Verification of a High-Order FEM-based CFD Code using the Method of Manufactured Solutions

Seyi Festus OLATOYINBO*

*Corresponding author
National Space Research and Development Agency,
Abuja FCT, Nigeria,
sfo0001@alumni.uah.edu

DOI: 10.13111/2066-8201.2023.15.2.8

Received: 01 April 2023/ Accepted: 18 April 2023/ Published: June 2023

Abstract: *A high-order computational fluid dynamics (CFD) code capable of solving compressible turbulent flow problems was developed. The CFD code employs the Flowfield Dependent Variation (FDV) scheme implemented in a Finite Element Method (FEM) framework. The FDV scheme is basically derived from the Lax-Wendroff Scheme (LWS) involving the replacement of LWS's explicit time derivatives with a weighted combination of explicit and implicit time derivatives. The code utilizes linear, quadratic and cubic isoparametric quadrilateral and hexahedral Lagrange finite elements with corresponding piecewise shape functions that have formal spatial accuracy of second-order, third-order and fourth-order, respectively. In this paper, the results of observed order-of-accuracy of the implemented FDV FEM-based CFD code involving grid and polynomial order refinements on uniform Cartesian grids are reported. The Method of Manufactured Solutions (MMS) is applied to governing 2-D Euler and Navier-Stokes equations for flow cases spanning both subsonic and supersonic flow regimes. Global discretization error analyses using discrete L_2 norm show that the spatial order-of-accuracy of the FDV FEM-based CFD code converges to the shape function polynomial order plus one, in excellent agreement with theory. Uniquely, this procedure establishes the wider applicability of MMS in verifying the spatial accuracy of a high-order CFD code.*

Key Words: *code verification, spatial order-of-accuracy, flowfield dependent variation scheme, method of manufactured solutions, finite element method, CFD code*

1. INTRODUCTION

The use of computer simulations of physical processes has dramatically increased in the last few decades particularly in the areas of designing, modelling, optimization and virtual prototyping of engineering systems, among others. Due to the resultant effects of modeling and simulation predictions on complex and high-risk engineering systems, the needs for substantial improvement in the credibility of the computational results arising thereof are of great concern to engineering designers, project managers and the decision makers [1]. Verification procedure is one of the tools for building and quantifying confidence in computational simulations and computer programs [2]. Verification is essentially a mathematical procedure that deals with the assessment of the accuracy of the solution to a computational model by comparison with a known solution [1, 3]. Basically, some strategies involved in verification include identification, quantification and reduction of errors in the computational model and its solution [4]. Code verification is directed towards identifying and

removing bugs in the source code, inherent errors in numerical algorithms and improving the computer program using software quality assurance practices [1, 2]. The code verification can be carried out using grid refinement study to bring out potential errors or through comparison between numerical results of two codes, which employ different numerical methods [5]. This verification needs to be performed only once for all the code options exercised unless the code is modified [5, 6]. A general and versatile approach to code verification is the use of Method of Manufactured Solutions (MMS) [3, 7]. Instead of trying to find an exact solution to a system of partial differential equations, the fundamental idea is to "manufacture" an exact solution *a priori*. This MMS does not require that the exact solution be related to a physically realistic problem [2, 8]. Though, it is necessary that the structure of the exact solution be sufficiently complex to exercise all the terms being tested in the governing equations [3].

One of the earliest published articles in employing manufactured solutions to identify coding errors was the work of Shih [9]. Shih et al. [10] applied the MMS to incompressible Navier-Stokes equations for laminar two-dimensional (2-D) flow. The authors generated an exact solution to the lid-driven problem for an arbitrary Reynolds number. The coupling of manufactured solutions with grid refinement for evaluating the observed order-of-accuracy was credited to Roache and Steinberg [5, 11]. In their pioneering code verification study, Roache et al. used the MMS to verify a code generating 3-D transformations for elliptic partial differential equations. Salari and Knupp [12] have systematically applied the MMS to compressible and incompressible Navier-Stokes equations. An extensive discussion of the manufactured solutions generated for code verification is presented in a book published by the authors [7]. Roy et al. [13] used the MMS to verify two compressible, finite volume-based CFD codes, namely Premo and WIND. In their work, the authors successfully verified the second-order spatial accuracy of both codes for 2-D inviscid Euler and laminar Navier-Stokes equations on uniform Cartesian grids. Based on the few reviewed literature, the need to conduct rigorous code verification study as a prerequisite for design, model validation and analysis studies cannot be over-emphasized [14]. The present code verification study aims at investigating the high-order spatial accuracy of the FDV scheme implemented in a compressible, FEM-based CFD code for various grid and polynomial order refinements, spanning both supersonic and subsonic flow regimes. Furthermore, this work also seeks to establish the wider applicability of MMS to verifying spatial accuracy of a high-order CFD code.

2. GOVERNING EQUATIONS

The conservative form of 2-D, compressible Euler equations can be written as

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

$$\frac{\partial(\rho u)}{\partial t}+\frac{\partial(\rho u^2+p)}{\partial x}+\frac{\partial(\rho uv)}{\partial y}=s_x,\frac{\partial(\rho v)}{\partial t}+\frac{\partial(\rho vu)}{\partial x}+\frac{\partial(\rho v^2+p)}{\partial y}=s_y \tag{2}$$

$$\frac{\partial(\rho e_t)}{\partial t}+\frac{\partial(\rho u e_t+pu)}{\partial x}+\frac{\partial(\rho v e_t+pv)}{\partial y}=s_e \tag{3}$$

where ρ is the fluid density, u and v are the Cartesian velocity components, p is the static pressure, and e_t is the total energy. The first term in Eq. (1) - (3) is the unsteady term, followed by the convective terms in x and y directions, respectively, and a general source term is added

at the right-hand side. Closure is achieved by adding two auxiliary energy relations for a calorically perfect gas. Internal energy is given by

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

Total energy is given by

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

The perfect gas equation of state is assumed and given by

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

where T is the temperature (T = 298K), γ is the ratio of specific heats (γ = 1.4) and R is the specific gas constant (R = 287.0 J/kgK).
For compressible viscous flows, Eq. (1) for mass conservation is still valid; whereas the conservative form of the 2-D momentum and energy equations can be written as

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p - \tau_{xx})}{\partial x} + \frac{\partial(\rho uv - \tau_{xy})}{\partial y} = s_x \quad \frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho vu - \tau_{xy})}{\partial x} + \frac{\partial(\rho v^2 + p - \tau_{yy})}{\partial y} = s_y \tag{7}$$

