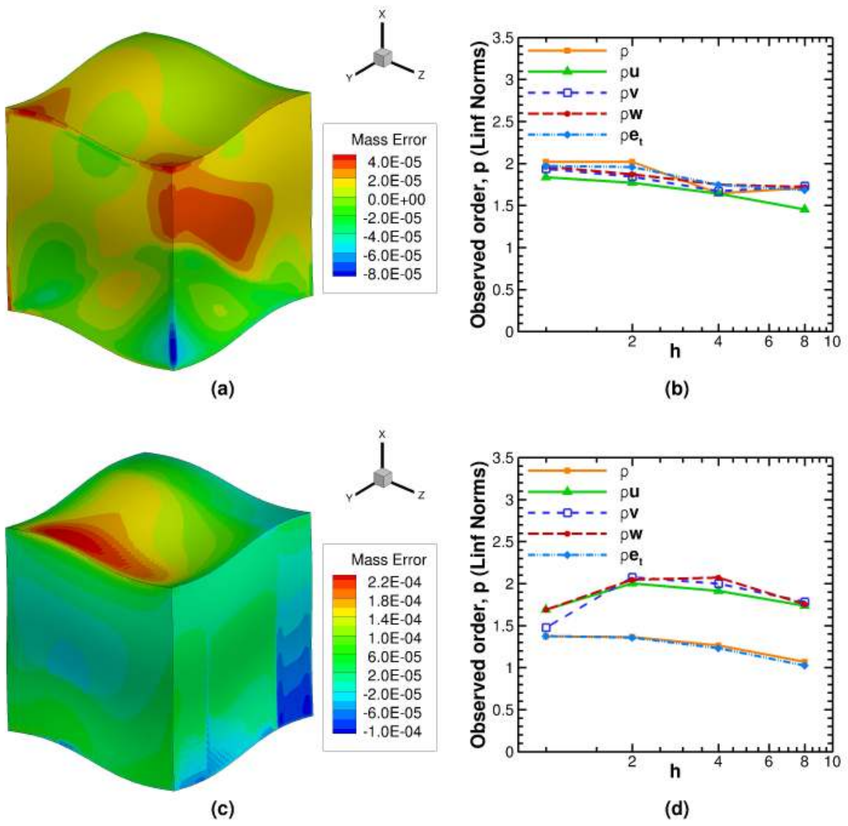

Figure 7: Subsonic outflow boundary condition verification on 3D general grids: (a) mass errors, (b) observed order of accuracy using $L_\infty$ norms of discretization error, for hexahedral cells using Euler equations; (c) mass errors, (d) observed order of accuracy using $L_\infty$ norms of discretization error, for hybrid boundaries using laminar Navier-Stokes equations.

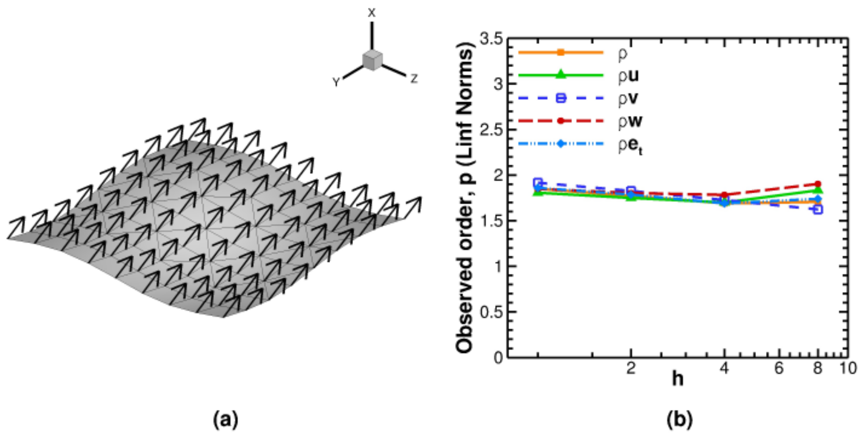

Figure 8: Supersonic extrapolation-based outflow boundary condition verification on 3D general hybrid grid with laminar Navier-Stokes equations: (a) velocity vectors at the outflow boundary, and (b) observed order of accuracy using $L_\infty$ norms of discretization error.

outward. The manufactured solution used for verification is similar to that in Eq. 24 using the constants for supersonic compressible flow given in Table A.2. Fig. 8(a) shows the outflow velocity vectors, and Fig. 8(b) shows the observed order of accuracy using $L_\infty$ norms of discretization error. This boundary condition is verified to be second order accurate on the 3D general grid with hybrid boundaries.

### *5.2.3. Wall boundary conditions*

The three wall boundary conditions addressed in this section are the adiabatic no-slip wall, the isothermal no-slip wall, and the slip-wall. The no-slip wall boundary conditions are used for modeling viscous walls and are tested here using the laminar Navier-Stokes equations on 3D general hybrid grids. The formulation requires no-slip condition on all velocity variables at the boundary. For adiabatic no-slip wall BC, the temperature gradients normal to the wall must be zero while for isothermal no-slip wall BC, the temperature values must approach a constant value at the boundary. The manufactured solutions for velocity variables are obtained by multiplying the boundary surface equation to the baseline velocity manufactured solutions of Eq. 7. For the adiabatic no-slip wall BC, the manufactured solutions for the state variables are selected to ensure adiabatic conditions as shown in Eq. 25. For isothermal no-slip wall BC, the density manufactured solution is formulated in terms of the temperature and pressure manufactured solutions using the perfect gas equation as shown in Eq. 26 since a temperature manufactured solution could not be directly input in Loci/CHEM manufactured solution input routines.

