International Journal of Multiphase Flow, 2016

Code Verification for Multiphase Flows Using the Method of Manufactured Solutions[☆,☆☆]

Aniruddha Choudhary[a,b,*], Christopher J. Roy[b], Jean-François Dietiker[c], Mehrdad Shahnam[a], Rahul Garg[a], Jordan Musser[a]

[a] *National Energy Technology Laboratory, Morgantown, WV 26507, USA*
[b] *Aerospace and Ocean Engineering Department, Virginia Tech, Blacksburg, VA 24061, USA*
[c] *West Virginia University Research Corporation, Morgantown, WV 26506, USA*

Abstract

Code verification is the process of ensuring, to the degree possible, that there are no algorithm deficiencies and coding mistakes (bugs) in a scientific computing simulation. Order of accuracy testing using the Method of Manufactured Solutions (MMS) is a rigorous technique that is employed here for code verification of the main components of an open-source, multiphase flow code - MFIX. Code verification is performed here on 2D and 3D, uniform and stretched meshes for incompressible, steady and unsteady, single-phase and two-phase flows using the two-fluid model of MFIX. Currently, the algebraic gas-solid exchange terms are neglected as these can be verified via techniques such as unit-testing. The no-slip wall, free-slip wall, and pressure outflow boundary conditions are verified. Temporal orders of accuracy for first-order and second-order time-marching schemes during unsteady simulations are also assessed. The presence of modified SIMPLE-based algorithm in the code requires divergence-free constraints on the velocity field for single-phase incompressible model and the volume fraction weighted velocity field for the two-phase incompressible model. A newly-developed curl-based manufactured solution is used to generate manufactured solutions that satisfy the divergence-free constraint during the verification of the incompressible governing equations. Manufactured solutions with constraints due to boundary conditions as well as due to divergence-free flow are derived in order to verify the boundary conditions.

Keywords: Multiphase flows, Code verification, Method of manufactured solutions, Order of accuracy, Two-fluid model

[☆] Distribution statement (to be decided)

[☆☆] Preliminary results of this work were presented at: ASME 2014 4th Joint US-European Fluids Engineering Division Summer Meeting, Chicago, Illinois, USA, August 3–7, 2014 (Choudhary et al., 2014)

[*] Corresponding author: aniruddhac@gmail.com, +1-540-239-4824

1. Introduction

With increased use of computational tools for engineering simulations of complex physical systems, it becomes important to perform verification and validation studies for various aspects of a computational simulation. For a Computational Fluid Dynamics (CFD) simulation, verification and validation activities are useful in assessing the correctness of the code, quantifying the numerical accuracy of the simulation, and determining the applicability of the selected mathematical model. 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. Validation deals with the physics of the model and assesses whether the selected mathematical model satisfactorily predicts the physics of interest.

Verification in scientific computing can be categorized into code verification and solution verification. Code verification is the process of examining whether or not there are coding mistakes (bugs) in the computer code and inconsistencies in the algorithm. Solution verification is the process of identifying and estimating different forms of errors present in numerical simulations: discretization error, iterative convergence error, and round-off error. The different criteria for assessing code verification are: expert judgment, error quantification, consistency/convergence, and order of accuracy (Roy, 2005). Out of these, the order of accuracy test is the recommended acceptance test for rigorous code verification (Knupp and Salari, 2003; Roy, 2005; Oberkampf and Roy, 2010). Order of accuracy test requires the evaluation of discretization error on multiple grid levels. Discretization error is defined as the difference between the numerical solution to the discretized equations and the exact solution to the partial differential (or integral) equations. Evaluation of discretization error requires the knowledge of the exact solution for the governing equations which is certainly not known for problems of practical interest. In this scenario, a technique called the Method of Manufactured Solutions (MMS) (Roache and Steinberg, 1984) can be used where a solution is "manufactured" and used as an exact solution. This manufactured solution exactly solves the modified governing equations obtained by adding certain source terms (or forcing functions) to the original governing equations; the source terms are obtained by substituting the manufactured solution into the original governing equations. MMS is based upon the philosophy that code verification deals with the mathematics of the problem and hence arbitrary functions (with certain requirements as discussed later) can be selected as exact solutions. The books by Roache (2009), Knupp and Salari (2003), and Oberkampf and Roy (2010) provide a comprehensive discussion of code verification, MMS, and order of accuracy tests.

CFD simulations of fluid-solids multiphase systems can be categorized into two basic types: (1) continuum approaches, and (2) Lagrangian-Eulerian approaches. In a continuum approach, which is also referred to as the Eulerian-Eulerian method or the Two-Fluid Method (TFM), the different phases are mathematically described as interpenetrating continua and the governing equations for mass, momentum, and energy are obtained by averaging quantities over a control volume. The interaction between different phases is modeled using various sub-models commonly referred to as constitutive relations or closure models. The constitutive relations can be used to formulate interphase exchange terms describing various physical interactions such as momentum transfer and heat transfer between fluids and solid, or solid and solid phases (e.g., see Lun et al. (1984); Gidaspow (1994); Oliveira and Issa (1994) for discussion on modeling of interphase exchange terms). Special constitutive relations (such as interphase drag models, solid-stress models) are needed for practical problems of interest and are developed based upon experiments, or theoretical modeling, or first-principles based numerical simulations (such as direct numerical simulations, (e.g. Tenneti et al., 2011)). Although assessment and improvement of multiphase constitutive models are important processes in multiphase flow dynamics, they are not the main focus of the current work. 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. The code verification of Lagrangian-Eulerian methods is not directly addressed in the current study. However, the methodology presented here is useful for verification of a continuum discrete multiphase model if the carrier-phase equations employ an Eulerian framework.

1.1. Previous work

Code verification of multiphase flows is not as common in the literature as that for single-phase flows. This is due to the presence of approximately twice as many governing equations in multiphase flows compared to single-phase flows, complex interphase interaction terms, and several constitutive relations which make it difficult to obtain manufactured solutions or simple exact solutions for these equations. Grace and Taghipour (2004) discussed the importance of verification and validation activities for CFD models as applied to fluidized beds and other dense multiphase flow systems. In addition, they correctly concluded from a survey of articles claiming “verification” or “validation” for numerical models simulating fluidized beds that these terms have often been used inconsistently with their accepted terminology.

There have been some MMS-based multiphase code verification studies in a multi-material context. “Multiphase” in this sense refers to the presence of materials in the domain with different physical properties thus resulting in solution discontinuities at the material interface. Roache et al. (1990) used MMS to verify a finite-difference ground flow code with discontinuous conductivities in the domain by selecting the manufactured solutions such that they explicitly satisfy the geological boundary conditions. Crockett et al.

(2011) applied MMS to verify a multi-material heat equation solver that uses a Cartesian cut-cell/embedded boundary method to represent the interface between the materials. In works by Roache et al. (1990) and Crockett et al. (2011), the interface locations are considered to be fixed and known a priori. Brady et al. (2012) presented a way to apply MMS to the finite volume multiphase code OSM which is a structured, Cartesian grid code for solving the heat equation. Manufactured solutions were generated using Heaviside and Dirac-delta functions to include the presence of moving interfaces in the domain for a typical immiscible two-phase system. They concluded that with such a discontinuity in material properties, the order of accuracy must reduce to first order for a second or higher order discretization scheme. This conclusion is also supported by Banks et al. (2008) who showed that the formally second order accuracy of the discrete system reduces to first order in the presence of nonlinear discontinuities and to non-integer values below one for linear discontinuities.

Shunn et al. (2012) used MMS to verify an unstructured variable density flow-solver for a miscible two-fluid system with manufactured solutions reflective of the physical behavior common to combustion problems such as convective propagation of density fronts and mixing of species through diffusion. Physically-realistic manufactured solutions for incompressible, single-phase flows were also proposed by Eça et al. (2007, 2012) for code verification of turbulent, wall-bounded flows. Vedovoto et al. (2011) performed a MMS-based code verification study of a pressure-based finite volume numerical scheme suited to variable density, single-phase flows generally encountered in combustion applications. In their work, the authors selected a manufactured solution mimicking the propagation of a corrugated flame front separating heavy from light gases. In all these studies, the manufactured solutions proposed satisfied the necessary criteria such as divergence-free velocity field, wall boundary conditions, or consistency with employed turbulence functions. There are some advantages in using such physically-realistic manufactured in cases such as turbulence model verification (Eça et al., 2012) where the function and roles of different terms change based upon the nature of the solution. However, the selected manufactured solution should not just be realistic but also exhibit enough variations to ensure that all terms in the governing equations are exercised during the verification test (Pelletier and Roache, 2000; Pelletier, 2010).

1.2. Current work

The focus of current work is MMS-based code verification of the numerical scheme implemented in the discretization of two-fluid model governing equations. We use mathematically general manufactured solutions consisting of sinusoidal functions thus ensuring a rigorous verification of all the discretized terms of the governing equations. The algorithm implemented in the code being investigated (i.e., MFIX, version

2014-1 (National Energy Technology Laboratory, 2014)) requires that the velocity field be divergence-free for the selected manufactured solutions. We incorporate this constraint in the current work by introducing a novel, curl-based method to derive manufactured solution for code verification of incompressible flows. We also verify three of the most commonly used boundary conditions, i.e., no-slip wall, free-slip wall, and pressure outflow. While verifying these boundary conditions, we derive the manufactured solutions to satisfy the divergence-free velocity field constraint as well as the boundary constraints. We also examine the temporal order of accuracy for first-order and second-order time-stepping schemes during unsteady simulations.

The current work is restricted to 3D, incompressible, laminar, two-phase flows. The emphasis is put upon systematically developing rigorous manufactured solutions for the verification of two-fluid governing equations and approaching code verification from a mathematical perspective as opposed to using less rigorous or physically realistic manufactured solutions. We make the simplification of omitting the verification of momentum transfer models (i.e., drag models), solid-stress models, and any chemical reactions or phase transformations. These constitutive models may be sufficiently verified through other approaches (e.g., unit testing) since these are are generally algebraic in nature. For example, Garg et al. (2012) verified an interphase interaction model (called the spring-dashpot model) in MFIX-DEM (the Lagrangian-Eulerian solver in MFIX) using a suite of benchmark problems with analytical solutions.