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

The 2-D viscous stress tensor τ_{ij} is given by

$$\tau_{xx} = \tfrac{2}{3}\mu\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right), \qquad \tau_{yy} = \tfrac{2}{3}\mu\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right) \quad \tau_{xy} = \mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) \tag{9}$$

The Fourier's law of heat transfer by conduction is assumed, the heat flux vector q_i is given by

$$q_x = -k\frac{\partial T}{\partial x}, \quad q_y = -k\frac{\partial T}{\partial y} \tag{10}$$

For the purpose of viscous flow simulations, the dynamic viscosity is chosen to be a large constant value ($\mu = 10\ Ns/m^2$ for subsonic and $\mu = 20\ Ns/m^2$ for supersonic simulations) so that the diffusion and the convection terms can be balanced [13]. The thermal conductivity is given by:

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

where Pr is the molecular Prandtl number taken to be unity (Pr = 1.0).

3. REVIEW OF FLOWFIELD DEPENDENT VARIATION SCHEME

The compressible Navier-Stokes system of equations can be expressed in conservative form as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_i}{\partial x_i} + \frac{\partial \boldsymbol{G}_i}{\partial x_i} = 0 \tag{12}$$

$$\text{where } \boldsymbol{U} = \begin{bmatrix} \rho \\ \rho v_j \\ \rho e_t \end{bmatrix}, \ \boldsymbol{F_i} = \begin{bmatrix} \rho v_i \\ \rho v_i v_j + p\delta_{ij} \\ v_i(\rho e_t + p) \end{bmatrix}, \ \boldsymbol{G}_i = \begin{bmatrix} 0 \\ -\tau_{ij} \\ -\tau_{ij} v_j + q_i \end{bmatrix} \tag{13}$$

From Eqs. (12) and (13), $\boldsymbol{U}$, $\boldsymbol{F}_i$ and $\boldsymbol{G}_i$ stand for the vectors of conserved variables, convective flux and diffusive flux, respectively. Basically, the Flowfield Dependent Variation (FDV) scheme may be considered as a variant of the Lax-Wendroff Scheme (LWS) that is obtained through replacement of LWS's explicit time derivatives by a weighted combination of explicit and implicit time derivatives [15, 16]. This replacement is carried out to ensure numerical stability and monotonicity of the FDV scheme [17].

Performing a Taylor series expansion of $\boldsymbol{U}^{n+1}$ about $\boldsymbol{U}^n$ yields

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \Delta t \frac{\partial \boldsymbol{U}^n}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 \boldsymbol{U}^n}{\partial t^2} + O(\Delta t^3) \tag{14}$$

where Δt is the time step between solutions $\boldsymbol{U}^n/\boldsymbol{U}^{n+1}$ and n denotes the time index. Furthermore, replacing the explicit time derivatives in Eq. (14) with a weighted explicit-implicit combination of FDV parameters, s_a and s_b, gives

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \Delta t \left[(1 - s_a) \frac{\partial \boldsymbol{U}^n}{\partial t} + s_a \frac{\partial \boldsymbol{U}^{n+1}}{\partial t} \right] + \frac{\Delta t^2}{2} \left[(1 - s_b) \frac{\partial^2 \boldsymbol{U}^n}{\partial t^2} + s_b \frac{\partial^2 \boldsymbol{U}^{n+1}}{\partial t^2} \right] + O(\Delta t^3) \tag{15}$$

where $0 \leq s_a \leq 1$ and $0 \leq s_b \leq 1$.

These FDV parameters act as weighting functions between explicit and implicit methods. If $s_a = s_b = 0$ (e.g., in the regions of zero gradients in a flowfield), the method is fully explicit similar to LWS; whereas if $s_a = s_b = 1$ (e.g., in the regions of high gradients in flow variables), the method becomes fully implicit.

By rearranging Eq. (12) through the separation of time and partial derivatives, the following equation results

$$\frac{\partial \boldsymbol{U}}{\partial t} = -\frac{\partial \boldsymbol{F}_i}{\partial x_i} - \frac{\partial \boldsymbol{G}_i}{\partial x_i} \tag{16}$$

Taking a time derivative of Eq. (16), interchanging spatial and time derivatives, and recognizing that $\boldsymbol{F}_i = \boldsymbol{F}_i(\boldsymbol{U})$, and $\boldsymbol{G}_i = \boldsymbol{G}_i(\boldsymbol{U}, \boldsymbol{U}_{,j})$, where $\boldsymbol{U}_{,j} = \partial \boldsymbol{U}/\partial x_j$, gives

$$\frac{\partial \boldsymbol{U}}{\partial t} = \frac{\partial}{\partial x_i} \left[(\boldsymbol{a}_i + \boldsymbol{b}_i) \left\{ \frac{\partial \boldsymbol{F}_j}{\partial x_j} + \frac{\partial \boldsymbol{G}_j}{\partial x_j} \right\} \right] + \frac{\partial^2}{\partial x_i \partial x_j} \left[\boldsymbol{c}_{ij} \left\{ \frac{\partial \boldsymbol{F}_k}{\partial x_k} + \frac{\partial \boldsymbol{G}_k}{\partial x_k} \right\} \right] \tag{17}$$

where

$$\boldsymbol{a}_i = \frac{\partial \boldsymbol{F}_i}{\partial \boldsymbol{U}}, \qquad \boldsymbol{b}_i = \frac{\partial \boldsymbol{G}_i}{\partial \boldsymbol{U}}, \qquad \boldsymbol{c}_{ij} = \frac{\partial \boldsymbol{G}_i}{\partial \boldsymbol{U}_{,j}} \tag{18}$$

Note that $\boldsymbol{a}_i$, $\boldsymbol{b}_i$ and $\boldsymbol{c}_{ij}$ are the Jacobians of convection, diffusion and diffusion gradient, respectively.