$$\begin{aligned}
&\rho = \rho_0 + \left(S_{\xi 0}\right)^2 \rho_1, \quad p = p_0 + \left(S_{\xi 0}\right)^2 p_1, \\
&u = S_{\xi 0}\left(u_0 + u_1\right), \quad v = S_{\xi 0}\left(v_0 + v_1\right), \quad w = S_{\xi 0}\left(w_0 + w_1\right).
\end{aligned} \tag{25}$$

$$\begin{aligned}
&\rho = \frac{p_0 + S_{\xi 0} p_1}{R\left(T_0 + S_{\xi 0} T_1\right)}, \quad p = p_0 + S_{\xi 0} p_1, \\
&u = S_{\xi 0}\left(u_0 + u_1\right), \quad v = S_{\xi 0}\left(v_0 + v_1\right), \quad w = S_{\xi 0}\left(w_0 + w_1\right).
\end{aligned} \tag{26}$$

Both the no-slip wall boundary conditions are verified to be second order accurate on general 3D grids with hexahedral cells as shown in Fig. 9. For grids with hybrid boundaries, both adiabatic and isothermal no-slip wall BCs fail the order accuracy test for mass and pressure variables as shown in Figs. 10(b) and (d) using the $L_\infty$ norms of discretization error. The errors are found to be located near the interfaces of hexahedral-tetrahedral and tetrahedral-prismatic cells as shown in Figs. 10(a) and (c). The $L_2$ norms of

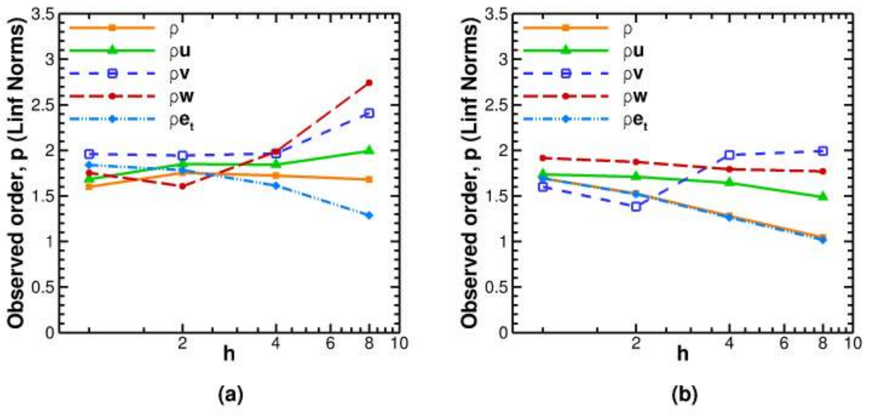

Figure 9: No-slip wall boundary condition verification on 3D grids with hexahedral cells using laminar Navier-Stokes equations: observed order of accuracy using $L_\infty$ norms of discretization error for (a) adiabatic wall conditions, and (b) isothermal wall conditions.

discretization error for these cases show near second order accuracy possibly indicating that the errors in state variables are concentrated only at the mixed cell interfaces of the no-slip wall.

The slip wall BC is implemented using a second order characteristic-based impermeable wall formulation. This boundary condition is tested here with Euler equations on 3D annular grids with hexahedral cells at the boundary. For hybrid grids, slip-wall boundary condition is more robust when implemented with a symmetry wall formulation instead which is not investigated in this work. Here, the slip wall BC is employed with an impermeable wall formulation for inviscid flow over stationary walls where the boundary constraints prescribe that the flow should have both zero mass flux and zero heat flux through the wall. This translates to requirements on the manufactured solutions that flow must be tangent to the boundary and temperature gradients normal to the wall must be zero. The tangency requirement for velocity can be written as:

$$\nabla S_{r0} \cdot \vec{V} = 0 \Rightarrow u l_x + v l_y + w l_z = 0 \tag{27}$$

where, $l_x$, $l_y$, $l_z$, are the direction cosines of the normal vector to the boundary surface as described in Eq. 22. Note that Eq. 27 represents a plane in 3D since there could be infinite vectors tangential to a curved surface at a given point. Thus, we impose an additional constraint that the velocity vector, $\vec{V}$, should be parallel to a fixed vector, $\vec{d} = \hat{i} + \hat{k}$, where $\hat{i}$ and $\hat{k}$ are Cartesian unit vectors in $x$ and $z$ directions, respectively. This selection of $\vec{d}$ as a fixed vector (i.e., not changing with respect to $(x, y, z)$) leads to some loss of generality, but a more general expression should be used carefully and any discontinuities in the manufactured solutions must be avoided. Based upon these two conditions, the relationship between the three components of velocity at the slip-wall is given as:

$$v = -\frac{u\,(l_x + l_z)}{l_y} \quad \text{and } u = w. \tag{28}$$

Finally, a possible manufactured solution for slip-wall verification can be given as

$$\begin{aligned} &\rho = \rho_0 + (S_{r0})^2 \rho_1, \quad p = p_0 + (S_{r0})^2 p_1, \quad u = \frac{y'}{\Psi}(u_0 + u_1), \\ &v = -\frac{(x' + z')}{\Psi}(u_0 + u_1), \quad w = \frac{y'}{\Psi}(u_0 + u_1), \\ &\text{where } \Psi = \sqrt{(x' + z')^2 + 2(y')^2}. \end{aligned} \tag{29}$$