2. Code and governing equations

MFIX (Multiphase Flow with Interphase eXchanges) (National Energy Technology Laboratory, 2014; Syamlal et al., 1993) 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 to simulate flows which commonly occur in energy conversion and chemical reactor processes such as bubbling, circulating, and spouted fluidized beds. MFIX software suite features different modeling methodologies to perform multiphase flow simulations such as Lagrangian-Eulerian methods (DEM and PIC models), quadrature based method (QMOM), and Eulerian-Eulerian method (or two-fluid model). Various drag models, frictional-stress models and kinetic theory models are available as constitutive relations in the multiphase framework of MFIX. For a complete description of multiphase governing equations and various constitutive relations employed in MFIX, see Benyahia et al. (2012). In Eqs. 1-9 below, we present only the governing equations which are relevant to the current discussion. (Einstein summation convention is implied for only i and j indices)

Gas continuity equation

$$\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g\right) + \frac{\partial}{\partial x_i}\left(\varepsilon_g \rho_g U_{gi}\right) = \sum_{n=1}^{N_g} R_{gn} \tag{1}$$

Solids continuity equation

$$\frac{\partial}{\partial t}\left(\varepsilon_m \rho_m\right) + \frac{\partial}{\partial x_i}\left(\varepsilon_m \rho_m U_{mi}\right) = \sum_{n=1}^{N_m} R_{mn} \tag{2}$$

Note that in a SIMPLE-based algorithm, the continuity equations are not directly solved but instead lead to the formulation of a Poisson equation for pressure correction. In the absence of species reaction terms, R_{gn} and R_{mn}, Eqs. 1 and 2 reduce to the divergence-free condition for steady-state, incompressible flows.

Gas momentum equation

$$\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g U_{gi}\right) + \frac{\partial}{\partial x_j}\left(\varepsilon_g \rho_g U_{gj} U_{gi}\right) = -\varepsilon_g \frac{\partial}{\partial x_i}\left(P_g\right) + \frac{\partial}{\partial x_j}\left(\tau_{gij}\right) - \left(\sum_{m=1}^{M} I_{gmi}\right) + f_{gi} + \varepsilon_g \rho_g g_i \tag{3}$$

Solids momentum equation

$$\frac{\partial}{\partial t}\left(\varepsilon_m \rho_m U_{mi}\right) + \frac{\partial}{\partial x_j}\left(\varepsilon_m \rho_m U_{mj} U_{mi}\right) = -\varepsilon_m \frac{\partial}{\partial x_i}\left(P_g\right) + \frac{\partial}{\partial x_j}\left(\tau_{mij}\right) + I_{gmi} - \left(\sum_{k=1}^{M} I_{kmi}\right) + \varepsilon_m \rho_m g_i \tag{4}$$

In Eqs. 3 and 4, the porous media force term (f_{gi}), gas-solid (I_{gmi}) and solid-solid (I_{kmi}) momentum exchange terms are modeled with algebraic models and thus not considered during the verification study. The gravity term ($\varepsilon_g \rho_g g_i$) has also been neglected.

Gas stress model

$$\tau_{gij} = 2\mu_g S_{gij} \quad \text{where, } S_{gij} = \frac{1}{2}\left(\frac{\partial U_{gi}}{\partial x_j} + \frac{\partial U_{gj}}{\partial x_i}\right) - \frac{1}{3}\left(\frac{\partial U_{gi}}{\partial x_i}\right)\delta_{ij} \tag{5}$$

Solids stress model

$$\tau_{mij} = 2\mu_m S_{mij} \quad \text{where, } S_{mij} = \frac{1}{2}\left(\frac{\partial U_{mi}}{\partial x_j} + \frac{\partial U_{mj}}{\partial x_i}\right) - \frac{1}{3}\left(\frac{\partial U_{mi}}{\partial x_i}\right)\delta_{ij} \tag{6}$$

For laminar flows, the gas and solid stress models are simplified to include all the derivative terms while the algebraic closure model terms are neglected as shown in Eqs. 5 and 6.

Gas energy equation

$$\varepsilon_g \rho_g C_{pg} \left[\frac{\partial T_g}{\partial t} + U_{gj} \frac{\partial T_g}{\partial x_j} \right] = -\frac{\partial q_{gi}}{\partial x_i} + \sum_{m=1}^{M} \gamma_{gm} \left(T_m - T_g \right) - \Delta H_g + \gamma_{Rg} \left(T_{Rg}^4 - T_g^4 \right) \tag{7}$$

Solids energy equation

$$\varepsilon_m \rho_m C_{pm} \left[\frac{\partial T_m}{\partial t} + U_{mj} \frac{\partial T_m}{\partial x_j} \right] = -\frac{\partial q_{mi}}{\partial x_i} - \gamma_{gm} \left(T_m - T_g \right) - \Delta H_m + \gamma_{Rm} \left(T_{Rm}^4 - T_m^4 \right) \tag{8}$$

In Eqs. 7 and 8, the gas-solid heat transfer terms (γ_{gm}), the heat of reaction terms (ΔH_g and ΔH_m), and radiative heat transfer terms (γ_{Rg} and γ_{Rm}) have not been considered for verification. The gas and solids heat conduction terms are given as, $q_{gi} = -\kappa_g \partial T_g / \partial x_i$ and $q_{mi} = -\kappa_m \partial T_m / \partial x_i$, respectively.

Solids granular energy equation

$$\frac{3}{2} \varepsilon_m \rho_m \left[\frac{\partial \Theta_m}{\partial t} + \frac{\partial \left(\Theta_m U_{mj} \right)}{\partial x_j} \right] = \frac{\partial}{\partial x_i} \left(\kappa_m \frac{\partial \Theta_m}{\partial x_i} \right) + \tau_{mij} \frac{\partial U_{mi}}{\partial x_j} + \Pi_m - \rho_m \varepsilon_m J_m \tag{9}$$

In Eq. 9, the collisional dissipation (Π_m) and granular energy exchange (J_m) terms are purely algebraic functions and not considered for verification.

The above two-fluid governing equations for the gas and solid phases are implemented within MFIX in a finite-volume formulation where the domain is discretized using structured Cartesian cells. The solution variables are stored in a staggered-grid formulation where the 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. For discretization of spatial derivative terms, various formally second-order centered and upwind schemes are available. For discretization of temporal derivative terms, first order (Euler implicit) and second order (2-step implicit Runge-Kutta) schemes are available. MFIX uses a modified SIMPLE-based algorithm employing a pressure projection method which imposes a divergence-free constraint on: (1) the velocity field when the phase volume fractions are assumed to be constants, and (2) the volume fraction weighted velocity field when the phase volume fractions are assumed to be variable over the domain.

3. Methodology

3.1. Order of accuracy test

The order of accuracy test determines whether the observed order of accuracy for the numerical scheme matches with the formal order of accuracy as the mesh is systematically refined in the asymptotic range. The observed order is determined directly from the simulations by evaluating the rate at which various norms (e.g., L_1, L_2, L_∞) of the solution discretization error over the domain reduce as the grid is refined. The formal order is usually obtained from a truncation error analysis of the discrete equations. The asymptotic range is defined as that range of discretization sizes (e.g., Δx, Δt) where the higher order terms of the truncation error are negligible compared to the lowest order term (i.e., the term which determines the formal order of the scheme). The observed order of accuracy generally fails 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 on multiple systematically refined mesh levels. Systematic mesh refinement (Oberkampf and Roy, 2010) 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. The number of mesh levels required to accurately determine the observed order of accuracy depends on whether the solution is in the asymptotic range and usually varies from one problem to another.

Systematically refined mesh levels for structured grids are usually obtained 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 a uniformly spaced computational space (ξ, η, ζ) 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.

3.2. Mesh types for verification

The model of MFIX under investigation in this study uses a structured, staggered-grid formulation with 2D (or 3D) stretched Cartesian grid cells. Thus, current verification does not address testing with grids containing curved, skewed, or rotated cells. For code verification in the current work, the 2D and 3D stretched Cartesian grids selected are shown in Fig. 1(a) and Fig. 1(b), respectively. The systematically refined grids for verification are generated using an internal layer grid equation of the following form

$$\begin{aligned}
&x = \xi + 2.5\,(0.4 - \xi)\,(1 - \xi)\,\xi, \quad y = \eta + 2.5\,(0.4 - \eta)\,(1 - \eta)\,\eta \quad \text{and} \\
&z = \zeta + 2.5\,(0.4 - \zeta)\,(1 - \zeta)\,\zeta,
\end{aligned} \tag{10}$$

where ξ, η, and ζ are coordinates on a uniformly 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 ($9 \times 9 \times 9$) to 1.04 on the finest mesh ($129 \times 129 \times 129$) ensuring sufficient variation in the cell quality in the domain. The planar boundary surfaces on these grids are simply given as

$$\begin{aligned}
&S_{x0}\,(x, y, z) \equiv x = 0, \quad S_{y0}\,(x, y, z) \equiv y = 0, \quad S_{z0}\,(x, y, z) \equiv z = 0 \\
&S_{x1}\,(x, y, z) \equiv x = 1, \quad S_{y1}\,(x, y, z) \equiv y = 1, \quad \text{and} \quad S_{z1}\,(x, y, z) \equiv z = 1.
\end{aligned} \tag{11}$$

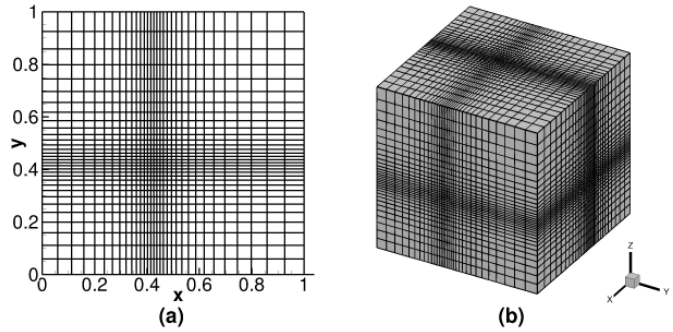

Figure 1: Grids used for code verification: (a) 2D stretched Cartesian mesh, and (b) 3D stretched Cartesian mesh.