Substituting Eqs. (16) and (17) into Eq. (15), and neglecting the product of $\boldsymbol{c}_{ij}$ with the third-order spatial derivatives of $\boldsymbol{F}_k$ and $\boldsymbol{G}_k$ yields

$$\Delta \boldsymbol{U}^{n+1} = \Delta t \left[\left\{-\frac{\partial \boldsymbol{F}_i^n}{\partial x_i} - \frac{\partial \boldsymbol{G}_i^n}{\partial x_i}\right\} + \mathrm{s_a}\left\{-\frac{\partial \Delta \boldsymbol{F}_i^{n+1}}{\partial x_i} - \frac{\partial \Delta \boldsymbol{G}_i^{n+1}}{\partial x_i}\right\}\right] + \frac{\Delta t^2}{2}\frac{\partial}{\partial x_i}\left[(\boldsymbol{a}_i + \boldsymbol{b}_i)\left\{\frac{\partial \boldsymbol{F}_j^n}{\partial x_j} + \frac{\partial \boldsymbol{G}_j^n}{\partial x_j}\right\}\right] + \frac{\Delta t^2}{2}\mathrm{s_b}\frac{\partial}{\partial x_i}\left[(\boldsymbol{a}_i + \boldsymbol{b}_i)\left\{\frac{\partial \Delta \boldsymbol{F}_j^{n+1}}{\partial x_j} + \frac{\partial \Delta \boldsymbol{G}_j^{n+1}}{\partial x_j}\right\}\right] \tag{19}$$

where $\Delta \boldsymbol{U}^{n+1} = \boldsymbol{U}^{n+1} - \boldsymbol{U}^n$, $\Delta \boldsymbol{F}^{n+1} = \boldsymbol{F}^{n+1} - \boldsymbol{F}^n$ and $\Delta \boldsymbol{G}^{n+1} = \boldsymbol{G}^{n+1} - \boldsymbol{G}^n$.
In a bid to handle the effects of the FDV parameters on convection and diffusion terms separately, $\mathrm{s_a}$ and $\mathrm{s_b}$ are split into $\mathrm{s_{a,conv}}/\mathrm{s_{a,diff}}$ and $\mathrm{s_{b,conv}}/\mathrm{s_{b,diff}}$, respectively and reassigned as follow:

$$\mathrm{s_a}\Delta \boldsymbol{F}_i \Rightarrow s_{\mathrm{a,conv}}\Delta \boldsymbol{F}_i, \qquad \mathrm{s_a}\Delta \boldsymbol{G}_i \Rightarrow s_{\mathrm{a,diff}}\Delta \boldsymbol{G}_i \tag{20}$$

$$\mathrm{s_b}\Delta \boldsymbol{F}_i \Rightarrow s_{\mathrm{b,conv}}\Delta \boldsymbol{F}_i, \qquad \mathrm{s_b}\Delta \boldsymbol{G}_i \Rightarrow s_{\mathrm{b,diff}}\Delta \boldsymbol{G}_i \tag{21}$$

These implicitness parameters are apportioned physical functions when their values are calculated from relevant flow quantities like Mach number (M) for convection and Reynolds number (Re) for diffusion. The first-order FDV parameters, $s_{\mathrm{a,conv}}$ and $s_{\mathrm{a,diff}}$, are dependent on the flowfield, while the second-order parameters, $s_{\mathrm{b,conv}}$ and $s_{\mathrm{b,diff}}$, have a power-law dependence on the first-order parameters as presented in Eqs. (22) – (25).

$$s_{\mathrm{a,conv}} = \begin{cases} \min(r,1) & r > \alpha \\ 0 & r > \alpha,\ M_{\min} \neq 0 \\ 1 & M_{\min} = 0 \end{cases} \tag{22}$$

$$s_{\mathrm{b,conv}} = \tfrac{1}{2}\left(1 + s_{a,\mathrm{conv}}^{\eta}\right), \text{ where } r = \sqrt{M_{\max}^2 - M_{\min}^2}/M_{\min} \tag{23}$$

$$s_{\mathrm{a,diff}} = \begin{cases} \min(r,1) & r > \alpha \\ 0 & r < \alpha,\ Re_{\min} \neq 0 \\ 1 & Re_{\min} = 0 \end{cases} \tag{24}$$

$$s_{\mathrm{b,diff}} = \tfrac{1}{2}\left(1 + s_{\mathrm{a,diff}}^{\eta}\right), \text{ where } r = \sqrt{Re_{\max}^2 - Re_{\min}^2}/Re_{\min} \tag{25}$$

The highest and lowest values of M and Re are evaluated from the nodal values within the element. For this present study, the parameters $\alpha = 0.001$ and $\eta = 0.1$ are used. The modification to the variation parameters, $\mathrm{s_a}$ and $\mathrm{s_b}$, helps in maintaining the second-order temporal accuracy of the scheme, thereby enhancing stability of flows with stronger shocks [18]. All the FDV parameters are updated iteratively at each time step.
By rewriting Eq. (19) in terms of FDV parameters $s_{\mathrm{a,conv}}$, $s_{\mathrm{a,diff}}$ and $s_{\mathrm{b,diff}}$, $s_{\mathrm{b,diff}}$, one obtains

$$\Delta \boldsymbol{U}^{n+1} = -\Delta t \left[\left\{\frac{\partial \boldsymbol{F}_i^n}{\partial x_i} + \frac{\partial \boldsymbol{G}_i^n}{\partial x_i}\right\} + \left\{s_{\mathrm{a,conv}}\frac{\partial \Delta \boldsymbol{F}_i^{n+1}}{\partial x_i} + s_{\mathrm{a,diff}}\frac{\partial \Delta \boldsymbol{G}_i^{n+1}}{\partial x_i}\right\}\right] + \frac{\Delta t^2}{2}\frac{\partial}{\partial x_i}\left[(\boldsymbol{a}_i + \boldsymbol{b}_i)\left\{\frac{\partial \boldsymbol{F}_j^n}{\partial x_j} + \frac{\partial \boldsymbol{G}_j^n}{\partial x_j}\right\}\right] + \frac{\Delta t^2}{2}\frac{\partial}{\partial x_i}\left[(\boldsymbol{a}_i + \boldsymbol{b}_i)\left\{s_{\mathrm{b,conv}}\frac{\partial \Delta \boldsymbol{F}_j^{n+1}}{\partial x_j} + s_{\mathrm{b,diff}}\frac{\partial \Delta \boldsymbol{G}_j^{n+1}}{\partial x_j}\right\}\right] \tag{26}$$