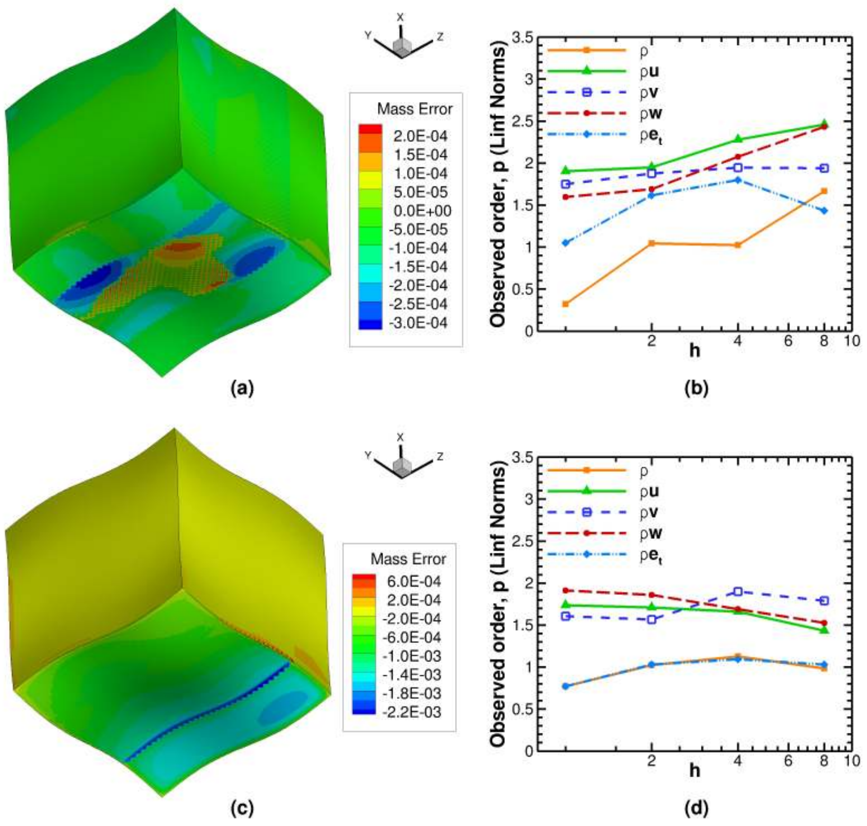

Figure 10: No-slip wall boundary condition verification on 3D general hybrid grid with laminar Navier-Stokes equations: (a) mass errors, and (b) observed order of accuracy using $L_\infty$ norms of discretization error, for adiabatic conditions; (c) mass errors, and (d) observed order of accuracy using $L_\infty$ norms of discretization error, for isothermal conditions.

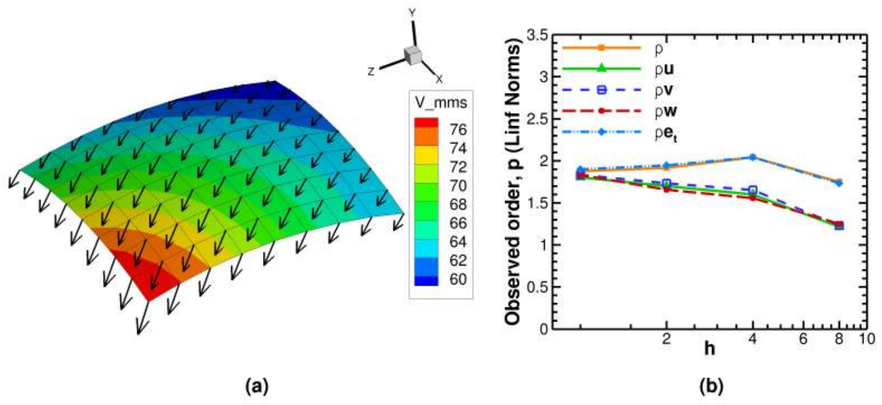

Figure 11: Slip wall boundary condition verification on 3D annular grid with hexahedral cells using Euler equations: (a) velocity vectors at the slip wall boundary, and (b) observed order of accuracy using $L_\infty$ norms of discretization error.

where, $x' = (x - 0.5)$, $y' = (y + 1.8)$, and $z' = (z - 0.5)$ are the scaled direction cosines of the unit normal vector at the concerned spherical cap surface. The baseline function, $(u_0 + u_1)$, in Eq. 29 is the same for all the velocity variables which ensures that the velocity vector satisfies tangential flow at the boundary, $\nabla S_{r0} \cdot \vec{V} = 0$. The scaling parameter, $\Psi$, is selected simply for convenience such that $\sqrt{u^2 + v^2 + w^2} = u_0 + u_1$. The expressions for state variables (i.e., pressure and density) follow the same derivation as in Eq. 23 for isentropic subsonic inflow.

As shown in Fig. 11(a), the direction of the flow over the surface is such that there is some variation in all three coordinate directions at the wall boundary. Second order accuracy is achieved as demonstrated by Fig. 11(b) for slip-wall when hexahedral cells are used at the boundary.

## 5.3. *Incompressible (divergence-free) flow*

The three boundary conditions discussed here for divergence-free flows are the no-slip wall, the slip wall, and the pressure outflow. The selection of manufactured solutions for no-slip wall is discussed in greater detail than for the other two boundary conditions.

