

# A convergence study of solutions using two two-equation RANS turbulence models on a finite volume solver for structured grids

Jatinder Pal Singh Sandhu\*, Anant Girdhar†, Rakesh Ramakrishnan‡, R.D. Teja § and Santanu Ghosh¶  
*Indian Institute of Technology Madras, Chennai, Tamil Naidu, India, 600036*

**In this paper, we present a new, modular, explicit time marching based, three-dimensional finite volume solver for compressible flows on structured grids: FEST3D (Finite volume Explicit STructured 3 Dimensional solver). We provide details of the solver along with the verification performed to test the implementation of the code. Also, we present a study on the convergence of RANS solutions for two different, two equation turbulence models:  $k - \omega/k - \epsilon$  (Menter, F. R., “Two-equation eddy-viscosity turbulence models for engineering applications,” AIAA Journal, Vol. 32, No. 8, 1994, pp. 1598–1605.) and  $k - kL$  (Abdol-Hamid, K. S., Carlson, J.-R., and Rumsey, C. L., “Verification and Validation of the k-kL Turbulence Model in FUN3D and CFL3D Codes,” 46th AIAA Fluid Dynamics Conference, 2016.). Our study shows that for subsonic flow, which do not have discontinuities, using a limiter for higher order state reconstruction only for turbulence variables improves the overall convergence of the solution.**

## Nomenclature

$C_d$	=	coefficient of drag
$C_L$	=	coefficient of lift
$\hat{e}_x, \hat{e}_y, \hat{e}_z$	=	unit vectors in the direction of the x, y, and z axes of a three dimensional Cartesian coordinate system
$h$	=	grid size
$k$	=	turbulent kinetic energy of flow
$u, v, w$	=	Cartesian components of velocity vector
$p$	=	static pressure
$\Phi_\infty$	=	free-stream value of any state variable represented here as $\Phi$
$\gamma$	=	heat capacity ratio
$\delta_{ij}$	=	Kronecker delta
$\mu$	=	dynamic molecular viscosity
$\mu_t$	=	dynamic turbulent (eddy) viscosity
$\rho$	=	density of the fluid
$\forall$	=	volume of cell

## I. Introduction

**T**HIS paper presents a new finite-volume solver for 3-D compressible flows on structured grids, christened FEST-3D (Finite Volume Explicit STructured 3-Dimensional). The code solves the variable density, 3-D Navier-Stokes equations to simulate compressible laminar flow and Favre-averaged [1] Navier-Stokes for compressible turbulent flow. Turbulent flow simulations require the solution of additional Favre-averaged equations of turbulence variables, which depend on the turbulence model being used, added to the system of governing equations. The two-equation RANS turbulence models, SST (Shear-Stress Transport) [2] and k-kL [3], have been successfully integrated into FEST3D code.

---

\*Ph.D. student, Department of Aerospace Engineering, IIT Madras, Chennai, India.

†B.Tech student (2015), Department of Aerospace Engineering, IIT Madras, Chennai, India.

‡Dual degree student (2016), Department of Aerospace Engineering, IIT Madras, Chennai, India.

§B.Tech student (2016), Department of Aerospace Engineering, IIT Madras, Chennai, India.

¶Assistant Professor, Department of Aerospace Engineering, IIT Madras, Chennai, India, Member

Simple flows with an analytic solution, and verification results of CFL3D code [4], given on Langley Research Center's turbulence modeling resource website\*, were used to verify the implementation of various models in the code. Further, some additional verification cases were obtained from Higher order CFD workshop [5].

An important consideration for the solution of steady state simulations is the level of convergence of the residual. The simulation of low-speed flows with explicit time-marching techniques especially suffers from poor convergence, and the situation is worse when the flow is turbulent. In this work we investigate the convergence of the simulations for three different flows at low-to-medium speeds: Mach 0.2 laminar and turbulent flow past a flat plate, Mach 0.2 turbulent flow past a 3D bump, and Mach 0.5 turbulent flow past a Joukowski airfoil. The turbulent-flow simulations are computed with both the SST (shear-stress/transport) [2] and k-kL [3] models. Although the simulations presented are for subsonic flows that do not have discontinuities, it is observed that the use of limiters in the reconstruction of the turbulence variables only result in improved convergence.

The outline for the rest of the paper is as follows. Section II contains the detailed description of the governing equations used in the code. Numerical schemes used in the code for state reconstruction at interface, inviscid and viscous flux formulation, gradient calculation, time integration and local time step are detailed in section III. Section IV provides details of boundary and initial conditions that have been implemented in the code. Section V contains details of these verification cases and results obtained for these validation cases with FEST3D. Finally, the study on the convergence of the solver for turbulent flow simulations, with limiting of either all variables, or just the turbulent flow variables, are presented in Section V.

## II. Governing Equations

The 3D Favre-averaged Navier-Stokes equations, which constitute the governing equations for the solver, can be written in conservative differential form as,

$$\frac{\partial \tilde{q}_c}{\partial t} + \nabla \cdot (\vec{F}_I - \vec{F}_V) = 0 \quad (1)$$

where the vector of conservative variables  $\tilde{q}_c$ , inviscid flux vector  $\vec{F}_I$  and viscous flux vector  $\vec{F}_V$  are given by,

$$\tilde{q}_c = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E_t \end{pmatrix} \quad (2)$$

$$\vec{F}_I = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho u H_t \end{pmatrix} \hat{e}_x + \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho v H_t \end{pmatrix} \hat{e}_y + \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho w H_t \end{pmatrix} \hat{e}_z \quad (3)$$

---

\*Data available online at <https://turbmodels.larc.nasa.gov/index.html>

$$\begin{aligned}
\vec{F}_V = & \begin{pmatrix} 0 \\ t_{xx} - \tau_{xx} \\ t_{xy} - \tau_{xy} \\ t_{xz} - \tau_{xz} \\ u(t_{xx} + \tau_{xx}) + v(t_{xy} + \tau_{xy}) + w(t_{xz} + \tau_{xz}) - q_x - Q_x \end{pmatrix} \hat{e}_x \\
& + \begin{pmatrix} 0 \\ t_{xy} - \tau_{yx} \\ t_{yy} - \tau_{yy} \\ t_{yz} - \tau_{yz} \\ u(t_{yx} + \tau_{yx}) + v(t_{yy} + \tau_{yy}) + w(t_{yz} + \tau_{yz}) - q_y - Q_y \end{pmatrix} \hat{e}_y \\
& + \begin{pmatrix} 0 \\ t_{zx} - \tau_{zx} \\ t_{zy} - \tau_{zy} \\ t_{zz} - \tau_{zz} \\ u(t_{zx} + \tau_{zx}) + v(t_{zy} + \tau_{zy}) + w(t_{zz} + \tau_{zz}) - q_z - Q_z \end{pmatrix} \hat{e}_z
\end{aligned} \tag{4}$$