Replacing $\Delta \boldsymbol{F}_j^{n+1}$ and $\Delta \boldsymbol{G}_j^{n+1}$ with their respective Jacobians stated in Eq. (18) gives

$$\left[\boldsymbol{I} + \boldsymbol{D}_i\frac{\partial}{\partial x_i} + \boldsymbol{E}_{ij}\frac{\partial^2}{\partial x_i \partial x_j}\right]\Delta \boldsymbol{U}^{n+1} = \boldsymbol{Q}^n \tag{27}$$

where $\boldsymbol{I}$ is the identity matrix.
Finally, presenting Eq. (26) in residual form results in the following expression:

$$\boldsymbol{R} = \Delta \boldsymbol{U}^{n+1} + \boldsymbol{D}_i^n \Delta \boldsymbol{U}_{,i}^{n+1} + \boldsymbol{E}_{ij}^n \Delta \boldsymbol{U}_{,ij}^{n+1} - \boldsymbol{Q}^n = O(\Delta t^3) \tag{28}$$

$$\text{such that } \boldsymbol{D}_i^n = \Delta t\left(s_{\text{a,conv}} \boldsymbol{a}_i + s_{\text{a,diff}} \boldsymbol{b}_i\right) \tag{29}$$

$$\boldsymbol{E}_{ij}^n = \Delta t s_{\text{a,diff}} \boldsymbol{c}_{ij} - \frac{\Delta t^2}{2}\left[(\boldsymbol{a}_i + \boldsymbol{b}_i)(s_{\text{b,conv}} \boldsymbol{a}_j + s_{\text{b,diff}} \boldsymbol{b}_j)\right] \tag{30}$$

$$\boldsymbol{Q}^n = \Delta t\left(\boldsymbol{F}_{i,i}^n + \boldsymbol{G}_{i,i}^n\right) - \frac{\Delta t^2}{2}(\boldsymbol{a}_i + \boldsymbol{b}_i)\left(\boldsymbol{F}_{j,ji}^n + \boldsymbol{G}_{j,ji}^n\right) \tag{31}$$

Numerical stability mechanisms are inbuilt into FDV scheme a priori using the implicitness parameters, hence finite element discretization is sufficient. By employing standard Galerkin method, the following expression is obtained:

$$\int_\Omega \Phi_\alpha \boldsymbol{R}(\boldsymbol{U}, \boldsymbol{F}_i, \boldsymbol{G}_i) d\Omega = 0 \tag{32}$$

Note that Φ_α stands for nodal shape function with global index α. By substituting Eq. (28) into Eq. (32), integrating by parts and arranging compactly yields the assembled finite element equations in global form, which can be expressed as:

$$\left(A_{\alpha\beta rs}^n + B_{\alpha\beta rs}^n\right)\Delta U_{\beta s}^{n+1} = H_{\alpha r}^n + N_{\alpha r}^n \tag{33}$$

$$A_{\alpha\beta rs}^n = \int_\Omega \left(\Phi_\alpha \Phi_\beta \delta_{rs} - \Phi_{\alpha,i} \Phi_\beta D_{irs}^n - \Phi_{\alpha,i} \Phi_{\beta,j} E_{ijrs}^n\right) d\Omega \tag{34}$$

$$B_{\alpha\beta rs}^n = \int_\Gamma \left(\overset{*}{\Phi}_\alpha \overset{*}{\Phi}_\beta \overset{*n}{D}_{irs} + \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_{\beta,j} \overset{*n}{E}_{ijrs}\right) n_i d\Gamma \tag{35}$$

$$H_{\alpha r}^n = \int_\Omega \left(\Delta t \Phi_{\alpha,i}(F_{ir}^n + G_{ir}^n) - \frac{\Delta t^2}{2} \Phi_{\alpha,i}(a_{irs} + b_{irs})\left(F_{js,j}^n + G_{js,j}^n\right)\right) d\Omega \tag{36}$$

$$N_{\alpha r}^n = \int_\Gamma \left(-\Delta t \overset{*}{\Phi}_\alpha \left(\overset{*n}{F}_{ir} + \overset{*n}{G}_{ir}\right) + \frac{\Delta t^2}{2} \overset{*}{\Phi}_\alpha (a_{irs} + b_{irs})\left(\overset{*n}{F}_{js,j} + \overset{*n}{G}_{js,j}\right)\right) n_i d\Gamma \tag{37}$$

In Eqs. (34) - (37), α and β stand for global nodal indices; $i, j = 1, 2, 3$ represent physical coordinate indices; $r, s = 1, 2, \ldots, 5$ denote indices for the five conserved variables; δ_{rs} stands for Kronecker delta; the superscript * denotes variable along the inter-element contour; n_i is normal to the element surface, the integral over Ω and Γ represent the element and contour integral, respectively. In Eqs. (35) and (37), the contour integrals cancel out each other along inter-element boundaries insuring flux conservation. Through these contours, Neumann boundary conditions can be implemented. Similarly, Dirichlet boundary conditions can be imposed through element-by-element discretization procedure.

4. ORDER-OF-ACCURACY ANALYSIS

The code verification test determines whether the observed order-of-accuracy of the discretized solutions converges to the formal order-of-accuracy of the discretization method as the grid is refined [2, 6, 13]. The formal order-of-accuracy of the discretization method may be unattainable due to coding mistakes, defective numerical algorithms, subtleties in nonlinear problems, non-attainment of asymptotic grid convergence range, among others [19,20]. For FEM employed in the CFD code, the formal order-of-accuracy is determined by interpolation theory. In order to reduce computational time and memory requirements, the code utilizes

higher order FEM that can use much smaller grids to accurately capture complex physics such as laminar instabilities and turbulent phenomena.

In a bid to evaluate the observed order-of-accuracy, there is need to calculate the discretization errors in the simulation results. Discretization Error (DE) is a measure of the difference between the exact solution to the governing equations and the numerical solution to the discretized equations [6]. Considering a series expansion of the discretization error in terms of h_j, a measure of the element size on grid level *j*, then DE can be written as

$$DE_j = \varphi_j - \varphi_{exact} = Ch_j^{np} + HOT \tag{38}$$

where φ_j is the numerical solution on grid level *j*, φ_{exact} is the exact solution, *C* is the coefficient of the leading-error term, *np* is the observed order-of-accuracy and HOT are the higher order terms. Generally, it is assumed that the leading-order error term dominates the total discretization error, thus the grid spacing, *h*, is in the asymptotic range [3]. Therefore, HOT can be neglected. The DE equation for a fine and coarse grid levels *k* and *k*+1, respectively, can be expressed as

$$DE_k = \varphi_k - \varphi_{exact} = Ch_k^{np} \text{ and } DE_{k+1} = \varphi_{k+1} - \varphi_{exact} = Ch_{k+1}^{np} \tag{39}$$