### 5.3.1. *Wall boundary conditions*

Apart from the divergence-free flow condition, the implementation in the incompressible code, MFIX, prescribes that for a no-slip wall the velocity must be zero at the wall. For a slip-wall, to ensure tangency at the boundary, the implementation in the code sets the normal component of the velocity vector as zero while the tangential components of velocity are set equal to those in the ghost cell resulting in a zero gradient normal to the wall for these (tangential) components. There are no constraints on pressure at the wall boundaries. The mathematical formulation of these constraints and the boundaries on which these BCs are verified are presented in Table 1.

Consider the no-slip wall boundary condition verification on the 2D stretched Cartesian mesh where the tested boundary is selected as, $S_{x0}\,(x, y) \equiv x = 0$. The manufactured solution satisfying the boundary constraints and divergence-free constraint can be derived as follows. Let $u$ and $v$ be the incompressible velocity manufactured solutions that satisfy the no-slip wall BC at $S_{x0}$ such that

$$u = u_1 x, \quad \text{and } v = v_1 x. \tag{30}$$

where $u_1$, $v_1$ are functions in $x$ and $y$. Then in order to satisfy the divergence-free condition,

$$\frac{\partial u_1 x}{\partial x} + \frac{\partial v_1 x}{\partial y} = 0 \Rightarrow \frac{\partial v_1}{\partial y} = -\frac{\partial u_1}{\partial x} - \frac{u_1}{x}. \tag{31}$$

Table 1: Mathematical constraints on the no-slip wall, slip-wall, and pressure outflow boundary conditions in the incompressible solver

| | No-slip wall | Slip wall | Pressure outflow |
|---|---|---|---|
| Tested boundary | $S_{x0} \equiv x = 0$ | $S_{x0} \equiv x = 0$ | $S_{y1} \equiv y = 1$ |
| Constraints | $\nabla \cdot \vec{V} = 0$ | $\nabla \cdot \vec{V} = 0$ | $\nabla \cdot \vec{V} = 0$ |
| | $\vec{V} = 0$, at $S_{x0}$ | $\vec{V} \cdot \nabla S_{x0} = 0$, at $S_{x0}$ | $\nabla u \cdot \nabla S_{y1} = 0$, at $S_{y1}$ |
| | | $\nabla v \cdot \nabla S_{x0} = 0$, at $S_{x0}$ | $\nabla v \cdot \nabla S_{y1} = 0$, at $S_{y1}$ |
| | | $\nabla w \cdot \nabla S_{x0} = 0$, at $S_{x0}$ | $\nabla w \cdot \nabla S_{y1} = 0$, at $S_{y1}$ |
| | | | $\nabla p \cdot \nabla S_{y1} = 0$, at $S_{y1}$ |

Selecting $u_1 = u_0 x \left(1 + \sin\left(\pi (x+y)^2\right)\right)$ where $u_0$ is a constant, the function $v_1$ can be found as

$$v_1 = \int \left(-\frac{\partial u_1}{\partial x} - \frac{u_1}{x}\right) dy + f_v(x). \tag{32}$$

where $f_v(x)$ is the integration constant selected as $5u_0$ here to ensure a dominantly upward flow. The gauge pressure manufactured solution can be a general function. Finally, the set of manufactured solutions for this case is given as

$$\begin{aligned} u &= u_0 x^2 \left(1 + \sin\left(\pi (x+y)^2\right)\right), \\ v &= u_0 x \left(5 - 3y + \frac{x}{2}\cos\left(2\pi\left(x+y\right)\right) + \frac{1}{2\pi}\sin\left(2\pi\left(x+y\right)\right)\right), \\ p &= p_0 x^2 \left(1 + \sin\left(\pi\left(x+y\right)\right)\right). \end{aligned} \tag{33}$$

For $u_0 = 10$ m/s, and $p_0 = 100$ Pa, the resulting manufactured solutions for x-velocity, y-velocity, and pressure variables are shown in Fig. 12.

It is evident that the procedure just described for verification of incompressible boundary conditions is cumbersome, involves trial-and-error in selection of the general functions and various constants to ensure reasonable flow variation in the domain, and is not easily extendible to 3D grids. For verification on 3D grids, a novel, more general method is presented next that uses the zero curl property of divergence-free velocity fields.

Assume that the velocity vector for the manufactured solution has the form, $\vec{V} = \vec{V}_1 S$, where $\vec{V}_1$ is a general vector function to be determined, and $S \equiv S(x,y,z) = 0$ is a general surface. This manufactured solution for velocity field, $\vec{V}$, satisfies the no-slip wall conditions at $S$. Requiring $\vec{V}$ to be divergence-free and using the mathematical identity that the divergence of a curl is zero, we select

$$\vec{V} = \vec{V}_1 S = \vec{\nabla} \times \vec{G} \Rightarrow \vec{V}_1 = \frac{\vec{\nabla} \times \vec{G}}{S} \tag{34}$$

where $\vec{G}$ is a general 3D vector field. Since the manufactured solution must be well defined over the domain, and $S$ goes to zero at the corresponding boundary, $S$ must be a multiplicative factor in $\vec{\nabla} \times \vec{G}$. Substituting $\vec{G} = S^2 \vec{H}$ in Eq. 34, where $\vec{H}$ is another general vector field, we get

$$\vec{V} = S^2 \vec{\nabla} \times \vec{H} + 2S \nabla S \times \vec{H}. \tag{35}$$

Here, $\vec{H}$ is selected as the general sinusoidal vector field $\{u, v, w\}^T$ described using the functions of Eq. 7 with the constants provided for incompressible flow in Table A.3. The manufactured solution in Eq. 35 meets all the constraints for no-slip wall as presented in Table 1.