Here,  $E_t$  is the total internal energy per unit mass and is given as

$$E_t = c_v T + \frac{1}{2}(u^2 + v^2 + w^2) \tag{5}$$

and  $H_t$  is the specific total enthalpy and is given as

$$H_t = c_p T + \frac{1}{2}(u^2 + v^2 + w^2) \tag{6}$$

where  $c_v$  and  $c_p$  are the specific heats at constant volume and constant pressure respectively. The viscous shear stress  $t_{ij}$  is defined for a Newtonian fluid as

$$t_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \tag{7}$$

The dynamic molecular viscosity is determined using Sutherland's law that is given by

$$\frac{\mu}{\mu_{\text{ref}}} = \left( \frac{T}{T_{\text{ref}}} \right)^{\frac{3}{2}} \frac{T_{\infty} + S1}{T + S1} \tag{8}$$

where  $S1$  is the Sutherland's constant and is taken as  $110 \text{ K}$ . The default reference values are  $\mu_{\text{ref}} = 1.716 \times 10^{-5} \text{ N} \cdot \text{s}/\text{m}^2$  and  $T_{\text{ref}} = 273.16 \text{ K}$ . Although these reference values can be changed to achieve particular Reynolds number. The Reynolds stress  $\tau_{ij}$  is defined as

$$\tau_{ij} = \mu_t \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij} \tag{9}$$

The value of  $\mu_t$  depends on the particular turbulence model used. For turbulent flows,  $\mu_t$  is non-zero, and Eq. (1) represents the Reynolds-(Favre-)Averaged Navier-Stokes (RANS) equations. The primitive variables in this case are Favre averaged - density weighted and time-averaged.

The laminar heat flux is defined as,

$$q_i = -K \frac{\partial T}{\partial x_i} \tag{10}$$

where coefficient of thermal conductivity,  $K$ , is determined as

$$K = \frac{\mu c_p}{Pr} \tag{11}$$

where the Prandtl number,  $Pr$ , is taken as 0.72 for air. The turbulent heat flux is defined as,

$$Q_i = -\frac{c_p \mu_t}{Pr_t} \frac{\partial T}{\partial x_j} \quad (12)$$

where  $Pr_t$  is considered to be 0.9. The gas is assumed to be calorically perfect and hence  $c_p$  is a constant. The equations are closed with the help of the perfect gas equation

$$p = \rho RT \quad (13)$$

where  $R = \frac{R_U}{M}$  is the specific gas constant for air, and  $R_U$  is the universal gas constant that has a value of 8314.34 J/kg-K, and  $M$  the molecular mass of air, which is calculated as 28.768 kg/kmol.

## A. Turbulence models

Two turbulence models, SST [2] and k-kL [3], have been implemented in FEST3D. Both of these models add two extra equations to the system of governing equations. When neither of the models is selected, turbulence viscosity is explicitly set to zero, which converts the Reynolds averaged (or more correctly Favre averaged) form of the governing equation to Navier-Stokes equation suitable for laminar flows.

### 1. SST

In SST model, two more density-weighted equations of  $k$ , and dissipation rate of turbulent kinetic energy,  $\omega$ , are solved along with Reynolds Averaged Navier-Stokes equations. In the current state of the FEST3D solver, SST model has been implemented according to [2]. However, to save computational effort and avoid numerical difficulties, the production term  $P$  is dealt differently as given in [6].

$$P = \mu_t \Omega^2 - \frac{2}{3} \rho k \delta_{ij} \frac{\partial u_i}{\partial x_j} \quad (14)$$

Where vorticity magnitude is given as

$$\Omega = \sqrt{2W_{ij}W_{ij}} \quad (15)$$

and

$$W_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) \quad (16)$$

is the anti-symmetric part of the velocity gradient tensor.

The  $\mu_t$  for SST is given as

$$\mu_t = \frac{\rho a_1 k}{\max(a_1 \omega, \Omega F_2)} \quad (17)$$

where  $a_1 = 0.32$  and  $F_2$  is blending function which is defined in in [2]. Second term in Production Eq. (14) is small for low speed subsonic flow and is neglected all the turbulent flow simulation presented in this paper.

### 2. k-kL

The  $k - kL$  model also adds two density-weighted equations of  $k$ , and product of turbulent kinetic energy  $k$  and integral length scale  $L$ , symbolized as  $kL$ , to the Reynolds Averaged Navier-Stokes equations. We have referred to [3] for implementing the model in FEST3D code. The turbulent viscosity for k-kL model is given as

$$\mu_t = C_\mu^{1/4} \frac{\rho(kL)}{k^{1/2}} \quad (18)$$

Where  $c_\mu = 0.09$

### III. Numerical Scheme

The semi-discrete equations obtained on integrating Eq. (1) over a control-volume (hexahedral element) is given by

$$\left(\frac{\partial \tilde{q}_c}{\partial t}\right) \forall + \sum_{k=1}^6 \left[ \left( \vec{F}_I - \vec{F}_V \right)_k \cdot \hat{n}_k \right] A_k = 0 \quad (19)$$

where  $k = 1$  to 6 denotes the 6 faces of the hexahedral element, and  $\hat{n}_k$  is the unit outward normal of any face. Here  $\forall$  denotes the volume of the cell and  $A_k$  denotes the area of the  $k^{th}$  face.

The finite volume formulation for a cell  $(i, j, k)$  as such can be compactly expressed as:

$$\left(\frac{\partial \tilde{q}_c}{\partial t}\right)_{i,j,k} \forall_{i,j,k} + \tilde{\mathbb{R}}_{i,j,k} = 0 \quad (20)$$

#### A. Inviscid flux formulation

In the solver, five flux reconstruction schemes are built in - Van Leer [7], AUSM [8] (Advection Upstream Splitting Method), LDFSS [9] (Low-Diffusion Flux-Splitting Scheme), SLAU [10] (Simple Low-Dissipation Advection Upwind) and HLLE<sup>†</sup> [11] (Harten-Lax-van Leer-Einfeldt). The van Leer scheme is included for instructional purposes only.