Since the exact solution φ_{exact} is known, these two equations can be solved for the observed order-of-accuracy *np*. Let the grid refinement ratio, ratio of coarse grid to fine grid spacing, be denoted by $r = h_{k+1}/h_k$, then an expression for the observed order-of-accuracy takes the form:

$$np^k = \frac{In(DE_{k+1}/DE_k)}{In(r)} \tag{40}$$

Fundamentally, the formal order-of-accuracy, fnp, and the shape function polynomial order, p, are related by $fnp \approx p + 1$ from FEM theory [21]. The observed order-of-accuracy can be evaluated either locally within the solution domain or globally through the use of any appropriate error norms. In the present code verification study, the discrete L_2 norm is used to examine the behaviour of global discretization error. Knowing that the computation of L_2 norm involves some averaging and the computed results tend to give better plots. The discrete L_2 norm for grid level *k* is written as

$$L_{2,k} = \sqrt{\frac{\sum_{i=1}^{N} |DE_{k,i}|^2}{N}} \tag{41}$$

where index *i* varies over all the grid nodes *N* in space with exception of the Dirichlet boundary nodes where DE is identically zero [22].

5. FORMS OF MANUFACTURING SOLUTIONS

Code verification is principally a mathematical procedure to ascertain that the numerical solution is a good representation of the exact solution to the continuum governing equations being solved [22]. The manufactured solutions employed herein have been chosen with no recourse for their physical reality, but the solution structures are sufficiently complex and smooth to exercise all the terms in the governing equations. In accordance with Roy et al. [13, 22, 23], the chosen manufactured solutions used in the present code verification test take the form:

$$\phi(x,y) = a_0 + a_1\theta_s\left(\frac{b_1\pi x}{L}\right) + a_2\theta_s\left(\frac{b_2\pi y}{L}\right) + a_3\theta_s\left(\frac{b_3\pi xy}{L^2}\right) \tag{42}$$

where $\phi = [\rho, u, v, p]^T$ represents any of the primitive variables, $\theta_s(.)$ denotes the sine or cosine functions, L is the domain length and a_i and b_i are constant coefficients. The specific form of the primitive variables for both Euler and Navier-Stokes equations for all test cases considered in this study are listed below:

$$\begin{aligned}
\rho(x,y) &= \rho_0 + \rho_1 sin\left(\frac{b_1\pi x}{L}\right) + \rho_2 cos\left(\frac{b_2\pi y}{L}\right) + \rho_3 cos\left(\frac{b_3\pi xy}{L^2}\right),\\
u(x,y) &= u_0 + u_1 sin\left(\frac{b_1\pi x}{L}\right) + u_2 cos\left(\frac{b_2\pi y}{L}\right) + u_3 cos\left(\frac{b_3\pi xy}{L^2}\right),\\
v(x,y) &= v_0 + v_1 cos\left(\frac{b_1\pi x}{L}\right) + v_2 sin\left(\frac{b_2\pi y}{L}\right) + v_3 cos\left(\frac{b_3\pi xy}{L^2}\right),\\
p(x,y) &= p_0 + p_1 cos\left(\frac{b_1\pi x}{L}\right) + p_2 sin\left(\frac{b_2\pi y}{L}\right) + p_3 sin\left(\frac{b_3\pi xy}{L^2}\right).
\end{aligned} \tag{43}$$

All constants used in the simulations of subsonic and supersonic Euler and Navier-Stokes cases are obtained from cited references [13] and [22].

6. RESULTS AND DISCUSIONS

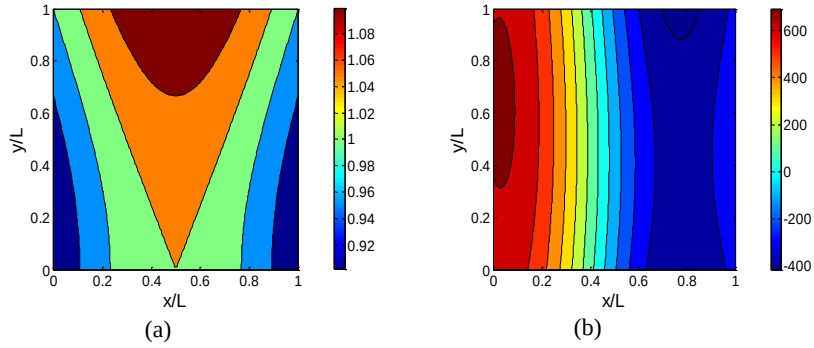
The observed order-of-accuracy results for 2-D Euler and Navier-Stokes equations, spanning both supersonic and subsonic flow regimes, are presented and discussed in this section. The test cases were chosen to investigate the effects of both *h*- and *p*-refinements. For all the four test cases considered, the numerical solutions are obtained on the computational domain comprising uniform Cartesian grids such that: $0 \leq x/L \leq 1$ and $0 \leq y/L \leq 1$ with $L = 1.0\,m$. Five different computational grids with grid refinement ratio of two ($r = 2$) are used for the grid convergence test and the shape function polynomial order p is varied between 1 and 3. The grid sizes are given in Table 1 where the grid spacing, h/h_{min}, is the ratio of element sizes at the *k*th grid level to the finest grid level.

Table 1 - MMS grid and polynomial order refinements

Grid Level	*h*-refinements	h/h_{min}	*p*-refinements
1	129^2	1	1, 2, 3
2	65^2	2	1, 2, 3
3	33^2	4	1, 2, 3
4	17^2	8	1, 2, 3
5	9^2	16	1, 2, 3

6.1 EULER EQUATIONS

The MMS approach is applied to the Euler equations given by Eqs. (1) - (3), which constitute the conservation of mass, momentum and energy for an inviscid fluid, coupled with the auxiliary relations stated in Eqs. (4) - (6). The general form of the chosen manufactured solution is given in Eq. (42), while the specific form for the primitive variables for all the test cases considered are listed in Eq. (43). The constants used in the simulation of the supersonic and subsonic Euler cases were obtained from cited references [13] and [22]. The manufactured solution for density (ρ) and the generated mass source term in supersonic Euler case are shown in Fig. 1.



(a) (b)

Fig. 1 - Manufactured solution of (a) ρ and (b) mass source term for supersonic Euler case

The Euler equations were applied to the chosen manufactured solutions using a code written in symbolic processing with MATLAB™ to generate the analytical source terms, which were later converted to FORTRAN code. The governing equations (Eqs. (1) - (6)) coupled with the analytical source terms were then discretized and solved numerically. The numerical solutions obtained were compared to the exact (manufactured) solution to determine the discretization error in the solutions. All the solutions presented in this study were integrated in time until the L_2 norm of the iterative error approached the iterative convergence tolerance of $1.0E-10$.