Following a similar method, manufactured solution is derived for the slip wall verification, the derivation of which is omitted in this discussion for for brevity. Note from Table 1 that the constraints on the tangential

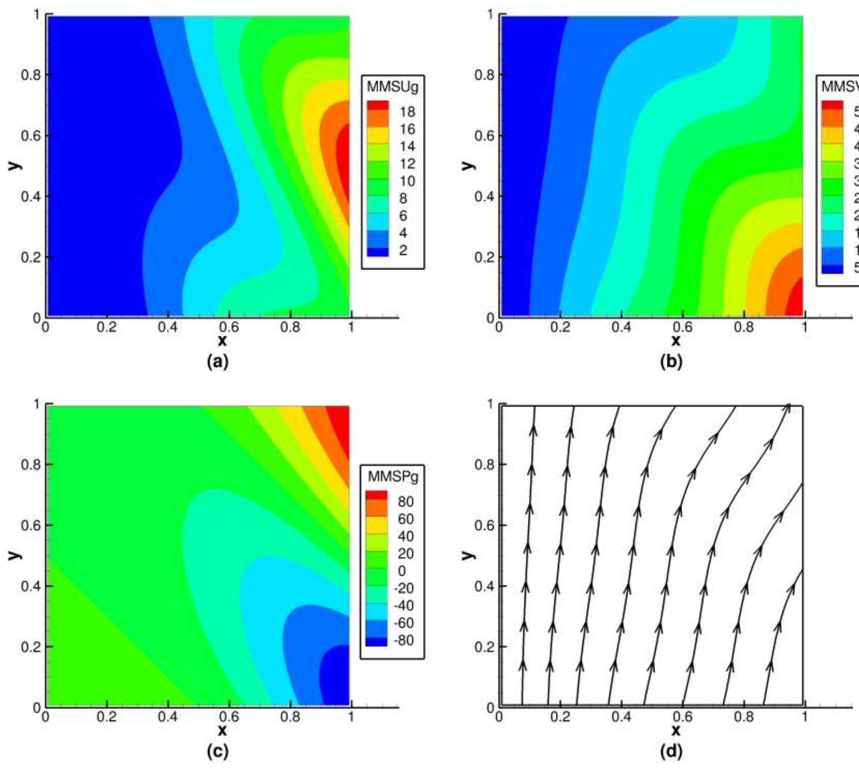

Figure 12: No-slip wall boundary condition manufactured solutions on 2D stretched Cartesian mesh with 0 as the no-slip boundary: (a) x-velocity, (b) y-velocity, (c) pressure, and (d) velocity streamlines.

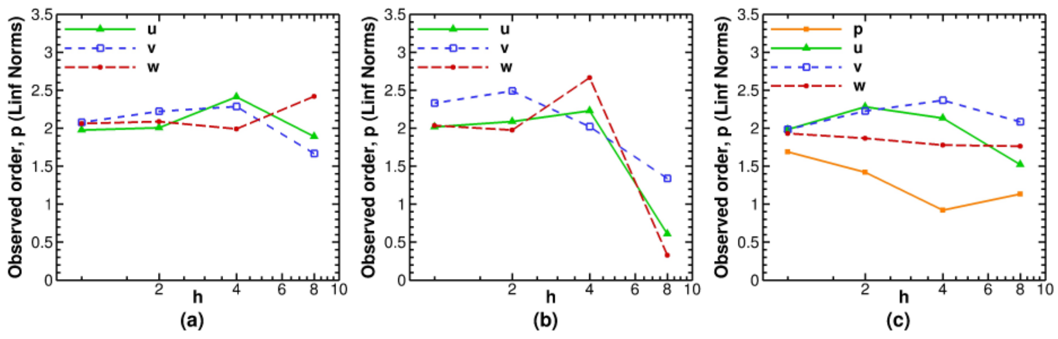

Figure 13: Observed order of accuracy using $L_\infty$ norms of discretization error for (a) no-slip wall, (b) slip-wall, and (c) pressure outflow boundary conditions, for incompressible laminar Navier-Stokes equations on 3D stretched Cartesian grids.

components of velocity at the slip-wall require that the gradients of $v$ and $w$ (i.e., $\nabla v$ and $\nabla w$) in the direction normal to the surface (i.e., $\nabla S_{x0}$) are set as zero. Satisfying these constraints results in a velocity vector, $\vec{V}$, that has a factor of $S_{x0}^2$ in the variable part. Finally, a possible manufactured solutions for slip-wall is given in Eq. 36.

$$\vec{V} = \vec{V_0} + S_{x0}^3 \vec{\nabla} \times \vec{H} + 3S_{x0}^2 \nabla S_{x0} \times \vec{H} \tag{36}$$

where, $\vec{V_0} = \{0, v_0, w_0\}^T$ consists of non-zero scalar constants for $v_0$ and $w_0$.

For wall boundary conditions, convergence issues were encountered for the selected manufactured solutions. This issue is likely due to the use of a source-term driven flow instead of a boundary condition driven flow and the fact that a pressure-projection algorithm is being used. Therefore, currently we have verified the no-slip wall and the slip wall boundary condition for the most complex grid type (3D stretched Cartesian mesh) for momentum equations only by specifying the pressure solution instead of solving for it which is possible in a SIMPLE-based algorithm. The no-slip wall and slip wall boundary conditions are verified to be second order accurate for the momentum equations as shown in Fig. 13(a)(b).