#### B. Viscous flux formulation

The computation of the viscous fluxes involve three steps:

- Computation of the cell-averaged gradients of velocity, temperature and turbulence variable using the Green-Gauss technique
- Reconstructing the gradients at the faces of each cell
- Using the face-reconstructed gradients to compute the viscous momentum and energy fluxes

The first two steps listed above are explained in some detail next.

##### 1. Cell-averaged gradients - Green Gauss technique

The Green-Gauss technique uses Gauss's theorem to estimate cell-averaged gradients of velocity (having three scalar components) and temperature of each cell as shown here,

$$\left(\vec{\nabla} W\right) \Delta \forall = \iiint_{\forall} \vec{\nabla} W d\forall = \oint_{\Omega} W \hat{n} dS = \sum_{k=1}^6 W_k \hat{n}_k A_k \quad (21)$$

that gives,

$$\left(\vec{\nabla} W\right) = \frac{1}{\Delta \forall} \sum_{k=1}^6 W_k \hat{n}_k A_k \quad (22)$$

where in Eq. (22):

- $W \in \{u, v, w, T, \text{Turbulence Variables}\};$
- $\left(\vec{\nabla} W\right)$  is the cell averaged value of the gradient;
- $W_k$  is the reconstructed average value of at the face  $k$ . To be more clear, for face  $i + \frac{1}{2}$ ,  $W_{i+\frac{1}{2}} = \frac{1}{2} (W_i + W_{i+1})$ ;
- $\hat{n}_k$  is the outward normal of face  $k$ .

---

<sup>†</sup> HLLE scheme is only available in 2D version of the code. 3D implementation is pending.

## 2. Gradients reconstructed at the faces

To compute the viscous fluxes, the value of gradients at the faces is required. A simple method to estimate (the face gradient) is to compute the average of the gradients at the neighboring cells of a face as shown in Eq. (23).

$$\overline{\vec{\nabla}W}_{i+\frac{1}{2}} = \frac{1}{2} \left[ \left( \vec{\nabla}W_i \right) + \left( \vec{\nabla}W_{i+1} \right) \right] \quad (23)$$

But this leads to an odd-even decoupling problem [12]. Hence, the following correction is made

$$\vec{\nabla}W_{i+\frac{1}{2}} = \overline{\vec{\nabla}W}_{i+\frac{1}{2}} + \left( \frac{W_{i+1} - W_i}{|\vec{r}_{i+1} - \vec{r}_i|} - \overline{\vec{\nabla}W}_{i+\frac{1}{2}} \cdot \hat{r}_{ij} \right) \hat{r}_{ij} \quad (24)$$

where,

$$\hat{r}_{ij} = \frac{\vec{r}_{i+1} - \vec{r}_i}{|\vec{r}_{i+1} - \vec{r}_i|}$$

Equation (24) describes the gradients calculated at the faces, where  $\vec{r}_i$  is the position vector of the  $i^{th}$  cell centre. It should be noted that at the boundary faces, calculation of  $\hat{r}_{ij}$  requires information about the position vector of the cell-center at the ghost cells. The ghost cell centroid is computed assuming that it is a mirror image, about the shared boundary face, of the cell centroid of its (adjacent) interior neighbor.

## C. State Reconstruction

Higher order interface state reconstruction is done using WENO (Weighted Essentially Non-Oscillatory) [13], PPM (Piece-wise Parabolic Method, [14]) or MUSCL (Monotonic Upwind Schemes for Scalar Conservation Laws, [15]). There are many limiters that can be used in conjunction with the MUSCL  $\kappa$  scheme. In this code, the symmetric *Koren* limiter [16] is used which is described as follows:

$$\begin{aligned} \Psi(r) &= \max \left( 0, \min \left( 2r, \frac{2+r}{3}, 2 \right) \right) \\ \lim_{r \rightarrow \infty} \Psi(r) &= 2 \end{aligned} \quad (25)$$

where  $r$  is the ratio of the adjacent slopes.

## D. Time integration

A first order, explicit time marching scheme used in the solver for steady state problems. For unsteady problems, RK-4 time integration [17] is used to get higher order time accuracy of the solution. A local time-step is used for steady-state problems to accelerate convergence. For better numerical stability, destruction terms in the (additional) equations for the turbulence quantities are treated using a point-implicit approach.

## E. Local time step

At each cell center  $i$ , local time step is calculated as:

$$\Delta t_i = \frac{CFL \times \forall_i}{\sum_k \left[ (|\vec{U}_k \cdot \hat{n}_k| + c_k) + \frac{2\gamma}{\rho(\vec{r}_j - \vec{r}_i) \cdot \vec{n}_k} \left( \frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) \right] A_k} \quad (26)$$

where index  $j$  represents the cell which is immediate neighbour to cell  $i$ , sharing the face  $k$ ,  $\vec{r}$  is position vector of the cell center.  $\vec{U}_k$ ,  $c_k$ ,  $A_k$  and  $\hat{n}_k$  are the velocity vector, speed of sound, area magnitude, and unit normal, respectively, at interface of cell  $i$  and  $j$  (face  $k$ ). For time-accurate simulation, either a constant time step is used, or the global minimum of the local time step of all the cell is selected as the global time step.

## F. Normalized Residual and Convergence Monitoring

The L2 norm of the residual  $\tilde{\mathbf{R}}$  of all cells is computed for convergence check.

$$\|\mathbf{R}\| = \left( \sum_{i,j,k} \left[ \left( \frac{\mathbf{R}_1}{\rho_\infty u_\infty} \right)^2 + \left( \frac{\mathbf{R}_2}{\rho_\infty u_\infty^2} \right)^2 + \left( \frac{\mathbf{R}_3}{\rho_\infty u_\infty^2} \right)^2 + \left( \frac{\mathbf{R}_4}{\rho_\infty u_\infty^2} \right)^2 + \left( \frac{\mathbf{R}_5}{\rho_\infty u_\infty \left( \frac{u_\infty^2}{2} + \frac{\gamma}{\gamma-1} \frac{p_\infty}{\rho_\infty} \right)} \right)^2 \right] \right)^{\frac{1}{2}} \quad (27)$$

where  $\mathbf{R}_1$ ,  $\mathbf{R}_2$ ,  $\mathbf{R}_3$ ,  $\mathbf{R}_4$ ,  $\mathbf{R}_5$  are the mass residual, momentum residual (3 components), and energy residual. In the above equation, the residual for each of the equations is normalized by a free-stream quantity having the same dimension. For the convergence studies presented later, the (normalized) residuals for the transport equations of the turbulence variables are also monitored separately.