3.3. MMS procedure

The MMS was first applied for code verification by Roache and Steinberg (1984) where it was used to verify a code for generation of 3D transformations for elliptic PDEs. A few CFD codes (e.g., Premo (Bond et al., 2007; Roy et al., 2004), WIND (Roy et al., 2004), and Loci/CHEM (Veluri et al., 2012)) have been comprehensively verified using the method of manufactured solutions. Hebert and Luke (2005) 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. (2008) 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 Knupp and Salari (2003); Roache (2009); Oberkampf and Roy (2010) for more examples).

Following the procedure described by Oberkampf and Roy (2010), the MMS is applied here to perform the order of accuracy test as follows:

Step 1: Determine the mathematical form of the governing equations to be verified.

Step 2: Select manufactured solution using analytical functions.

Step 3: Derive the 'MMS' source terms by substituting the manufactured solution into the governing equations of Step 1.

Step 4: Include the MMS source terms and MMS boundary conditions within the numerical algorithm in a minimally intrusive manner (i.e., without modifying the numerical algorithm to be verified).

Step 5: Solve the discrete form of the modified governing equations on multiple, systematically-refined meshes.

Step 6: Evaluate the global discretization error in the numerical solution.

Step 7: Apply the order of accuracy test to determine if the observed order of accuracy matches the formal order.

The selected manufactured solution can involve very general analytical functions and need not be physically realistic. Following the guidelines given by Roy (2005), we select the manufactured solution based upon the following criteria:

1. Manufactured solutions should be smooth, analytical functions of spatial (or temporal, in case of time-stepping scheme 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 manufactured solution cannot be accepted if the code uses square root of temperature to calculate the speed of sound).

3.4. Baseline manufactured solution

The baseline manufactured solution selected for the verification study is a combination of sine and cosine functions and takes the following general form (Roy et al., 2004)

$$\begin{aligned}\phi\left(x,y,z\right) = \phi_0 &+ \phi_x f_{\phi x}\left(\frac{a_{\phi x}\pi x}{L}\right) + \phi_y f_{\phi y}\left(\frac{a_{\phi y}\pi y}{L}\right) + \phi_z f_{\phi z}\left(\frac{a_{\phi z}\pi z}{L}\right) \\ &+ \phi_{xy} f_{\phi xy}\left(\frac{a_{\phi xy}\pi xy}{L^2}\right) + \phi_{yz} f_{\phi yz}\left(\frac{a_{\phi yz}\pi yz}{L^2}\right) + \phi_{zx} f_{\phi zx}\left(\frac{a_{\phi zx}\pi zx}{L^2}\right)\end{aligned} \tag{12}$$

where L represents a characteristic length, $\phi = \left[P_g, u_g, v_g, w_g, u_s, v_s, w_s, T_g, T_s, \Theta_s\right]^T$ represents the set of primitive variables being tested for order of accuracy, 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, ϕ_0, ϕ_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 Knupp and Salari (2003); Roache (2009); Oberkampf and Roy (2010) for more examples of manufactured solutions.) The manufactured solution constants used here are given in Table A.2. The 2D manufactured solutions can be obtained from the 3D manufactured solutions (Eq. 12) by removing the terms involving the third coordinate direction. The characteristic length parameter can be used to modify the periodicity of manufactured solutions and their derivatives within the domain. Herein, we select the characteristic length as equal to the domain length.

3.5. Boundary condition manufactured solution

The general approach for verification of boundary conditions involves obtaining the manufactured solution such that it satisfies all the boundary constraints as implemented within the code. For general velocity fields (i.e., without any divergence-free constraint) this method primarily involves multiplying a baseline standard manufactured solution as described in Eq. 12 by a function representing the boundary surface. This procedure has been previously used in Bond et al. (2007); Choudhary et al. (2011); Veluri et al. (2012); Folkner et al. (2014) for the boundary condition code verification with general flows and is briefly explained in the current section 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. 12 as follows

$$\phi\left(x,\ y\right)=\phi_0+\phi_x f_{\phi x}\left(\frac{a_{\phi x}\pi x}{L}\right)+\phi_y f_{\phi y}\left(\frac{a_{\phi y}\pi y}{L}\right)+\phi_{xy} f_{\phi xy}\left(\frac{a_{\phi xy}\pi xy}{L^2}\right) \tag{13}$$

Suppose the goal is to derive a manufactured solution for ϕ 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 ϕ normal to the boundary up to order $m-1$ are zero.

Following Bond et al. (2007), the manufactured solution satisfying these conditions is obtained by multiplying the sinusoidal parts of the manufactured solution from Eq. 13 with $\left(C-F\left(x,y\right)\right)^m$ as shown in Eq. 14

$$\phi_{BC}\left(x,y\right)=\phi_0+\left(C-F\left(x,y\right)\right)^m\left(\phi_x f_{\phi x}\left(\frac{a_{\phi x}\pi x}{L}\right)+\phi_y f_{\phi y}\left(\frac{a_{\phi y}\pi y}{L}\right)+\phi_{xy} f_{\phi xy}\left(\frac{a_{\phi xy}\pi xy}{L^2}\right)\right) \tag{14}$$

In further discussion and derivations of manufactured solutions, we represent the baseline manufactured solution of Eq. 12 with the following notation

$$\phi=\phi_0+\phi_1 \tag{15}$$

where ϕ_0 is the base term, and ϕ_1 is the variable term consisting of sinusoidal functions. The base term, ϕ_0, can be a constant or a variable function depending on the requirements of the case being verified.

3.6. Divergence-free manufactured solution

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). However, the solution for pressure is addressed differently in compressible flow algorithms than in incompressible flow algorithms. 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 (Patankar, 1980; Syamlal, 1998) 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 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

$$\vec{V} = \nabla \times \vec{H} \tag{16}$$

where, $\vec{H} = \{u_g(x,y,z), v_g(x,y,z), w_g(x,y,z)\}^T$ is a general 3D vector field consisting of functions of the form described in Eq. 12. The manufactured solution functions for other solution variables (pressure, temperature, etc.) are selected directly using Eq. 12.

The expression given by Eq. 16 is used when verification is performed for single-phase governing equations, or two-phase governing equations where phase volume fractions are assumed to be constants. However, when two-phase flows with variable volume fractions are verified then, from Eqs. 1 and 2, the divergence-free constraint is applied to the volume fraction weighted velocity field. The manufactured solution for the velocity field in this case takes the following form:

$$\vec{V}_m = \frac{\nabla \times \vec{H}}{\varepsilon_m} \tag{17}$$

where, ε_m is the volume fraction for the m^{th} phase and is a general function of (x,y,z). Furthermore, for verification of boundary conditions with incompressible flows, the manufactured solutions must be constructed with boundary constraints discussed in Sec. 3.5 as well as the divergence-free constraint discussed here.

4. Results and discussion

In this section, various test cases are described with results. The problems are tested in an increasing order of complexity starting from a 2D rectangular channel single-phase flow case where the analytical solution is known, followed by 2D and 3D, single-phase and two-phase MMS cases. Temporal order verification of time-stepping schemes and verification of boundary conditions is performed for the single-phase equations with divergence-free constraint. Verification of boundary conditions include techniques to derive manufactured solutions for no-slip wall, free-slip wall, and pressure outflow.

The results presented herein are obtained with double precision computations and the normalized residuals are iteratively converged to 10^{-12}. This ensures that the iterative errors are negligible in comparison with the discretization error and thus do not significantly affect the order of accuracy. Additionally, it is also ensured that the code provides similar results when the simulation is performed using different compilers and grid-decompositions for parallel computations since not all simulations are run on the same software and/or hardware configurations. The difference in results for the discretization error norms using different grid decompositions and compilers are found to be of the order of 10^{-10}.

4.1. 2D rectangular channel flow

It is helpful to use the method of exact solutions for problems with analytical solutions available prior to performing a rigorous MMS-based code verification exercise. This establishes some degree of confidence in the code and guides the test engineers towards modifications such as including the MMS source terms and MMS boundary conditions. In the case of plane Poiseuille flow (i.e., pressure driven flow) through a rectangular channel, the 2D, laminar, steady-state, Navier-Stokes equations reduce to a second order linear ODE:

$$\mu_g \frac{d^2 u_g}{dy^2} = \frac{dP_g}{dx} \tag{18}$$

Eq. 18 is analytically solvable when $dP_g/dx =$ constant and no-slip wall boundary conditions are assumed at the channel walls (at $y = 0$ and $y = H$ as shown in Fig. 2(a)). The analytical solution is given as:

$$u_g\left(y\right) = -\frac{dP_g}{dx}\frac{1}{2\mu_g}y(H-y) \tag{19}$$

This problem is solved for single-phase calculations, with cyclic (periodic) boundary conditions along x-direction and second order scheme for numerical discretization. The flow parameters are selected as: channel length, $L = 0.2$ m; channel height, $H = 0.01$ m; fluid density, $\rho_g = 1$ kg/m^3; fluid viscosity, $\mu_g = 0.001$ Pa·s; and pressure drop across the channel length, $\Delta P_g = 240$ Pa. The solution for the x-velocity variable, u_g, results in a second order accurate solution consistent with the simplicity of this problem. Fig. 2(b) shows

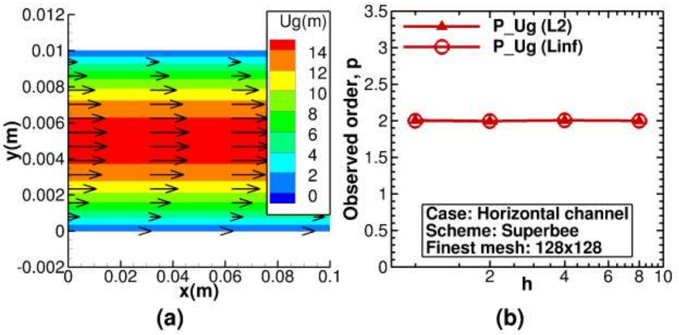

Figure 2: For 2D rectangular channel flow: (a) numerical solution for x-velocity, and (b) observed order of accuracy using L_2 and L_∞ norms of discretization error.

the observed order of accuracy for the x-velocity variable using L_2 and L_∞ norms of the discretization error. Herein, h is the grid size measure which means that for a refinement factor of 2, h for the finest (128×128 cells) to coarsest (8×8 cells) mesh level will be $1, 2, 4, 8,$ and 16, respectively. Pressure having linear variation and y-velocity being zero results in their exact calculation for a second order scheme and thus observed orders for these variables are undefined. This problem can be solved with far less number of cells (theoretically, only one cell) in the x-direction. However, refinement in x-direction tests that the pressure numerical solution is indeed exact (for second order schemes) and is first order accurate (for first order schemes).