### *5.3.2. Pressure outflow boundary condition*

For the pressure outflow BC, pressure and all three velocity components are required to have zero gradients normal to the outflow boundary. The constraints shown in Table 1 suggest that the velocity manufactured solutions for pressure outflow can be similar to that shown in Eq. 36 for the slip-wall. The pressure manufactured solution follows the basic derivation for functions requiring zero gradient normal to the boundary using Eq. 9 with $m = 2$ and $S_{y1} \equiv y = 1$ as the tested boundary. A possible manufactured solution for verification of the pressure outflow can be written as:

$$\begin{aligned} p &= p_0 + S_{y1}^2 p_1, \\ \vec{V} &= \vec{V_0} + S_{y1}^3 \vec{\nabla} \times \vec{H} + 3S_{y1}^2 \nabla S_{y1} \times \vec{H} \end{aligned} \tag{37}$$

where, $\vec{V_0} = \{u_0, v_0, w_0\}^T$ consists of non-zero scalar constants. The pressure outflow BC is verified to be second order accurate for momentum and pressure solutions as shown in Fig. 13(c) where no convergence issues were encountered since the problem has a proper outflow boundary.

## 6. Conclusions

In this work, we presented a code verification study of different boundary conditions commonly used in compressible and incompressible CFD simulations. We applied the techniques proposed by Bond et al.

[6] to subsonic inflow, subsonic outflow, supersonic outflow, no-slip wall, and slip-wall boundary conditions with laminar Navier-Stokes equations by identifying the boundary constraints and deriving appropriate manufactured solutions for each case. Since a given boundary condition can be implemented with different formulations (such as Neumann or extrapolation-based), many of the manufactured solutions presented vary from previous works and add to the database of boundary condition verification techniques.

Verification techniques for the subsonic inflow boundary conditions were presented for the first time where we proposed the use of simplified curved boundaries (e.g., spherical caps) in order to easily derive the manufactured solutions satisfying the relevant boundary constraints. Furthermore, we proposed the use of general meshes with hybrid boundaries (i.e., boundaries containing cells of mixed types) for rigorous code verification. In the case of no-slip wall boundary conditions, we identified errors at the interface of different cell types at the wall boundary which remained masked with similar analysis on boundaries with all hexahedral cells. Similar issues were encountered when tetrahedral and prismatic cells were used at the boundaries while testing the subsonic inflow and subsonic outflow boundary conditions.

Lastly, we presented two methods for performing boundary condition code verification when a divergence-free velocity field is an added constraint. A new, curl-based method to derive manufactured solutions for divergence-free flows was proposed along with the verification of slip-wall, no-slip wall, and pressure outflow boundary conditions for an incompressible flow solver that employs a pressure-projection algorithm.

The methods we presented here are general and can be applied to verify boundary conditions in other compressible and incompressible CFD codes. For instances where the boundary conditions have been implemented under different assumptions, these methods should be modified accordingly. On the same note, it is our recommendation that code developers interested in boundary condition verification document the exact nature of boundary conditions implemented in their code specifying the physical, mathematical, numerical aspects, as well as boundary flux formulation procedures for finite-volume schemes.

## Acknowledgements

The Loci/CHEM verification study was supported by the National Aeronautics and Space Administration's Constellation University Institutes Program (CUIP) with Claudia Meyer of NASA Glenn Research Center serving as program manager and Kevin Tucker and Jeffrey West of NASA Marshall Space Flight Center serving as technical monitors. The MFIX verification study was supported by the National Energy Technology Laboratory (NETL) through URS Corp. under contract T034:4000.3.671.238.003.413 with Mike Bergen as the URS subcontract technical representative. The authors would like to thank the following members of NETL for multiple communications regarding MFIX - Mehrdad Shahnam, Jordan Musser, Jean-Francois Dietiker, Rahul Garg, Tingwen Li, and Aytekin Gel. The authors also acknowledge Advanced Research Computing at Virginia Tech for providing computational resources and technical support that have contributed to the results reported within this paper. URL: http://www.arc.vt.edu

## References

[1] C. J. Roy, Review of code and solution verification procedures for computational simulation, Journal of Computational Physics 205 (1) (2005) 131 – 156. `doi:http://dx.doi.org/10.1016/j.jcp.2004.10.036`.
URL `http://www.sciencedirect.com/science/article/pii/S0021999104004619`

[2] P. J. Roache, S. Steinberg, Symbolic manipulation and computational fluid dynamics, AIAA Journal 22 (10) (1984) 1390–1394. `doi:10.2514/3.8794`.
URL `http://dx.doi.org/10.2514/3.8794`

[3] P. Knupp, K. Salari, Verification of Computer Codes in Computational Science and Engineering, Chapman & Hall/CRC, 2003.

[4] P. J. Roache, Fundamentals of Verification and Validation, Hermosa Publishers, 2009.

[5] W. L. Oberkampf, C. J. Roy, Verification and validation in scientific computing, Cambridge University Press Cambridge, 2010.