## IV. Boundary and Initial Conditions

In the solver, which is finite volume based, boundary conditions are implemented by specifying flow properties in the ghost cells. No-slip and no-penetration velocity boundary conditions are used at no-slip / viscous wall, and only a no-penetration condition is used at a slip wall. The no-penetration boundary condition is enforced by setting the convective fluxes to zero, and for the no-slip wall, the velocity vector in the ghost cell is made opposite to that of the immediate interior cell. In both cases, density and pressure values at ghost cells are copied from the interior cells.

For the supersonic inflow boundary condition, all primitive variables are specified at the ghost cells, and for the supersonic outflow case, all primitive variables are extrapolated from the interior using first order accurate approximations at present. For subsonic inflow, velocity and density are fixed, and pressure is extrapolated, whereas for subsonic outflow, the pressure is fixed, and density and velocity are extrapolated. For total inlet boundary condition, velocity and speed of sound at a boundary face are found using Riemann invariant, and pressure at a boundary face is based on prescribed total pressure, and after that density is calculated using state relation ([4]).

For far-field boundary condition, velocity and speed of sound at a boundary face are found using Riemann invariants, and pressure and density are calculated using isentropic relations ([18]). The fixed Riemann invariant  $R_\infty$ , based on free-stream conditions, and the variable Riemann invariant  $R_e$ , based upon the interior cell closest to boundary face, are given as

$$R_\infty = \tilde{\mathbf{U}}_\infty \cdot \hat{\mathbf{n}} - \frac{2c_\infty}{\gamma - 1} \quad (28)$$

$$R_e = \tilde{\mathbf{U}}_e \cdot \hat{\mathbf{n}} + \frac{2c_e}{\gamma - 1} \quad (29)$$

where  $\tilde{\mathbf{U}}$  is the velocity vector,  $\hat{\mathbf{n}}$  is the unit outward face normal and  $c$  is the speed of sound. Face normal velocity and speed of sound at the boundary face are obtained using  $R_\infty$  and  $R_e$  as

$$\tilde{\mathbf{U}}_{face} \cdot \hat{\mathbf{n}} = \frac{1}{2} (R_\infty + R_e) \quad (30)$$

$$c_{face} = \frac{\gamma - 1}{4} (R_\infty - R_e) \quad (31)$$

Based on the direction of face normal velocity (pointing into or out of domain), the tangential velocity at a face is obtained from either the free-stream conditions (pointing into) or data at the adjacent interior cell (pointing out). Using the resultant velocity at face  $\tilde{\mathbf{U}}_{face}$ , speed of sound at face  $c_{face}$ , and isentropic relations, pressure and density at a face are calculated.

$$\tilde{\mathbf{U}}_{face} = \tilde{\mathbf{U}}_\infty + (\tilde{\mathbf{U}}_{face} \cdot \hat{\mathbf{n}} - \tilde{\mathbf{U}}_\infty \cdot \hat{\mathbf{n}}) \hat{\mathbf{n}} \quad (32)$$

$$s_{face} = \frac{p_\infty}{\rho_\infty^\gamma} \quad (33)$$

or

$$\vec{U}_{face} = \vec{U}_e + (\vec{U}_{face} \cdot \hat{n} - \vec{U}_e \cdot \hat{n}) \hat{n} \quad (34)$$

$$s_{face} = \frac{p_e}{\rho_e^\gamma} \quad (35)$$

and

$$\rho_{face} = \left[ \frac{(c_{face})^2}{\gamma s_{face}} \right]^{\frac{1}{\gamma-1}} \quad (36)$$

$$p_{face} = \frac{\rho_{face} (c_{face})^2}{\gamma} \quad (37)$$

FEST3D is distributed-memory based multiblock code, wherein the computational domain is divided in to multiple hexahedral blocks that communicate with each other using MPI messages. A one-to-one mapping methodology is adopted for the code, wherein, a face of any block can be shared with only one other block. An interface boundary condition is used for such faces, which are essentially internal block boundaries generated due to the domain decomposition. Geometry-related data and state variables of three layers of cells (adjacent to the interface and on either side of it) are exchanged at such boundaries using MPICH subroutines.

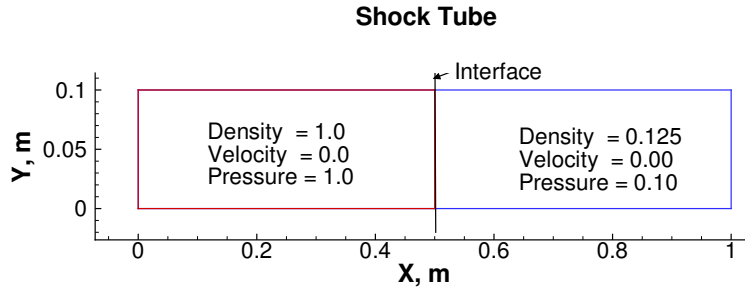
By default, the domain is initialized with free-stream conditions for the simulations. For special initialization requirement, a new portion of code is hardwired in the FEST3D solver. A restart mechanism is also programmed which allows the solution to be restarted using a previous data dump.