For the supersonic Euler case, the L_2 norm of density (ρ) discretization error evaluated from Eq. (41) is presented in Fig. 2 for the five grid levels plotted on a log-log scale. The norm shows trends of second-order, third-order and fourth-order slopes, respectively, for the linear, quadratic and cubic isoparametric finite elements used for spatial discretization as the grid is refined. Also shown in Fig. 2 is the second-order slope for easy comparison. The computed results of the observed order-of-accuracy calculated from Eq. (40) are shown in Fig. 3. These plots confirm that the formal order-of-accuracy of the numerical scheme for linear (second-order), quadratic (third-order) and cubic (fourth-order) elements is reproduced with corresponding *h*- and *p*-refinements. Similar behaviour was observed for the other conserved variables as well (i.e., ρu, ρv and ρe_t).

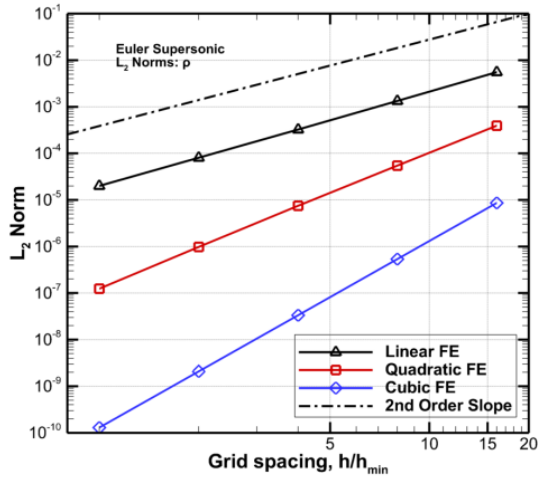


Fig. 2 - Behaviour of density discretization error norm with *h*- and *p*-refinements for supersonic Euler case

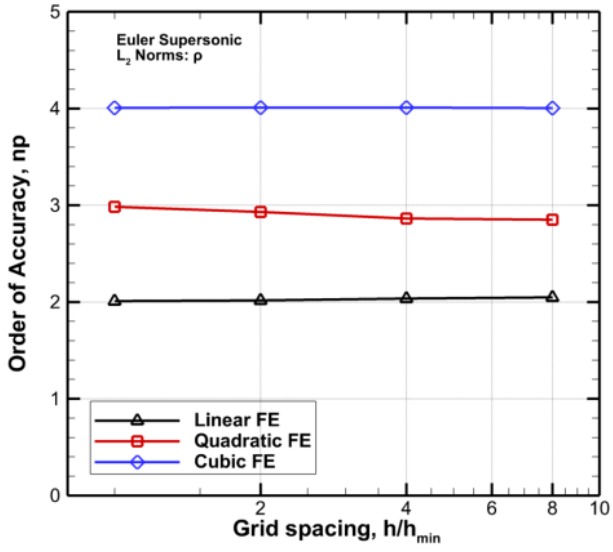


Fig. 3 - Observed order-of-accuracy of density discretization error norm with *h*- and *p*-refinements for supersonic Euler case

Figure 4 shows the behaviour of L_2 norm for x-momentum (ρu) discretization error for the subsonic Euler case. Again, the plots of the computed error norm exhibit second-order, third-order and fourth-order behaviour on all grid levels. The observed order-of-accuracy, plotted in Fig. 5, confirms that the code is reproducing the formal spatial order-of-accuracy of the numerical scheme. In fact, the observed order-of-accuracy of the quadratic finite element appears to be slightly higher than third-order in this subsonic Euler case. The error norms of the other conserved variables showed similar behaviour as well.

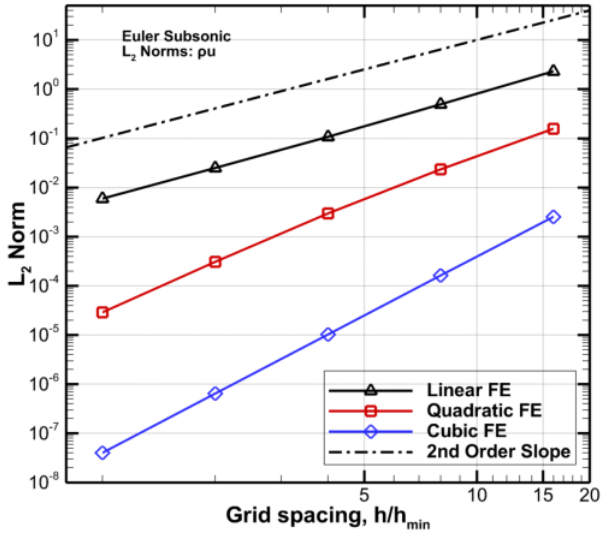


Fig. 4 - Behaviour of x-momentum (ρu) discretization error norms with *h*- and *p*-refinements for subsonic Euler case

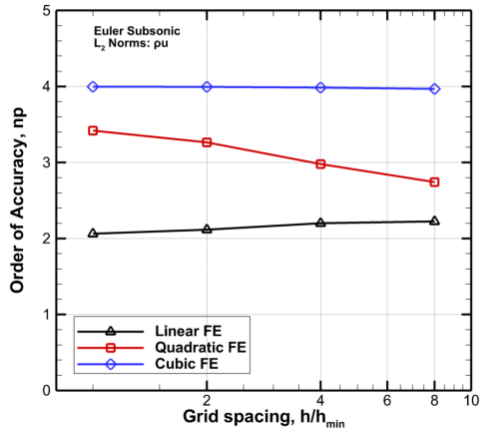


Fig. 5 - Observed order-of-accuracy of x-momentum (ρu) discretization error norm with h- and p-refinements for subsonic Euler case

6.2 NAVIER-STOKES EQUATIONS

The MMS approach is similarly applied to the Navier-Stokes equations comprising of Eqs. (1), (7) and (8) along with the auxiliary relations given in Eqs. (4) - (6) and (9) - (11). The constants used in the manufactured solutions for the test cases involving supersonic and subsonic flows were obtained from cited references [13] and [22].