[6] R. B. Bond, C. C. Ober, P. M. Knupp, S. W. Bova, Manufactured solution for computational fluid dynamics boundary condition verification, AIAA Journal 45 (9) (2007) 2224–2236. `doi:10.2514/1.28099`.
URL `http://arc.aiaa.org/doi/abs/10.2514/1.28099`

[7] T. Smith, C. Ober, A. Lorber, Sierra/premo-a new general purpose compressible flow simulation code, in: Fluid Dynamics and Co-located Conferences, American Institute of Aeronautics and Astronautics, 2002. doi:10.2514/6.2002-3292.
URL http://dx.doi.org/10.2514/6.2002-3292

[8] S. P. Veluri, C. J. Roy, E. A. Luke, Comprehensive code verification techniques for finite volume cfd codes, Computers & Fluids 70 (0) (2012) 59–72. http://dx.doi.org/http://dx.doi.org/10.1016/j.compfluid.2012.04.028 doi:http://dx.doi.org/10.1016/j.compfluid.2012.04.028.
URL http://www.sciencedirect.com/science/article/pii/S0045793012001697

[9] E. A. Luke, X.-L. Tong, J. Wu, L. Tang, P. Cinnella, Chem: A chemically reacting flow solver for generalized grids (2003).
URL http://www.tetraresearch.com/locichem/about-locichem/

[10] A. Choudhary, C. J. Roy, E. A. Luke, S. P. Veluri, Issues in verifying boundary conditions for 3d unstructured cfd codes, American Institute of Aeronautics and Astronautics, 2011, doi:10.2514/6.2011-3868. doi:doi:10.2514/6.2011-386810.2514/6.2011-3868.
URL http://dx.doi.org/10.2514/6.2011-3868

[11] D. Folkner, A. Katz, V. Sankaran, Design and verification methodology of boundary conditions for finite volume schemes, Computers & Fluids 96 (2014) 264275. doi:10.1016/j.compfluid.2014.03.028.
URL http://dx.doi.org/10.1016/j.compfluid.2014.03.028

[12] C. Hirsch, Numerical Computation of Internal and External Flows: Computational Methods for Inviscid and Viscous Flows, John Wiley & Sons Inc, 1990.

[13] K. W. Thompson, Time dependent boundary conditions for hyperbolic systems, Journal of Computational Physics 68 (1) (1987) 124. doi:10.1016/0021-9991(87)90041-6.
URL http://dx.doi.org/10.1016/0021-9991(87)90041-6

[14] K. W. Thompson, Time-dependent boundary conditions for hyperbolic systems, ii, Journal of Computational Physics 89 (2) (1990) 439461. doi:10.1016/0021-9991(90)90152-q.
URL http://dx.doi.org/10.1016/0021-9991(90)90152-Q

[15] T. Poinsot, S. Lele, Boundary conditions for direct simulations of compressible viscous flows, Journal of Computational Physics 101 (1) (1992) 104129. doi:10.1016/0021-9991(92)90046-2.
URL http://dx.doi.org/10.1016/0021-9991(92)90046-2

[16] H. C. Yee, Numerical approximation of boundary conditions with applications to inviscid equations of gas dynamics, Report NASA-TM-81265, A-8480, NASA Ames Research Center (1981).
URL http://ntrs.nasa.gov/?R=19810011307

[17] H. C. Yee, R. M. Beam, R. F. Warming, Boundary approximations for implicit schemes for one-dimensional inviscid equations of gasdynamics, AIAA Journal 20 (9) (1982) 1203–1211. doi:10.2514/3.51181.
URL http://arc.aiaa.org/doi/abs/10.2514/3.51181

[18] J.-R. Carlson, Inflow/outflow boundary conditions with application to fun3d, nasa/tm ; 2011-217181, Technical report, Langley Research Center (Oct 2011).

[19] E. A. Luke, X.-L. Tong, J. Wu, P. Cinnella, R. Chamberlain, Chem 3.2: A finite-rate viscous chemistry solver the user guide,, Report, Mississippi State University (2010).

[20] T. H. Pulliam, Notes on solution methods in computational fluid dynamics, Report, VKI Fluid Mechanics Lecture Series (January 20-24 1986).

[21] F. Grasso, C. Meola, Euler and Navier-Stokes equations for compressible flows: Finite-volume methods, Academic Press, London, 1996, book section 4, pp. 235–237. doi:10.1016/B978-012553010-1/50005-0.
URL http://www.sciencedirect.com/science/article/pii/B9780125530101500050

[22] T. Phillips, Residual-based discretization error estimation for computational fluid dynamics, dissertation, Virginia Tech (October 2014).

[23] E. A. Luke, T. George, Loci: a rule-based framework for parallel multi-disciplinary simulation synthesis, J. Funct. Program. 15 (3) (2005) 477–502. doi:10.1017/s0956796805005514.

[24] C. Roy, E. Tendean, S. Veluri, R. Rifki, E. Luke, S. Hebert, Verification of rans turbulence models in loci-chem using the method of manufactured solutions, in: Fluid Dynamics and Co-located Conferences, American Institute of Aeronautics and Astronautics, 2007. doi:10.2514/6.2007-4203.
URL http://dx.doi.org/10.2514/6.2007-4203

[25] S. Hebert, E. Luke, Honey, i shrunk the grids! a new approach to cfd verification, American Institute of Aeronautics and Astronautics, 2005. doi:10.2514/6.2005-685.
URL http://dx.doi.org/10.2514/6.2005-685

[26] S. P. K. Veluri, Code verification and numerical accuracy assessment for finite volume cfd codes, dissertation, Virginia Tech (2010).
URL http://hdl.handle.net/10919/28715