## V. Results

### A. Verification

#### 1. Shock Tube

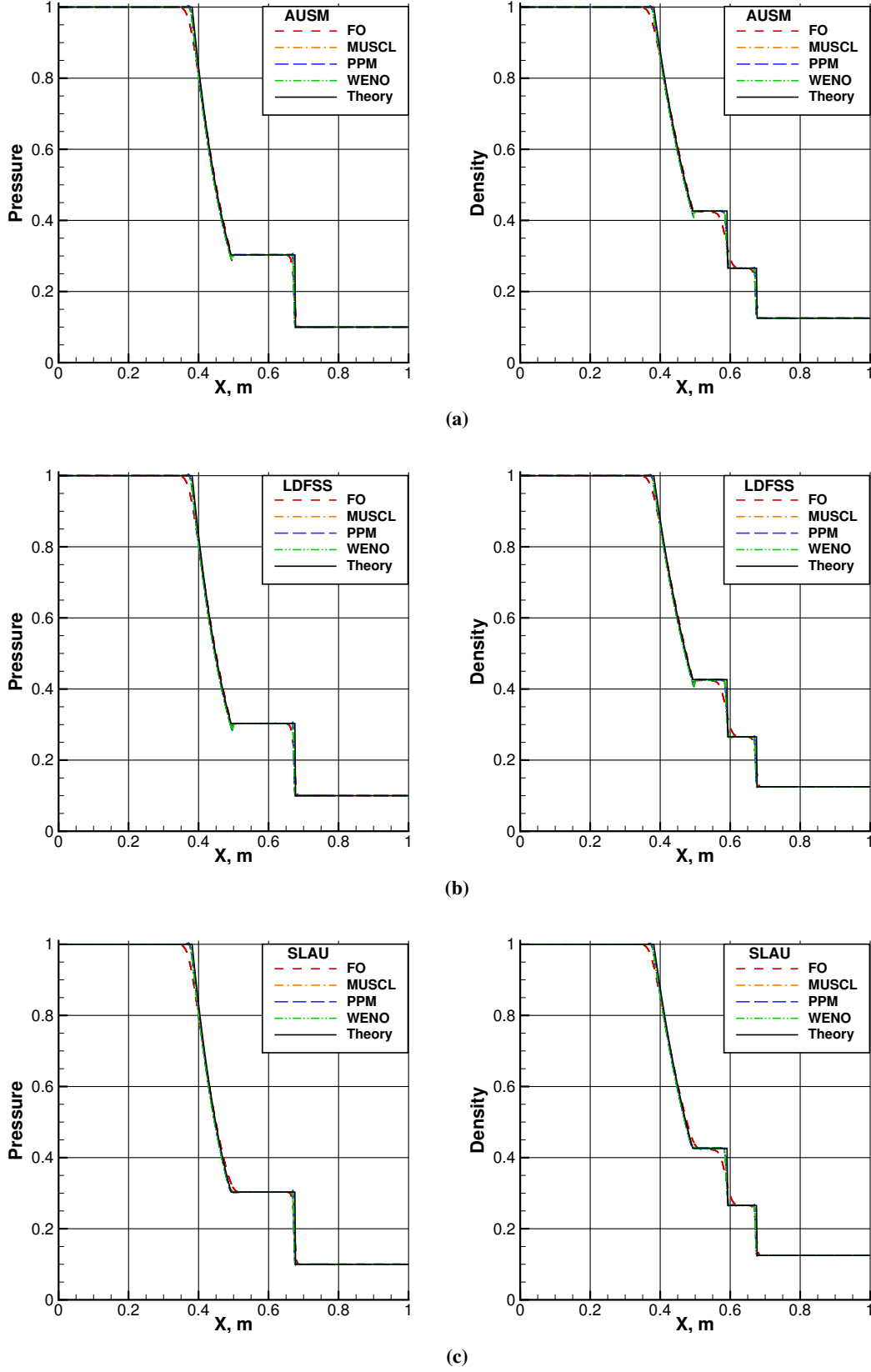
Shock tube is a primary test case for a finite volume Riemann solver. It is one of the few problems in fluid mechanics for which analytic solution is available. Domain of computation and initial state used for shock-tube [19] test case are defined in the Fig. 1. All the physical boundaries of the domain were treated as slip wall.



**Fig. 1 Domain and initial state used for shock tube test case.**

A uniform structured grid of size  $512 \times 2 \times 2$  was used for the test case. Flux reconstruction schemes AUSM, LDFSS, and SLAU were verified with all face reconstruction methods implemented in FEST3D: First Order (FO), MUSCL ( $\kappa = 1/3$ ), PPM, WENO. The computation was time marched with a global time step of  $10^{-4}$  s for 1000 iterations with 4<sup>th</sup> order Range-Kutta method. Solution calculated with FEST3D was evolved from 0 seconds to 0.1 seconds and compared with the analytic solution at 0.1 seconds. The FORTRAN program written by Bruce Fryxell [20] was used to find the analytic solution to the problem.





**Fig. 2** Pressure (left column) and density (right column) for shock tube test case at 0.1 seconds for a) AUSM Scheme; b) LDFSS scheme; c) SLAU scheme with different face reconstruction scheme implemented in FEST3D.

Figure 2 shows the comparison of pressure and density for the different face reconstruction methods and flux schemes implemented in FEST3D. Although higher order face reconstruction method performs better than first order face reconstruction method in resolving the discontinuities, they produce more oscillation in the solution. For WENO scheme, the oscillations observed can be an outcome of not using TVD Runge-Kutta method for time integration. A small overshoot and relatively large undershoot is visible at either end of expansion fan for all higher order methods when AUSM and LDFSS are used for flux reconstruction; however the same is not present when SLAU is used as flux reconstruction method.

## 2. Inviscid Flow Through a Channel with a Smooth Bump

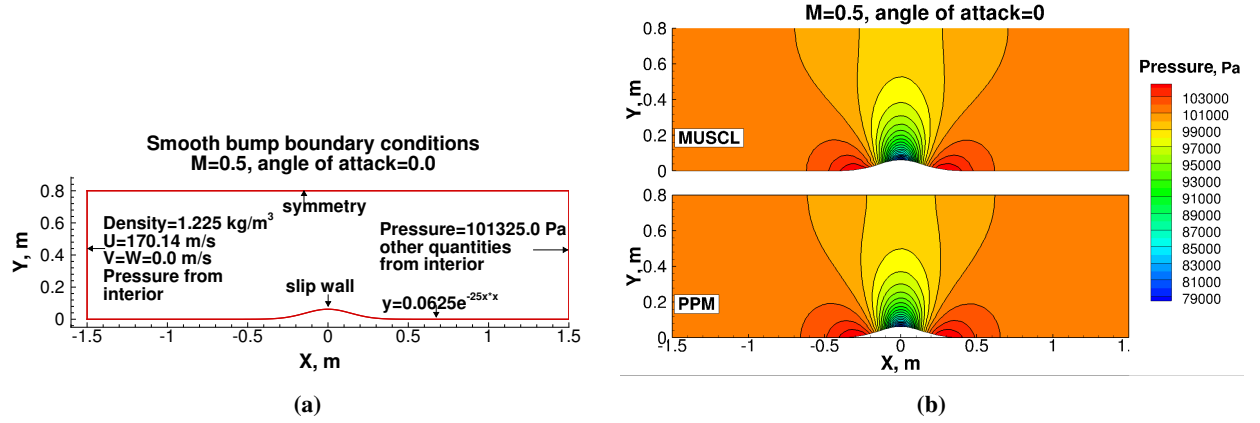


Fig. 3 Inviscid flow over smooth bump: a) Domain and boundary conditions used; b) pressure contour for different face reconstruction scheme used.