4.2. 2D steady-state, single-phase flows

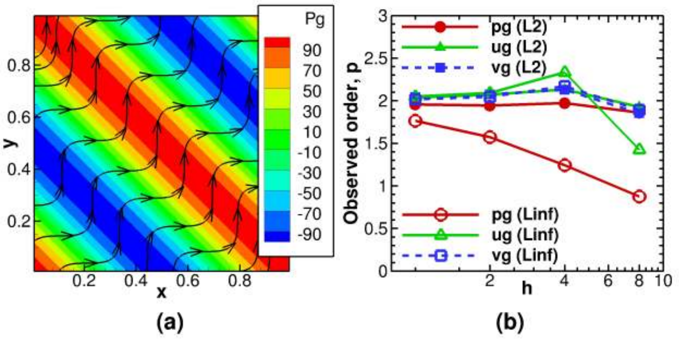

Figure 3: For 2D, steady-state, single-phase flows using simple solenoidal manufactured solution: (a) pressure contours and velocity streamlines, and (b) observed order of accuracy using L_2 and L_∞ norms of discretization error.

For the verification of steady-state, single-phase flows on 2D grids, a simple sinusoidal divergence-free manufactured solution (Vedovoto et al., 2011) is selected as defined in Eq. 20 and shown in Fig. 3.

$$u_g = u_{g0} \sin\left(2\pi\left(x+y\right)\right)^2 \quad v_g = v_{g0} \cos\left(2\pi\left(x+y\right)\right)^2 \quad P_g = P_{g0} \cos\left(2\pi\left(x+y\right)\right) \tag{20}$$

where, $u_{g0} = v_{g0} = 5.0$ m/s, and $P_{g0} = 100$ Pa. For this and other problems verified using MMS, the boundary conditions are prescribed using the manufactured solution. In this problem, u_g and v_g are prescribed from analytical functions at the inflows, while P_g is prescribed using the analytical solution at the outflows. In a SIMPLE-based algorithm using pressure-correction, the numerical solution for pressure matches the exact solution in variation (gradients) and not in actual value. Thus a translational scaling of the pressure solution is performed before comparing the numerical and exact solution for observed order testing.

During initial testing, it was found that the calculation for the cross-terms of the strain-tensor in the momentum equations was not performed within the steady-state sub-iterations which lead to incorrect results. These errors did not show up in cases with zero shear at the boundaries or for unsteady simulations. This issue was subsequently fixed for steady-state simulations. Another important finding was that the Superbee scheme (a second order upwind scheme) resulted in first order accuracy at the west, south, and bottom (for 3D) boundaries when these boundaries were set as inflows. This issue is an artifact of MFIX using the staggered grid with different data structure at west/south/bottom boundaries compared to east/north/top boundaries. One must be careful to make special modifications in the scheme at the boundaries while using upwind approaches. In the rest of the analysis in this work, a central scheme has been used for numerical calculations for which the solution was found to be second order accurate as shown in Fig. 3(b).

Note that, on a staggered grid, the manufactured solution for a variable and the MMS source terms for the corresponding governing equation must be evaluated where the finite-volume approximation of that variable is computed. For example, Fig. 4 shows a representative staggered-grid in 2D where the x-momentum and y-momentum variables are computed at the corresponding face-centers while the scalar variables (such as pressure) are computed at the cell centers. Thus, the manufactured solution and source term for the u_g variable must be evaluated and compared with the numerical solution at the location $(i-1/2, j)$, $(i+1/2, j)$ etc., where (i, j) is the cell-center.

4.3. 3D steady-state, single-phase flows

For code verification of 3D, single-phase, steady-state flows we use the curl-based manufactured solution presented in Eq. 16 for velocity while for pressure, the baseline manufactured solution shown in Eq. 12 is used. The x-velocity, y-velocity, z-velocity and pressure manufactured solutions are shown in Fig. 5. The

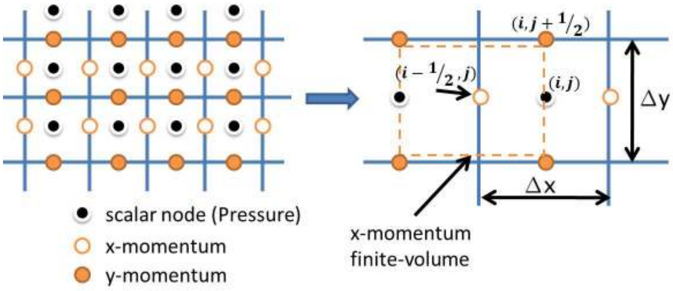

Figure 4: An example staggered-grid showing location of scalar and vector variables

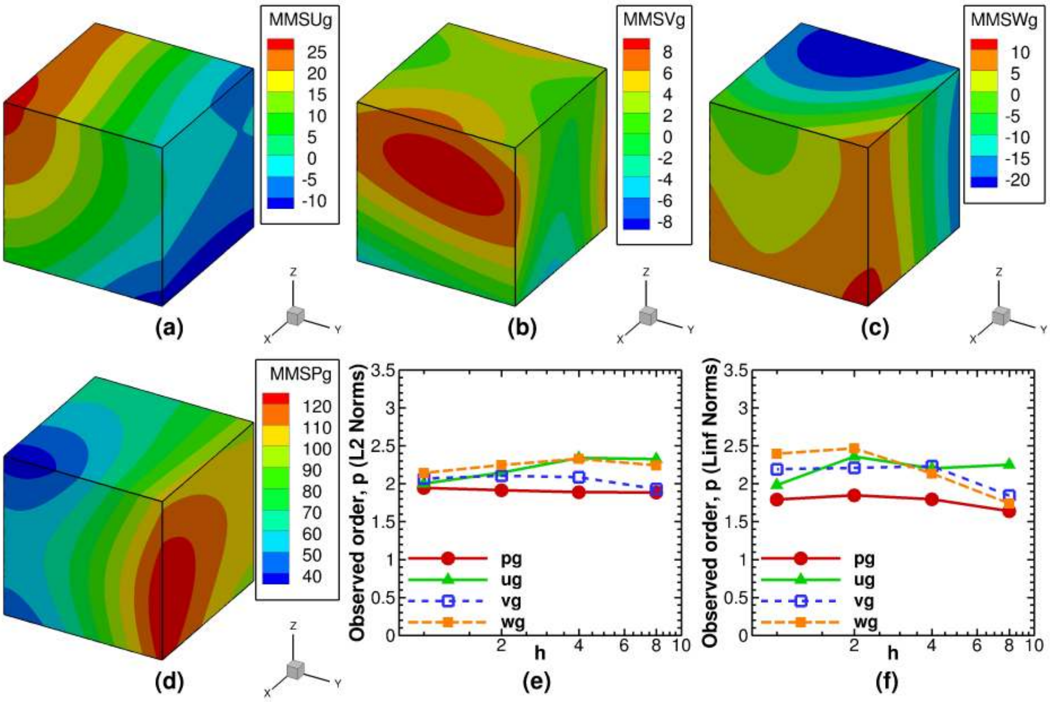

Figure 5: For 3D, steady-state, single-phase flows using curl-based manufactured solution: (a) x-velocity, (b) y-velocity, (c) z-velocity, (d) pressure contours; and observed order of accuracy using (e) L_2 and (f) L_∞ norms of discretization error.

constants selected for the manufactured solutions as given in Table A.2 demonstrate sufficient variation in all concerned variables over the domain. Velocity variables are set at the inflow boundaries and pressure is set at the outflow boundaries using the manufactured solution. A translational scaling is performed for the pressure solution before evaluating the discretization error for order testing. The order of accuracy test gives second order accuracy for L_2 norms as shown in Fig. 5(a)-(d). However, the pressure variable shows an order of 1.7 in the L_∞ norms as shown in Fig. 5(e)-(f). This was confirmed using a finer mesh level ($256 \times 256 \times 256$ cells) and it was found that the observed order of accuracy in pressure calculations does not reduce.

The larger errors in the pressure variable (though still approaching second order accuracy) are found to be related to the quality of the mesh used for verification. For uniform meshes, it was verified that all variables achieve an order of accuracy of two in L_2 and L_∞ norms (not shown here for conciseness). For the grid shown in Fig. 6(a), which is defined using the cubic transformation given in Eq. 10, the larger errors only appear at the boundaries as seen in Fig. 6(b). For this type of grid, the stretching factor assumes a value of one at the boundary due to a common assumption in this and other CFD codes that the ghost cell size equals the first interior cell size. Fig. 6(c) shows another grid where constant stretching factors are used within three segments (i.e., segment 1: $0 \leq x \leq 0.25$, segment 2: $0.25 \leq x \leq 0.75$, and segment 3: $0.75 \leq x \leq 1.0$) along each coordinate direction. The stretching factor is exactly one at every segment intersection (i.e., at $x = 0.25$, $x = 0.75$, etc.). The large pressure errors are clearly seen from Fig. 6(d) to be along these locations of discontinuous variation in stretching factor. This analysis suggests that not having a smoothly varying mesh and having large stretching factor discontinuities at the boundaries due to the common assumption about ghost cell sizes may results in large local errors. Usually, this is not a problem for physical flows where large flow variations at the boundaries are very uncommon.