In the case of supersonic Navier-Stokes simulations, the dynamic viscosity was chosen as $\mu = 20\ Ns/m^2$ in a bid to strike a balance between the convective and diffusive terms, this will minimize any possibility of a "false positive" on the order-of-accuracy test [2, 21]. The behaviour of the computed L_2 norm of discretization error for the conserved variable (ρe_t) is shown in Fig. 6. The norm shows trends of second-order, third-order and fourth-order behaviour in nearly all grid levels, except for some deviations observed in quadratic finite element plot. The observed order-of-accuracy results are plotted in Fig. 7. The entire plots asymptotically approach the formal order-of-accuracy of the linear, quadratic and cubic isoparametric elements used for the discretization as the grid is refined. Similar behaviour was observed for other conserved variables.

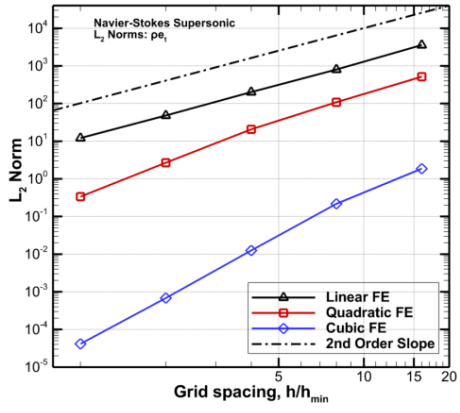


Fig. 6 - Behaviour of energy (ρe_t) discretization error norm with h- and p-refinements for supersonic Navier-Stokes case

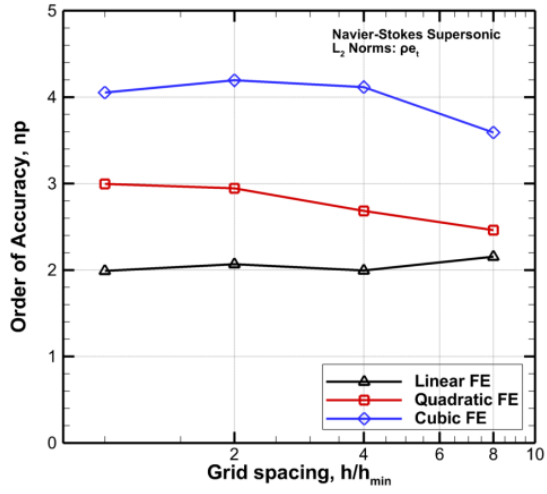
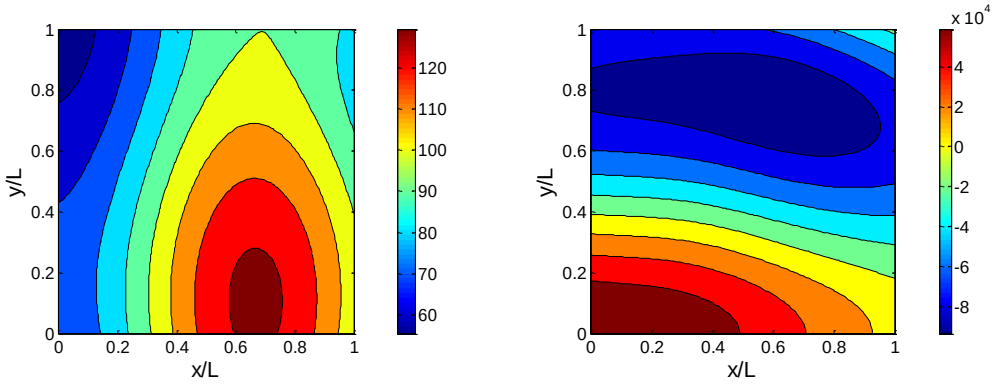


Fig. 7 - Observed order-of-accuracy of energy (ρe_t) discretization error norm with *h*-and *p*-refinements for supersonic Navier-Stokes case

In the subsonic Navier-Stokes case, the dynamic viscosity was chosen as $\mu = 10\ Ns/m^2$. Again, this is done to ensure that the viscous terms are of the same order of magnitude as the convective terms [13].

Figure 8 depicts the manufactured solution of conserved variable (ρv) and the generated y-momentum source term for subsonic Navier-Stokes case.



(a) (b)

Fig. 8 – Manufactured solution of (a) ρv and (b) y-momentum source term for subsonic Navier-Stokes case

The L_2 error norm for the conserved variable (ρv) is plotted in Fig. 9. The norm exhibits second-order, third-order and fourth-order slopes, respectively, for linear, quadratic and cubic isoparametric elements as the grid is refined.

The behaviour of the error norm for spatial convergence is confirmed by the trends of the observed order-of- accuracy shown in Fig. 10.

In fact, the observed order-of-accuracy of the cubic finite element appears to be slightly higher than fourth-order in this subsonic Navier-Stokes case. Other conserved variables exhibited similar behaviour.

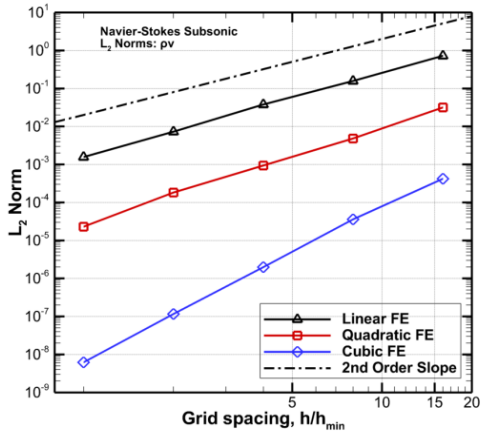


Fig. 9 - Behaviour of y-momentum (ρv) discretization error norms with *h*- and *p*-refinements for subsonic Navier-Stokes case

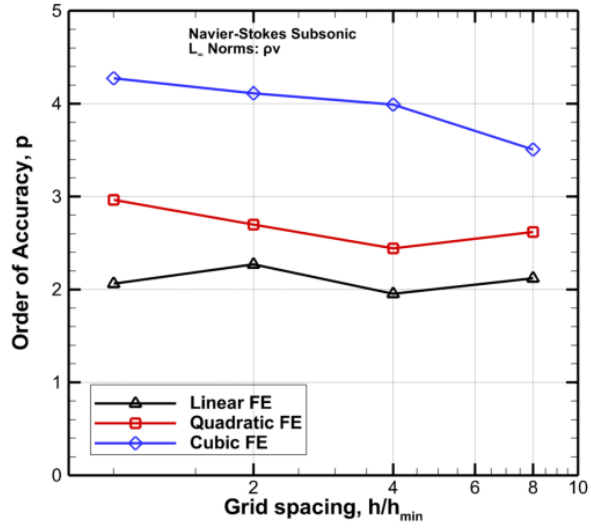


Fig. 10 - Observed order-of-accuracy of y-momentum (ρv) discretization error norms with *h*- and *p*-refinements for subsonic Navier-Stokes case

7. CONCLUSIONS

The method of manufactured solutions has been applied to verify the spatial order-of-accuracy of Flowfield Dependent Variation scheme implemented in a compressible CFD code that employs finite element method framework, which supports higher order quadrilateral and hexahedral isoparametric Langrange elements.