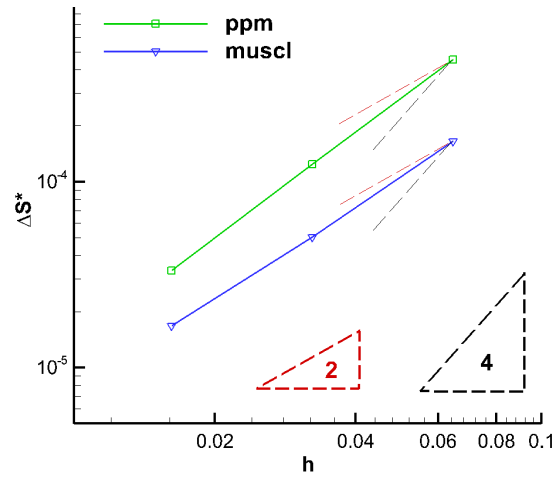


Fig. 4 For inviscid flow over smooth bump: entropy error vs grid size

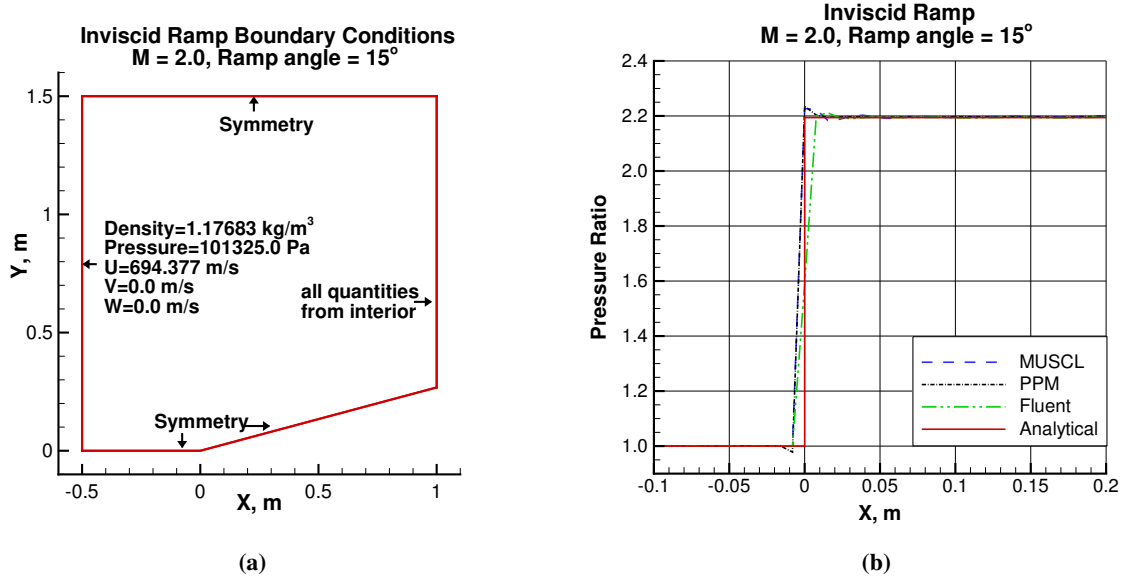
The subsonic Mach 0.5 flow over a smooth bump with zero degree angle of attack is simulated. This verification case was obtained from [5]. The domain used, and boundary conditions applied to this domain are illustrated in Fig. 3a. The code was run with AUSM scheme for flux formulation. Two different higher order state reconstruction schemes, MUSCL ( $\kappa = 1/3$ ), and PPM, were tested using this verification case. A 4<sup>th</sup> order Runge-Kutta time integration scheme was used to reach steady state. Pressure contours obtained for MUSCL and PPM on the finest, uniform, structured grid  $97 \times 49 \times 2$  are shown in Fig. 3b. There is no exact solution available, but the geometry is smooth, and so is the flow. As such, the entropy of the flow should not change. Based on this reasoning, change in the entropy,  $\Delta s^*$ , relative to free

stream value  $s_\infty$ , was taken as the measure of error in the solution obtained from the FEST3D code. The equation used for calculation of  $\Delta s^*$  is given by

$$\Delta s^* = \sqrt{\frac{\int_{\Omega} \left( \frac{p}{p_\infty} \left( \frac{\rho_\infty}{\rho} \right)^\gamma - 1 \right)^2 dV}{s_\infty^2 V_{total}}} \quad (38)$$

Figure 4 shows the relative entropy error in the solution for three different grids:  $25 \times 13$ ,  $49 \times 25$  and  $97 \times 49$ . As expected, the solution with interface state reconstruction using PPM has a higher order of convergence than the solution with the MUSCL scheme.