[27] Mfix - multiphase flow with interphase exchanges.
URL https://mfix.netl.doe.gov/

[28] S. Benyahia, M. Syamlal, T. O'Brien, Summary of mfix equations 2012-1 (January 2012).
URL https://mfix.netl.doe.gov/documentation/Theory.pdf

[29] M. Syamlal, W. Rogers, T. O'Brien, Mfix documentation: theory guide, tech. rep. doe/metc-95/1013, ntis/de95000031, Web Page May 28, 2014, National Energy Technology Laboratory (1993).
URL https://mfix.netl.doe.gov/documentation/Theory.pdf

[30] A. Choudhary, C. J. Roy, J.-F. Dietiker, M. Shahnam, R. Garg, Code verification for multiphase flows using the method of manufactured solutions, in: Proceedings of the ASME 2014 4th Joint US-European Fluids Engineering Division Summer

Meeting, 2014.
[31] C. J. Roy, C. Nelson, T. Smith, C. Ober, Verification of euler/navierstokes codes using the method of manufactured solutions, International Journal for Numerical Methods in Fluids 44 (6) (2004) 599–620.
[32] J. L. Thomas, B. Diskin, C. L. Rumsey, Towards verification of unstructured-grid solvers, AIAA Journal 46 (12) (2008) 3070–3079. doi:10.2514/1.36655.
URL http://arc.aiaa.org/doi/abs/10.2514/1.36655
[33] S. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere Series on Computational Methods in Mechanics and Thermal Science, CRC Press, New York, USA, 1980.
[34] M. Syamlal, Mfix documentation: Numerical technique, tech. rep. doe/mc31346-5824, ntis/de98002029, Web Page June 6, 2014, National Energy Technology Laboratory (1998).
URL https://mfix.netl.doe.gov/documentation/numerics.pdf

## Appendix A. Functions and constants in manufactured solutions

Note: The amplitude constants employ SI units while the frequency constants are dimensionless. For incompressible manufactured solutions, pressure ($p$) is the gauge pressure.

Table A.1: Sinusoidal functions and frequency constants used in the compressible and incompressible manufactured solutions

| Variable, $\phi$ | $f_{\phi x}$ | $f_{\phi y}$ | $f_{\phi z}$ | $f_{\phi xy}$ | $f_{\phi yz}$ | $f_{\phi zx}$ |
|---|---|---|---|---|---|---|
| $\rho$ | cos | sin | sin | cos | sin | cos |
| $u$ | sin | cos | cos | cos | sin | cos |
| $v$ | sin | cos | cos | cos | sin | cos |
| $w$ | cos | sin | cos | sin | sin | cos |
| $p$ | cos | cos | sin | cos | sin | cos |
| $T$ | cos | sin | sin | cos | sin | cos |
| Variable, $\phi$ | $a_{\phi x}$ | $a_{\phi y}$ | $a_{\phi z}$ | $a_{\phi xy}$ | $a_{\phi yz}$ | $a_{\phi zx}$ |
| $\rho$ | 0.75 | 0.45 | 0.8 | 0.65 | 0.75 | 0.5 |
| $u$ | 0.5 | 0.85 | 0.4 | 0.6 | 0.8 | 0.9 |
| $v$ | 0.8 | 0.8 | 0.5 | 0.9 | 0.4 | 0.6 |
| $w$ | 0.85 | 0.9 | 0.5 | 0.4 | 0.8 | 0.75 |
| $p$ | 0.4 | 0.45 | 0.85 | 0.75 | 0.7 | 0.8 |
| $T$ | 0.75 | 1.25 | 0.5 | 0.65 | 0.75 | 0.5 |

Table A.2: Amplitude constants used in the subsonic (supersonic) compressible manufactured solutions

| Variable, $\phi$ | $\phi_0$ | $\phi_x$ | $\phi_y$ | $\phi_z$ | $\phi_{xy}$ | $\phi_{yz}$ | $\phi_{zx}$ |
|---|---|---|---|---|---|---|---|
| $\rho$ | 1 | 0.15 | -0.1 | 0.1 | 0.08 | 0.05 | 0.12 |
| $u$ | 70 (800) | 5 (50) | -15 (-150) | -10 (-100) | 7 (70) | 4 (40) | -4 (-40) |
| $v$ | 90 (800) | -5 (-50) | 10 (100) | 5 (50) | -11 (-110) | -5 (-50) | 5 (50) |
| $w$ | 80 (800) | -10 (-100) | 10 (100) | 12 (120) | -12 (-120) | 11 (110) | 5 (50) |
| $p$ ($10^5$) | 1 | 0.2 | 0.5 | 0.2 | -0.25 | -0.1 | 0.1 |
| $T$ | 350 | 10 | -30 | 20 | 10 | -12 | 15 |

Table A.3: Amplitude constants used in the subsonic, incompressible manufactured solutions

| Variable, $\phi$ | $\phi_0$ | $\phi_x$ | $\phi_y$ | $\phi_z$ | $\phi_{xy}$ | $\phi_{yz}$ | $\phi_{zx}$ |
|---|---|---|---|---|---|---|---|
| $u$ | 7 | 3 | -4 | -3 | 2 | 1.5 | 2 |
| $v$ | 9 | -5 | 4 | 5 | -3 | 2.5 | 3.5 |
| $w$ | 8 | -4 | 3.5 | 4.2 | -2.2 | 2.1 | 2.5 |
| $p$ | 100 | 20 | -50 | 20 | -25 | -10 | 10 |