The code verification test was carried out by comparing the numerical solutions to the manufactured solutions on a series of five different consistently-refined grid levels and three different shape function polynomial order. Four test cases examined are governed by 2-D Euler and Navier-Stokes equations spanning both subsonic and supersonic flow regimes with varied *h*- and *p*-refinements on uniform Cartesian grids.

In all the four test cases considered, the observed order-of-accuracy converges to the formal order-of-accuracy of the FDV FEM-based CFD code with corresponding *h*- and *p*-refinements in excellent agreement with finite element theory. Furthermore, this code verification study establishes the wider applicability of the MMS approach in the verification of formal order-of-accuracy of a high-order CFD code. This procedure enhances credibility and confidence that there are no coding mistakes in spatial discretization of the CFD code on uniform Cartesian grids.

The coding options verified by the MMS approach employed in the current code verification study include: inviscid Euler and viscous Navier-Stokes equations, FDV FEM-based scheme's spatial accuracy and the Dirichlet boundary conditions in both subsonic and supersonic flow regimes. Some options that are not verified include solver stability, curvilinear grids and temporal accuracy, among others.

REFERENCES

[1] W. L. Oberkampf and T. G. Trucano, Verification and validation in computational fluid dynamics, *Progress in Aerospace Science*, vol. **38**, no. 3, pp**.** 209-272, 2002.

[2] * * * AIAA, Guide for the verification and validation of computational fluid dynamics simulations, *AIAA G-077-1998(2002)*, eISBN:978-1-56347-285-5, 1998.

[3] P. J. Roache, Code verification by the method of manufactured solutions, *Journal of Fluids Engineering*, vol. **124**, no.1 pp. 4-10, 2002.

[4] W. L. Oberkampf and F. G. Blottner, Issues in computational fluid dynamics code verification and validation, *AIAA Journal*, vol. **36**, no. 5, pp. 687-695, 1998.

[5] S. Steinberg and P. J. Roache, Symbolic manipulation and computational fluid dynamics. *Journal of Computational Physics*, Vol. **57**, No. 2, pp. 251-284, 1985.

[6] C. J. Roy, Review of code and solution verification procedure in computational simulation, *Journal of Computational Physics*, vol. **205**, no. 1, pp. 131-156, 2005.

[7] P. Knupp and K. Salari, *Verification of computer codes in computational science and engineering*, 1st ed. Chapman and Hall/CRC New York, 2002.

[8] C. J. Roy, Grid convergence error analysis for mixed-order numerical schemes, *AIAA Journal*, vol. **41**, no. 4, pp.595-604, 2001.

[9] T. M. Shih, A procedure to debug computer programs, *International Journal of Numerical Methods in Engineering*, vol. **21**, no. 6, pp. 1027-1037, 1985.

[10] T. M. Shih, C. H. Tan and B. C. Hwang, Effects of grid staggering on numerical schemes, *International Journal of Numerical Methods in Fluids*, vol. **9**, no. 2, pp. 193-212, 1989.

[11] P. J. Roache and S. Steinberg, Symbolic manipulation and computational fluid dynamics *AIAA Journal*, vol. **22**, no. 10, pp. 1390-1394, 1984.

[12] K. Salari and P. Knupp, Code verification by the method of solutions *SAND 2000-1444*, Sandia National Laboratories, Albuquerque, New Mexico, 2000.

[13] C. J. Roy, C. C. Nelson, T. M. Smith and C. C. Ober, Verification of euler/navier-stokes codes using the method of manufactured solutions, *International Journal of Numerical Methods in Fluids*, vol. **44**, no. 6, pp. 599-620, 2004.

[14] C. J. Roy, C. L. Rumsey and E. C. Tinoco, Summary data from the sixth AIAA computational fluid dynamics drag prediction workshop: code verification, *Journal of Aircraft*, vol. **55**, no. 4, pp. 1338-1351, 2018.

[15] B. R. Girgis, S. L. Rani and A. Frendi, Flowfield dependent variation method: a numerical scheme for the solution of low-to-high Mach number flow problems, *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. **26**, no. 5, pp. 1486-1525, 2015.

[16] S. F. Olatoyinbo, S. L. Rani and A. Frendi, Large eddy simulation of decaying isotropic turbulence using the flowfield dependent variation method, *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. **27**, no. 1, pp. 235-262, 2017.

[17] S. F. Olatoyinbo, *Investigation of the numerical characteristics of the flowfield dependent variation method*, PhD Dissertation, Department of Mechanical & Aerospace Engineering, University of Alabama in Huntsville, Alabama, USA, 2015.

[18] G. A. Richardson, J. T. Cassibry, T. J. Chung and S. T. Wu, Finite element form of FDV for widely varying flowfields, *Journal of Computational Physics*, vol. **229**, no. 1, pp. 145-167, 2010.

[19] P. J. Roache, Perspective: a method for uniform reporting of grid refinement studies, *Journal of Fluids Engineering*, vol. **116**, no. 3, pp. 405-413, 1994.
[20] P. J. Roache, Verification of codes and calculations, *AIAA Journal*, vol. **36**, no. 5, pp. 696-702, 1998.
[21] O. C. Zienkiewicz, R. L. Taylor and J. Z. Zhu, *The finite element method: its basis and fundamentals*, 6th ed. Elsevier, 2005.
[22] C. J. Roy, T. M. Smith and C. C. Ober, Verification of a compressible CFD code using the method of manufactured solutions, Paper presented at the 32nd AIAA Fluids Dynamics Conference, St. Louis, Missouri, 24-26 June 2002, *AIAA 2002-3110*, 2002.
[23] C. J. Roy, E. Tendean, S. P. Veluri, R. Rifki, E. A. Luke and S. Hebert, Verification of RANS turbulence models in Loci-CHEM using the method of manufactured solutions, Paper presented at the 18th AIAA Computational Fluids Dynamics Conference, Miami, Florida, 25-28 June 2007, *AIAA 2007-4203*, 2007.