4.4. 3D steady-state, two-phase flows

Fewer cases with analytical solutions are available for multiphase flows. One such case is the steady fully-developed two-phase flow consisting of solid dispersed phase in a fluid carrier phase between two stationary parallel walls (Drew and Passman, 1999; Kleinstreuer, 2003). An exact solution is available when constant volume fraction and a favorable constant pressure gradient are assumed. However, the governing equations for this problem are difficult to replicate using MFIX governing equations (i.e., Eqs. 1-9) without making significant changes to the code which is counterproductive in a code verification study. Instead, the 3D two-phase flow is directly verified using general manufactured solutions.

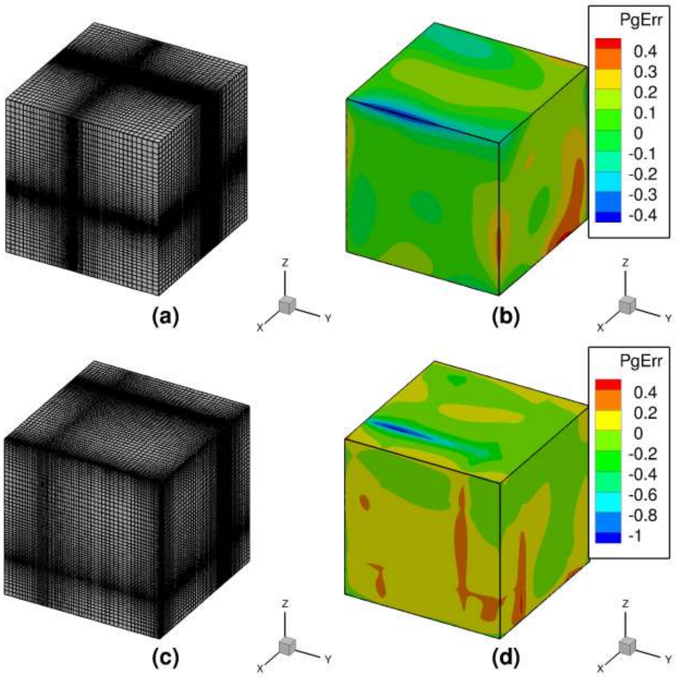

Figure 6: Stretched grid created using smooth transformations: (a) mesh, and (b) errors in pressure. Stretched grid with discontinuous stretching factors: (a) mesh, and (b) errors in pressure. (Note: Grids contain 64 × 64 interior cells)

Extension of the 3D, single-phase, steady-state relations from the previous section to two-phase flows is relatively straightforward; however, there is more work involved due to the presence of many more variables for a constant volume fraction, incompressible flow case (i.e., $\phi = [P_g, u_g, v_g, w_g, u_s, v_s, w_s, T_g, T_s, \Theta_s]^T$. The source terms are generated using the Mathematica symbolic manipulation software (Wolfram Research Inc., 2013) but can contain hundreds of terms. This does not result in a significant computational overhead since the manufactured solutions and the source terms are evaluated only once during each time-step (or only once during the simulation for steady-state flows).

The manufactured solution for both fluid and solid velocity fields are required to be divergence-free and can be defined using either the curl-based approach mentioned in Eq. 16 or a simple solenoidal velocity field similar to that in Eq. 20 with a constant value for w_g in the third direction. Here, we use the curl-based approach for the fluid velocity functions and the simple solenoidal field for the solid velocity functions. The simple solenoidal field was helpful in performing a separate set of inexpensive tests to detect any bias in the code towards a specific coordinate direction. No such coordinate bias was found in the code since when the x-, y-, and z-velocity functions were interchanged, the solution error changed accordingly. The manufactured solution for pressure, fluid temperature, solid temperature, and granular temperature are

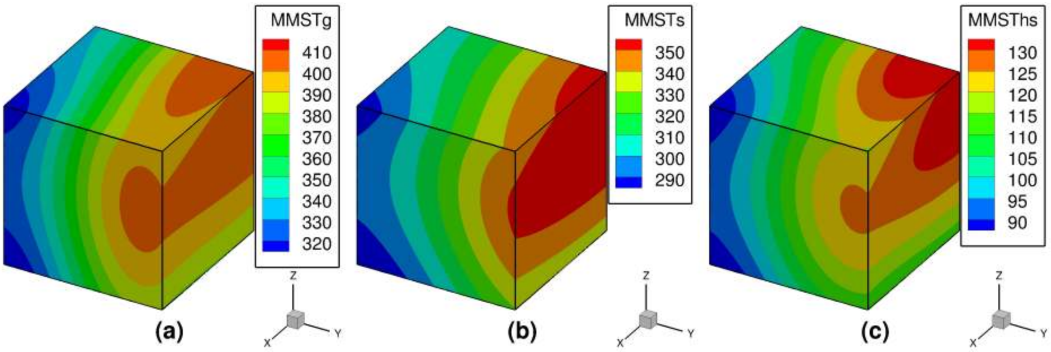

Figure 7: For 3D, steady-state, two-phase flows manufactured solution: (a) gas temperature, (b) solid temperature, (c) solid granular energy.

selected using the general form described in Eq. 12. In addition to the manufactured functions shown in Figs. 5(a)-(d), the temperature related manufactured functions are shown in Figs. 7(a)(b)(c). All governing equations described using Eqs. 1-9 are used during the numerical calculations in this problem with the caveat that constant volume fraction is assumed for this problem (i.e., constant ε_g, and therefore, constant ε_s since $\varepsilon_g + \varepsilon_s = 1$). The inflow and outflow boundaries are set using the appropriate manufactured solution as in previous MMS problems and translational scaling for pressure solution is performed before order testing. The order of accuracy is confirmed to be second order for this case using both L_2 and L_∞ norms as seen from Figs. 8(a) and (b), respectively.

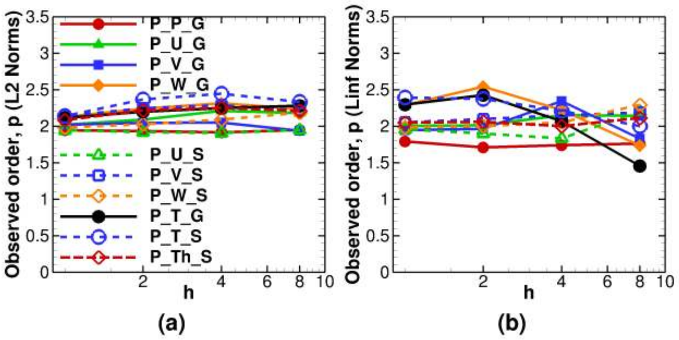

Figure 8: For 3D steady-state two-phase flows, observed order of accuracy using (a) L_2 norms, and (b) L_∞ norms of the discretization error.

For verifying the two-fluid governing equations described in Eqs. 1-9 using variable volume fraction, the general form described in Eq. 12 is used for the solid volume fraction, ε_s, where the sinusoidal functions, frequency constants, and amplitude constants are given in Tables A.2 and A.1. The gas volume fraction is simply specified as $\varepsilon_g = 1 - \varepsilon_s$. The constants in the solid volume fraction manufactured function are selected with sufficient variation in all directions as shown in Fig. 9(a). It is ensured that the gas volume fraction is well above its packed bed (i.e., a state of maximum solids packing) value which is typically set in the current code as $\varepsilon_g^* = 0.4$. Approaching this limit introduces a plastic stress model for solid viscosity which changes the solid momentum equation, Eq. 4, and is not the subject of current investigation. The velocity manufactured solution is then selected to satisfy the divergence free constraint for the volume fraction weighted velocity field as described in Eq. 17.

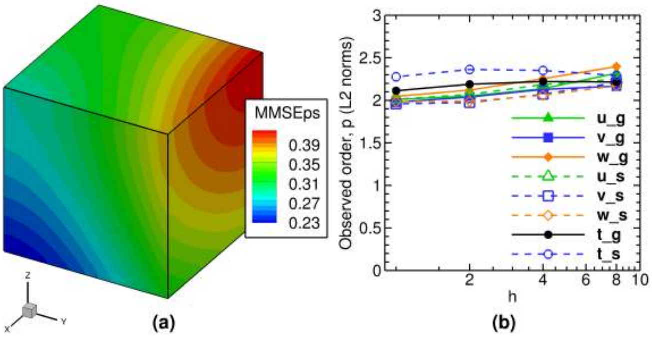

Figure 9: 3D steady-state two-phase flow verification with variable volume fraction. (a) Manufactured solution for solid volume fraction, and (b) Observed order of accuracy for momentum and energy variables using L_2 norms of the discretization error.

For the set of amplitude constants shown in Table A.1 for the solid volume fraction (ε_s) manufactured solution, large oscillations in solution appeared during numerical iterations leading to non-convergence. In the current work, the linear solver selected from the available options in MFIX uses the Biconjugate Gradient Stabilized (BiCGSTAB) method and even sufficient under-relaxation of the linear solver did not aid in convergence. It was our experience during this study that stability in iterative convergence is highly sensitive to large variations (which are of a very non-physical nature) in volume fraction. One of the drawbacks of MMS is that it does not test for robustness in the numerical solution algorithm (Roy et al., 2004) and thus a conclusion about stability of the linear solver cannot be made here. When the amplitude constants in volume fraction manufactured solution are selected to result in very small variation, the numerical solution converges iteratively with second order accuracy but it defeats the purpose of this verification test. For the current test, the continuity variables (ε_s and P_g) are kept fixed at their respective manufactured solution values (and not solved) to obtain iterative convergence. Since MMS allows for non-physical simulations, such a simulation is entirely possible. Thus, the continuity and pressure equations are not solved in this case which restricts us from making any observations about the accuracy of these equations during two-phase flow simulations with variable volume fraction. However, the observed order of accuracy matches the formal order for the momentum and energy equations as shown in Fig. 9(b).