### 3. Supersonic inviscid flow past a ramp.



**Fig. 5 Supersonic flow over inviscid ramp: a) domain and boundary condition used; b) pressure ratio compared with theory and fluent simulation.**

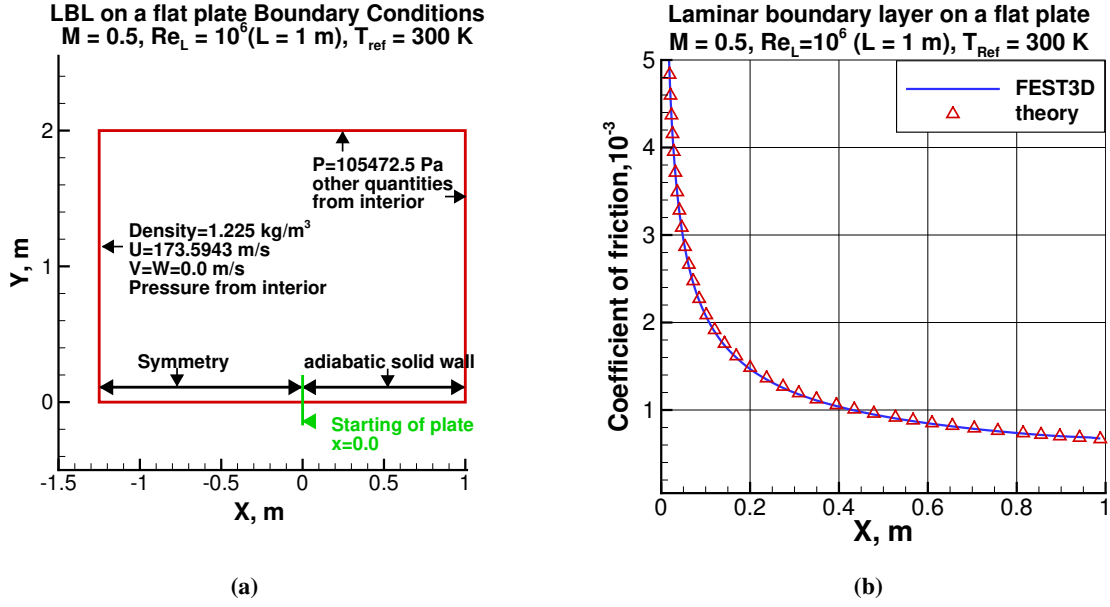
The supersonic inviscid flow of Mach 2.0 past a 15-degree ramp has been simulated. The domain used, and boundary condition applied to the domain are illustrated in Fig. 5a. The case definition and grid used were obtained from NPARC Alliance Validation Archive<sup>‡</sup>. A uniform grid of size  $153 \times 91 \times 2$  was used for simulation. The AUSM scheme was used for flux formulation and both PPM and MUSCL ( $\kappa = 1/3$ ) face reconstruction schemes are tested for this verification case. The first-order time integration was used to integrate the solution to steady state. To make sure steady state is achieved, normalized residual (summed) of continuity, momentum and, energy equation, as defined in Eq. (27), was typically driven below  $10^{-12}$ . Pressure ratio ( $p/p_\infty$ ) from the simulations on FEST3D, extracted near the surface of the ramp, are compared with results obtained from Fluent simulation and exact solution as shown in Fig. 5b. The Fluent simulation was run using a higher order face reconstruction option. Pressure ratio across shock matches well with the exact solution as well as the Fluent result. Both, PPM and MUSCL higher order schemes show oscillations after the shock, but PPM exhibits an undershoot before the shock also.

### 4. Laminar boundary layer (LBL) on a flat plate.

A Mach 0.5 flow past a flat plate having Reynolds number 1 million (based on plate length of 1 meter) has been simulated. The domain used and boundary condition applied is shown in Fig. 6a. The case definition and grid were obtained from website on 4th International Workshop on High-Order CFD Methods<sup>§</sup>. The simulation was run using AUSM scheme for

<sup>‡</sup>Data available online at <https://www.grc.nasa.gov/WWW/wind/valid/validation.html>

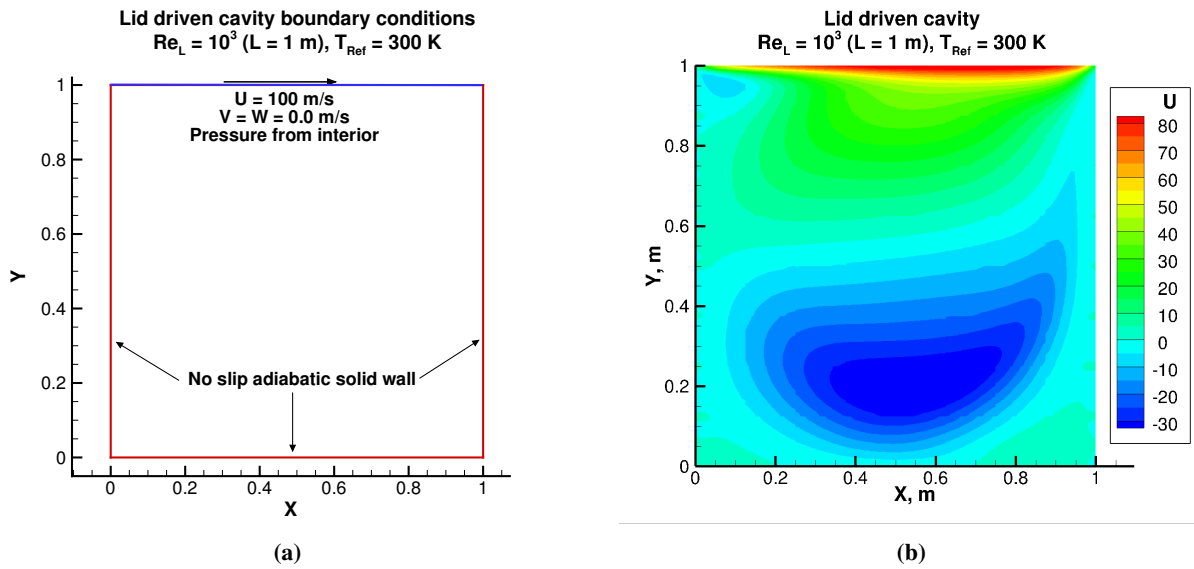
<sup>§</sup>Data available online at <https://www.grc.nasa.gov/hio CFD/>