4.5. Boundary condition verification

For verification of boundary conditions, the selected manufactured solutions should satisfy not only the constraints of the boundary condition being verified but also the zero divergence constraint on the velocity field for incompressible flows. During this verification, the condition on the verified boundary itself is prescribed in a manner similar to that in a physically realistic (not MMS) simulation. The conditions on the boundaries not being verified are set using the manufactured solution as done for problems presented in Secs. 4.2-4.4. A detailed discussion of the methodology of verification of boundary conditions under various constraints on stretched Cartesian grids, curved grids, and hybrid grids (i.e., grids with hexahedral, tetrahedral and prismatic cells) using compressible and incompressible CFD codes is presented in Choudhary et al. (in Review) and Choudhary (2015, Ch. 2).

Herein, the three commonly used boundary conditions are verified which have been implemented in the current code with the following constraints:

- No-slip wall: The velocity at the (stationary) no-slip wall is zero. There is no constraint on pressure.
- Slip wall: The normal component of the velocity vector is zero at the (stationary) slip-wall. The tangential component of the velocity vector is set equal to that in the ghost cell. This results in a zero gradient condition normal to the slip-wall for the tangential velocity components. There is no constraint on pressure.
- Pressure outflow: Pressure and all velocity components are set to have zero gradients normal to the outflow at the pressure outflow boundary.

A summary of these mathematical constraints and the boundaries on which these boundary conditions are verified is presented in Table 1. Note that, in the current section and in Sec. 4.6, variables u, v, w, and P are used instead of u_g, v_g, w_g, and P_g, respectively, to avoid unnecessary subscripts since the boundary conditions and temporal schemes are assessed for only single-phase incompressible flows in the current work. For verification of boundary conditions, a novel method is proposed to derive the manufactured solutions using the divergence-free property of the curl of a vector field.

4.5.1. Wall boundary conditions (No-slip wall and Slip wall)

For no-slip wall, we begin by assuming that the velocity field has the form, $\vec{V} = \vec{V}_1 S$, where $\vec{V}_1$ is a general 3D vector field to be determined and $S \equiv S(x, y, z) = 0$ is the verified boundary surface. Thus, $\vec{V}$ satisfies the no-slip wall condition at the surface, S. Next, we require $\vec{V} = \vec{V}_1 S$ to be divergence-free by

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	$\nabla \cdot \vec{V} = 0$ $\vec{V} = 0$, at S_{x0}	$\nabla \cdot \vec{V} = 0$ $\vec{V} \cdot \nabla S_{x0} = 0$, at S_{x0} $\nabla v \cdot \nabla S_{x0} = 0$, at S_{x0} $\nabla w \cdot \nabla S_{x0} = 0$, at S_{x0}	$\nabla \cdot \vec{V} = 0$ $\nabla u \cdot \nabla S_{y1} = 0$, at S_{y1} $\nabla v \cdot \nabla S_{y1} = 0$, at S_{y1} $\nabla w \cdot \nabla S_{y1} = 0$, at S_{y1} $\nabla p \cdot \nabla S_{y1} = 0$, at S_{y1}

equating it to a curl of another general 3D vector field, $\vec{G}$ as shown in Eq. 21.

$$\vec{V} = \vec{V_1} S = \vec{\nabla} \times \vec{G} \Rightarrow \vec{V_1} = \frac{\vec{\nabla} \times \vec{G}}{S} \tag{21}$$

For a manufactured solution to be valid, it must be well-defined over the domain. Therefore, since 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. 21, where $\vec{H}$ is another general vector field, we get

$$\vec{V} = S^2 \vec{\nabla} \times \vec{H} + 2S \nabla S \times \vec{H}. \tag{22}$$

Now, $\vec{H}$ is selected as the general sinusoidal vector field $\{u, v, w\}^T$ described using the functions of Eq. 12 with the constants provided in Table A.2. The divergence-free velocity manufactured solution given by Eq. 22 meets all the constraints for no-slip wall presented in Table 1.

Similarly, manufactured solution is derived for the slip wall boundary (detailed derivation is omitted here for conciseness). From Table 1, note that the constraints on the tangential components of velocity at the slip-wall require that the gradients of v and w (i.e., ∇v and ∇w) in the direction normal to the surface (i.e., ∇S_{x0} in this case) 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 by Eq. 23.

$$\vec{V} = \vec{V_0} + S_{x0}^3 \vec{\nabla} \times \vec{H} + 3S_{x0}^2 \nabla S_{x0} \times \vec{H} \tag{23}$$

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. Therefore, currently the no-slip wall and the slip wall boundary conditions have been verified for the most complex grid type (i.e., 3D stretched Cartesian mesh) for only the momentum equations by specifying the pressure over the domain directly using the manufactured solution (instead of iteratively solving for pressure). The no-slip wall and slip wall boundary conditions are verified to be second order accurate for the momentum equations as shown in Figs. 10(a)(b). Although a partial slip boundary condition is commonly used in multiphase flows, the two-phase implementation of boundary conditions is not investigated here due to the complexity of deriving a manufactured solution for such a boundary condition.

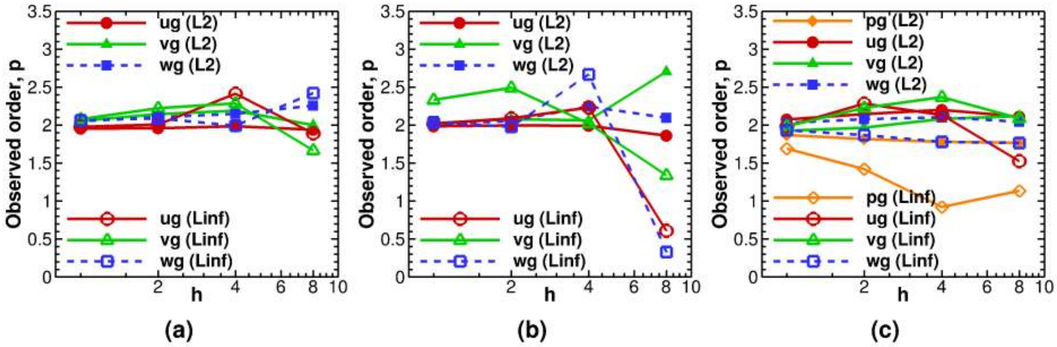

Figure 10: Observed order of accuracy using L_2 and L_∞ norms of discretization error for (a) no-slip wall, (b) slip-wall, and (c) pressure outflow boundary conditions, for incompressible laminar Navier-Stokes equations on 3D stretched Cartesian grids.

4.5.2. Pressure outflow boundary condition

For pressure outflow, the constraints shown in Table 1 suggest that the velocity manufactured solution in this case can be similar to that shown in Eq. 23 for the slip-wall. The pressure manufactured solution follows the basic derivation for functions requiring zero gradient normal to the boundary using Eq. 14 with $m = 2$ and $S_{y1} \equiv y = 1$ as the tested boundary. Thus, a possible manufactured solution for verification of the pressure outflow can be written as:

$$\begin{aligned} P &= P_0 + S_{y1}^2 P_1, \\ \vec{V} &= \vec{V}_0 + S_{y1}^3 \vec{\nabla} \times \vec{H} + 3 S_{y1}^2 \nabla S_{y1} \times \vec{H} \end{aligned} \tag{24}$$

where, $\vec{V}_0 = \{u_0, v_0, w_0\}^T$ consists of non-zero scalar constants. The pressure outflow is verified to be second order accurate for momentum and pressure equations as shown in Fig. 10(c). Here, no convergence issues were encountered since the problem has a physically-realistic outflow boundary.

4.6. Temporal order verification

Code verification for unsteady flows requires a combined order of accuracy test for both spatial and temporal terms as both the grid and the time-step are refined simultaneously. There are different methods in literature to perform temporal order verification (see Kamm et al. (2003)). The method used in the current discussion follows the description in Oberkampf and Roy (2010); Veluri et al. (2012).

Neglecting the higher order terms, the discretization error for a scheme with spatial and temporal terms, can be written as

$$\varepsilon_{h_x}^{h_t} = g_x h_x^{\hat{p}} + g_t h_t^{\hat{q}} \tag{25}$$

where $\hat{p}$ and $\hat{q}$ are the observed orders of accuracy in space and time, respectively, g_x and g_t are the coefficients of spatial and temporal terms, respectively, and h_x and h_t are normalized spatial and temporal discretizations, respectively. The following steps are employed to determine the temporal order of accuracy of the code.

1. A constant time-step is selected making the temporal discretization error term in Eq. 25 (i.e., $g_t h_t^{\hat{q}}$) a constant with respect to grid refinement. Discretization error is found on three systematically refined mesh levels to determine g_x and $\hat{p}$ using Eq. 26 and Eq. 27 (Oberkampf and Roy, 2010).

$$\hat{p} = \frac{ln\left(\frac{\left\|\varepsilon_{r_x^2 h_x}^{h_t}\right\| - \left\|\varepsilon_{r_x h_x}^{h_t}\right\|}{\left\|\varepsilon_{r_x h_x}^{h_t}\right\| - \left\|\varepsilon_{h_x}^{h_t}\right\|}\right)}{ln\left(r_x\right)} \tag{26}$$

$$g_x = \frac{\left\|\varepsilon_{r_x h_x}^{h_t}\right\| - \left\|\varepsilon_{h_x}^{h_t}\right\|}{h_x^{\hat{p}}\left(r_x^{\hat{p}} - 1\right)} \tag{27}$$

   where, r_x is the refinement factor for spatial refinement.

2. Similarly, a constant grid-size is selected and discretization error is found for three different time-step values to determine g_t and $\hat{q}$ in the temporal part of the discretization error expression with a temporal refinement factor of r_t.

3. To ensure that the temporal and spatial errors are of similar orders, h_x and h_t are selected such that $g_x h_x^{\hat{p}} \approx g_t h_t^{\hat{q}}$. This selection requires trial and error, taking cost of computation and numerical stability

into consideration.

4. For the selection of spatial and temporal discretization sizes and known formal orders, multiple cases are run to determine the temporal observed order of accuracy for the combined spatial-temporal refinements.

Unsteady equations are solved for multiple grid-levels and time steps using time dependent manufactured solutions as given in Eq. 28 for 3D, single-phase flows to obtain observed spatial and temporal accuracy orders.

$$
\begin{aligned}
u &= u_0 \sin\left(2\pi\left(x+y+z+t\right)\right)^2 \\
v &= u_0 \cos\left(2\pi\left(x+y+z+t\right)\right)^2 \\
w &= w_0 \\
P &= P_0 \cos\left(2\pi\left(x+y+z+t\right)\right)
\end{aligned}
\tag{28}
$$

These relatively simple unsteady manufactured solutions are used instead of curl-based functions since unsteady calculations can be very expensive, especially for levels with small cell sizes and small time-steps. The variable time-stepping algorithm available in MFIX is turned off to ensure that the selected time step is being used for these calculations. All boundaries are prescribed using the manufactured solutions appropriately.

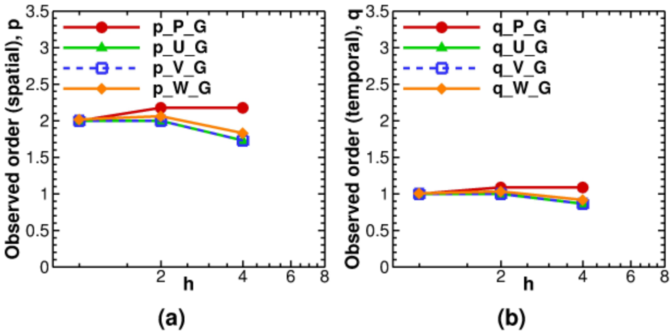

Figure 11: (a) Spatial scheme, and (b) temporal scheme observed orders of accuracy using L_2 norms of discretization error when Euler implicit (formally first order) time-stepping and central (formally second order) discretization of spatial terms are employed in a combined mesh and time-step refinement study.

The first order (implicit Euler) time-stepping method in MFIX is verified to be first order accurate in time and second order accurate in space as shown in Fig. 11(a) and Fig. 11(b), respectively. However, the formally second order time-stepping method (i.e., implicit 2-step Runge-Kutta method or TSRK) failed the order of accuracy test. The error was traced to a simplification made in the code for iterative robustness. In theory, the scheme can be described as follows:

$$\begin{aligned} &\text{Step-1 (implicit Euler): } \phi^{n+1/2} = \phi^n + (\Delta t/2)\phi^{n+1/2} \\ &\text{Step-2 (explicit extrapolation): } \phi^{n+1} = 2\phi^{n+1/2} - \phi^n \end{aligned} \tag{29}$$

where, $\phi^{n+1/2}$ is the solution at the intermediate time step between n and $n+1$. The simplification in the code did not perform Step-2 from Eq. 29 for the pressure (P_g) variable leading to an inconsistent time-stepping scheme.

5. Conclusions

A code verification study employing the method of manufactured solutions was presented for the governing equations of the two-fluid model used in multiphase flows. Different features of the code were verified such as the discretized terms of momentum, pressure-correction, and energy equations, for 2D and 3D, steady/unsteady, single-phase and two-phase incompressible flows. A newly-developed curl-based method to derive manufactured solutions was introduced for divergence free flows during the verification of baseline governing equations as well as during boundary condition verification. No-slip and free-slip wall boundary conditions were verified to be second order accurate for momentum equations while the pressure outflow boundary condition was found to be second order accurate for momentum as well as pressure-correction equations on 3D grids using single-phase flow equations.

Simplifications were made by assuming incompressible flow, constant viscosity, and by neglecting the algebraic constitutive relations that describe the essential interactions between phases in a multiphase flow simulation. Even with these simplifications, many useful lessons were learned from this code verification study about the implementation of discrete governing equations in this code. The multiphase code, MFIX, was found to be second order accurate for the core part of the algorithm (i.e., the two-fluid model governing equations). Some issues were uncovered in the implementation of Superbee scheme at the boundaries, in the evaluation of cross-terms of the strain-tensor during steady-state simulations, and in the second order time-stepping method. It was observed that the common practice of setting the ghost cell size equal to the first interior cell size leads to some reduction in solution accuracy without completely failing the order test.

Code verification of multiphase flows is challenging, especially in cases where the interface between the phases is resolved using discrete elements (such as in Lagrangian-Eulerian methods). Though current study addresses only continuum governing equations for code verification and is similar to single-phase flow code verification in many ways, the emphasis is put upon using rigorous, mathematical methods such as order of accuracy testing for code verification. The verification of algebraic constitutive models based upon unit-testing in conjunction with the MMS-based verification of discretized terms of the governing equation presented here, along with other practices such as solution error estimation and model validation are ultimately necessary to build greater confidence in multiphase flow simulations.

Nomenclature

Roman symbols	
C_{pg}	Specific heat of the fluid phase
C_{pm}	Specific heat of the m^{th} solids phase
f_{gi}	Fluid flow resistance due to porous media
$f_{\phi x}, f_{\phi y} \dots$	Sinusoidal functions in the manufactured solutions
$\vec{G}$, $\vec{H}$	General 3D vector fields
g_i	Acceleration due to gravity
g_t	Coefficient of temporal terms in the spatial-temporal discretization error expression
g_x	Coefficient of spatial terms in the spatial-temporal discretization error expression
ΔH_g	Heat of reaction in the fluid phase
ΔH_m	Heat of reaction in the m^{th} solids phase
h_t	Normalized temporal discretization
h_x or h	Normalized spatial discretization (measure of grid size)
I_{gmi}	Gas-solid interaction force
I_{kmi}	Solid-solid interaction force
J_m	Collisional dissipation term for solid phase m
L	Characteristic length in manufactured solutions
M	Total number of solids phases
N_g	Total number of chemical species for gas phase
N_m	Total number of chemical species for solid phase m
P_g	Pressure in the fluid phase
$\hat{p}$	Spatial observed order of accuracy
$\hat{q}$	Temporal observed order of accuracy
q_{gi}	Fluid-phase conductive heat flux
q_{mi}	Solids-phase m conductive heat flux
R_{gn}	Rate of production of the n^{th} chemical species in the fluid phase
R_{mn}	Rate of production of the n^{th} chemical species in the m^{th} solids phase
T_g	Thermodynamic temperature of the fluid phase
T_m	Thermodynamic temperature of the solids phase m
T_{Rg}	Fluid phase radiation temperature

T_{Rm}	Solids phase m radiation temperature
U_g	Fluid phase velocity
U_m	Solids phase m velocity

Greek symbols	
δ_{ij}	Kronecker delta function
ε_g	Volume fraction of the fluid phase (void fraction)
ε_g^*	Void fraction at maximum packing of solids
ε_m	Volume fraction of the m^{th} solids phase
γ_{gm}	Fluid-solids heat transfer coefficient corrected for interphase mass transfer
γ_{Rg}	Fluid-phase radiative heat transfer coefficient
Θ_m	Granular temperature of phase m
Θ_s	Solid phase granular temperature
κ_g	Fluid phase conductivity
κ_m	Solids phase m conductivity
μ_g	Molecular viscosity of the fluid phase
μ_m	Molecular viscosity of the solid phase m
Π_m	Energy exchange term for the solid phase m
ρ_g	Microscopic (material) density of the fluid phase
ρ_m	Microscopic (material) density of the m^{th} solids phase
τ_{gij}	Fluid-phase stress tensor
τ_{mij}	Solids phase m stress tensor
$\phi_0, \phi_x \dots$	Amplitude constants in the manufactured solutions
$a_{\phi x}, a_{\phi y} \dots$	Frequency constants in the manufactured solutions

Subscripts	
g	Subscript indicating gas (fluid) phase
m	Index of the m^{th} solids phase
n	Index of the n^{th} chemical species
s	Subscript indicating solid phase

Acknowledgements

The authors would like to thank Dr. Tingwen Li of NETL, Morgantown, WV, and Dr. Aytekin Gel of ALPEMI Consulting, LLC., Phoenix, AZ for numerous discussions over the period of this study. The MFIX verification study was supported by the National Energy Technology Laboratory (NETL) through URS Corp. (Grant No. 4000.3.671.052.002.411). This research was supported in part by an appointment to the National Energy Technology Laboratory Research Participation Program, sponsored by the U.S. Department of Energy and administered by the Oak Ridge Institute for Science and Education. 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

Banks, J., Aslam, T., Rider, W., 2008. On sub-linear convergence for linearly degenerate waves in capturing schemes. Journal of Computational Physics 227, 6985–7002. doi:10.1016/j.jcp.2008.04.002.

Benyahia, S., Syamlal, M., O'Brien, T., 2012. Summary of MFIX Equations 2012-1. Technical Report. National Energy Technology Laboratory. URL: https://mfix.netl.doe.gov/download/mfix/mfix_current_documentation/MFIXEquations2012-1.pdf. [Accessed: 2015-05-06].

Bond, R.B., Ober, C.C., Knupp, P.M., Bova, S.W., 2007. Manufactured solution for computational fluid dynamics boundary condition verification. AIAA Journal 45, 2224–2236. doi:10.2514/1.28099.

Brady, P., Herrmann, M., Lopez, J., 2012. Code verification for finite volume multiphase scalar equations using the method of manufactured solutions. Journal of Computational Physics 231, 2924–2944. doi:10.1016/j.jcp.2011.12.040.

Choudhary, A., 2015. Verification of Compressible and Incompressible Computational Fluid Dynamics Codes and Residual-based Mesh Adaptation. Ph.D. thesis. Virginia Tech. URL: http://hdl.handle.net/10919/51169.

Choudhary, A., Roy, C.J., Dietiker, J.F., Shahnam, M., Garg, R., 2014. Code verification for multiphase flows using the method of manufactured solutions, in: ASME 2014 4th Joint US-European Fluids Engineering Division Summer Meeting collocated with the ASME 2014 12th International Conference on Nanochannels, Microchannels, and Minichannels, ASME. p. V01CT23A010. doi:10.1115/fedsm2014-21608.

Choudhary, A., Roy, C.J., Luke, E.A., Veluri, S.P., 2011. Issues in verifying boundary conditions for 3D unstructured CFD codes, in: 20th AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics. doi:10.2514/6.2011-3868.

Choudhary, A., Roy, C.J., Luke, E.A., Veluri, S.P., in Review. Code verification of boundary conditions for compressible and incompressible computational fluid dynamics codes. Computers & Fluids Manuscript Ref. No.: CAF-D-14-00759.

Crockett, R., Colella, P., Graves, D., 2011. A Cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions. Journal of Computational Physics 230, 2451–2469. doi:10.1016/j.jcp.2010.12.017.

Drew, D.A., Passman, S.L., 1999. Theory of Multicomponent Fluids. Applied Mathematical Sciences, Springer New York. doi:10.1007/b97678.

Eça, L., Hoekstra, M., Hay, A., Pelletier, D., 2007. A manufactured solution for a two-dimensional steady wall-bounded incompressible turbulent flow. International Journal of Computational Fluid Dynamics 21, 175–188. doi:10.1080/10618560701553436.

Eça, L., Hoekstra, M., Vaz, G., 2012. Manufactured solutions for steady-flow Reynolds-averaged NavierStokes solvers. International Journal of Computational Fluid Dynamics 26, 313–332. doi:10.1080/10618562.2012.717617.

Folkner, D., Katz, A., Sankaran, V., 2014. Design and verification methodology of boundary conditions for finite volume schemes. Computers & Fluids 96, 264–275. doi:10.1016/j.compfluid.2014.03.028.

Garg, R., Galvin, J., Li, T., Pannala, S., 2012. Open-source MFIX-DEM software for gas-solids flows: Part I - verification studies. Powder Technology 220, 122–137. doi:10.1016/j.powtec.2011.09.019. selected papers from the 2010 NETL Multiphase Flow Workshop.

Gidaspow, D., 1994. Multiphase Flow and Fluidization: Continuum and Kinetic Theory Descriptions. Academic Press, San Diego, CA.

Grace, J.R., Taghipour, F., 2004. Verification and validation of CFD models and dynamic similarity for fluidized beds. Powder Technology 139, 99–110. doi:10.1016/j.powtec.2003.10.006.

Hebert, S., Luke, E.A., 2005. Honey, I shrunk the grids! a new approach to CFD verification, in: 43rd AIAA Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics. doi:10.2514/6.2005-685.

Kamm, J., Rider, W., Brock, J., 2003. Combined space and time convergence analysis of a compressible flow algorithm, in: 16th AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics. doi:10.2514/6.2003-4241.

Kleinstreuer, C., 2003. Two-Phase Flow: Theory and Applications. CRC Press.

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

Lun, C.K.K., Savage, S.B., Jeffrey, D.J., Chepurniy, N., 1984. Kinetic theories for granular flow: Inelastic particles in Couette flow and slightly inelastic particles in a general flowfield. Journal of Fluid Mechanics 140, 223–256. doi:10.1017/S0022112084000586.

National Energy Technology Laboratory, 2014. MFIX — Multiphase Flow with Interphase eXchanges. URL: https://mfix.netl.doe.gov/. [Accessed: 2015-05-06].

Oberkampf, W.L., Roy, C.J., 2010. Verification and Validation in Scientific Computing. Cambridge University Press, Cambridge, UK.

Oliveira, P.J., Issa, R.I., 1994. On the numerical treatment of interphase forces in two-phase flow. Numerical Methods in Multiphase Flows, ASME FED 185, 131–140.

Patankar, S., 1980. Numerical Heat Transfer and Fluid Flow. Hemisphere Series on Computational Methods in Mechanics and Thermal Science, CRC Press, New York, USA.

Pelletier, D., 2010. Verification, validation, and uncertainty in computational fluids dynamics. Can. J. Civ. Eng. 37, 1003–1013. doi:10.1139/l10-032.

Pelletier, D., Roache, P.J., 2000. Verification and Validation of Computational Heat Transfer. John Wiley & Sons Inc.. chapter 13. pp. 417–442. doi:10.1002/9780470172599.ch13.

Roache, P.J., 2009. Fundamentals of Verification and Validation. Hermosa Publishers, Albuquerque, NM.

Roache, P.J., Knupp, P., Steinberg, S., Blaine, R.L., 1990. Experience with benchmark test cases for groundwater flow, in: Benchmark test cases for computational fluid dynamics. ASME FED H00598-1990 93, 49–56.

Roache, P.J., Steinberg, S., 1984. Symbolic manipulation and computational fluid dynamics. AIAA Journal 22, 1390–1394. doi:10.2514/3.8794.

Roy, C.J., 2005. Review of code and solution verification procedures for computational simulation. Journal of Computational Physics 205, 131–156. doi:10.1016/j.jcp.2004.10.036.

Roy, C.J., Nelson, C.C., Smith, T.M., Ober, C.C., 2004. Verification of Euler/Navier-Stokes codes using the method of manufactured solutions. International Journal for Numerical Methods in Fluids 44, 599–620. doi:10.1002/fld.660.

Shunn, L., Ham, F., Moin, P., 2012. Verification of variable-density flow solvers using manufactured solutions. Journal of Computational Physics 231, 3801–3827. doi:10.1016/j.jcp.2012.01.027.

Syamlal, M., 1998. MFIX Documentation: Numerical Technique. Tech. Rep DOE/MC31346-5824, NTIS/DE98002029. National Energy Technology Laboratory. URL: https://mfix.netl.doe.gov/download/mfix/mfix_legacy_manual/numerics.pdf. [Accessed: 2015-05-06].

Syamlal, M., Rogers, W., O'Brien, T., 1993. MFIX Documentation: Theory Guide. Tech. Rep. DOE/METC-95/1013, NTIS/DE95000031. National Energy Technology Laboratory. URL: https://mfix.netl.doe.gov/download/mfix/mfix_legacy_manual/Theory.pdf. [Accessed: 2015-05-06].

Tenneti, S., Garg, R., Subramaniam, S., 2011. Drag law for monodisperse gas-solid systems using particle-resolved direct numerical simulation of flow past fixed assemblies of spheres. International Journal of Multiphase Flow 37, 1072–1092. doi:10.1016/j.ijmultiphaseflow.2011.05.010.

Thomas, J.L., Diskin, B., Rumsey, C.L., 2008. Towards verification of unstructured-grid solvers. AIAA Journal 46, 3070–3079. doi:10.2514/1.36655.

Vedovoto, J.M., Silveira Neto, A.d., Mura, A., Figueira da Silva, L.F., 2011. Application of the method of manufactured solutions to the verification of a pressure-based finite-volume numerical scheme. Computers & Fluids 51, 85–99. doi:10.1016/j.compfluid.2011.07.014.

Veluri, S.P., Roy, C.J., Luke, E.A., 2012. Comprehensive code verification techniques for finite volume CFD codes. Computers & Fluids 70, 59–72. doi:10.1016/j.compfluid.2012.04.028.

Wolfram Research Inc., 2013. Mathematica, version 9.0.1. URL: http://www.wolfram.com/mathematica/. [Accessed: 2015-05-06].

Appendix A. Functions and constants in manufactured solutions

Note: The amplitude constants employ SI units while the frequency constants are dimensionless. Pressure (P_g) is the gauge pressure.

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

Variable, ϕ	$f_{\phi x}$	$f_{\phi y}$	$f_{\phi z}$	$f_{\phi xy}$	$f_{\phi yz}$	$f_{\phi zx}$
u_g	sin	cos	cos	cos	sin	cos
v_g	sin	cos	cos	cos	sin	cos
w_g	cos	sin	cos	sin	sin	cos
P_g	cos	cos	sin	cos	sin	cos
T_g	cos	cos	sin	cos	sin	cos
T_s	cos	cos	sin	cos	sin	cos
Θ_s	cos	cos	sin	cos	sin	cos
ε_s	cos	cos	sin	-	-	-
Variable, ϕ	$a_{\phi x}$	$a_{\phi y}$	$a_{\phi z}$	$a_{\phi xy}$	$a_{\phi yz}$	$a_{\phi zx}$
u_g	0.5	0.85	0.4	0.6	0.8	0.9
v_g	0.8	0.8	0.5	0.9	0.4	0.6
w_g	0.85	0.9	0.5	0.4	0.8	0.75
P_g	0.4	0.45	0.85	0.75	0.7	0.8
T_g	0.75	1.25	0.8	0.65	0.5	0.6
T_s	0.5	0.9	0.8	0.5	0.65	0.4
Θ_s	0.8	1.25	0.7	0.5	0.6	0.7
ε_s	0.4	0.5	0.5	-	-	-

Table A.2: Amplitude constants used in the subsonic, incompressible manufactured solutions

Variable, ϕ	ϕ_0	ϕ_x	ϕ_y	ϕ_z	ϕ_{xy}	ϕ_{yz}	ϕ_{zx}
u_g	7	3	-4	-3	2	1.5	2
v_g	9	-5	4	5	-3	2.5	3.5
w_g	8	-4	3.5	4.2	-2.2	2.1	2.5
P_g	100	20	-50	20	-25	-10	10
T_g	350	10	-30	20	-12	10	8
T_s	300	15	-20	15	-10	12	10
Θ_s	100	5	-10	12	-8	10	7
ε_s	0.3	0.06	0.1	0.06	-	-	-