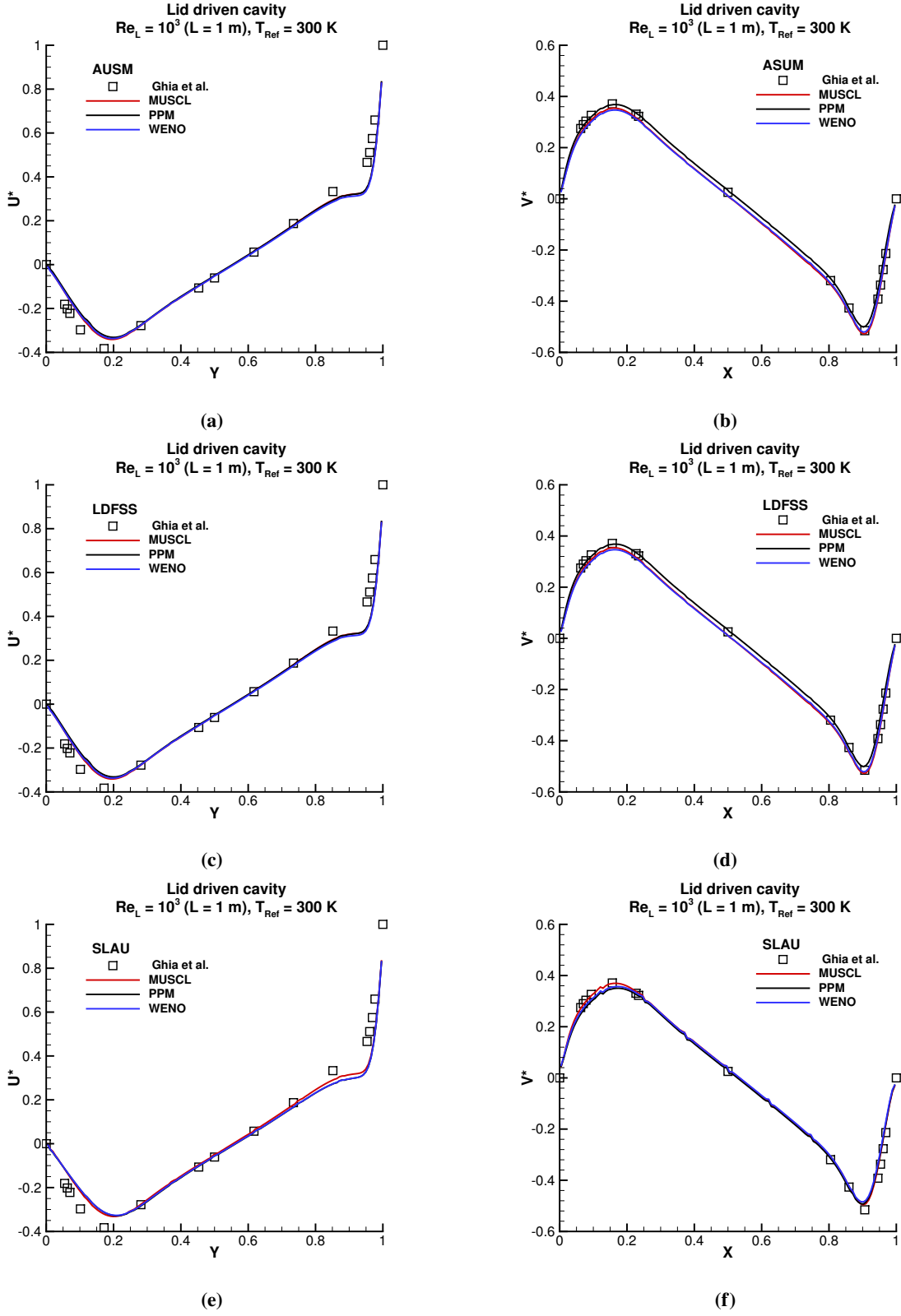
**Fig. 6 Laminar boundary layer simulation: a) Domain and boundary condition used; b) coefficient of friction compared with theory.**

inviscid flux formulation, MUSCL ( $\kappa = 1/3$ ) as state reconstruction scheme, and 4<sup>th</sup> order Runge-Kutta time integration method. The code was run till scaled residual of continuity equation reaches the limit of  $10^{-15}$ . Residual history is shown in Fig. 14a. For the finest mesh, which has 225x161 grid points and 145 grid points on the plate, the coefficient of drag  $C_d$  obtained with FEST3D code ( $1.326 \times 10^{-3}$ ) matches well with the theoretical value of  $1.328 \times 10^{-3}$ . In Fig. 6b, coefficient of friction  $C_f$  along the length of plate obtained with FEST3D solver for this case shows a good match with  $C_f$  obtained with the theoretical formula  $\frac{0.664}{\sqrt{Re_x}}$  for incompressible flows.

##### 5. Lid driven cavity



**Fig. 7 Lid-driven cavity flow: a) domain and boundary conditions, and b) contours of  $u$  velocity.**

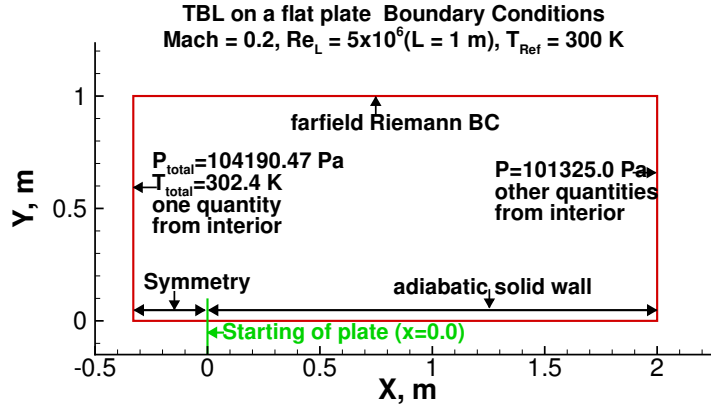


**Fig. 8** Variation in a) pressure and b) density for AUSM Scheme; c) pressure and d) density for LDFSS scheme and e) pressure and f) density for SLAU scheme for different face reconstruction scheme used.

A laminar flow in a square cavity is a very standard verification test case for an incompressible flow solver. This test case is attempted with the FEST3D code, which solves the compressible Navier-Stokes equations. As it was observed with FEST3D that it is not possible to get good convergence with the original boundary condition given in [21], a modified set of boundary conditions, as shown in Fig. 7a were used for the simulation. For comparison with work done by Ghia et al [21], results obtained with FEST3D were normalized with the lid velocity, and are shown in Fig. 8.

A uniform structured grid of 129x129x2 grid points was used for the computation. All the flux reconstruction schemes: AUSM, LDFSS, and SLAU, and all the higher order method for reconstruction of state at face: MUSCL, PPM, and WENO, implemented in FEST3D were used. Solution was advanced in time with RK4 method for a total of  $10^5$  iteration; which drove normalized residual (summed) of continuity, momentum and energy equations, as given in Eq. (27) below  $10^{-3}$ . Figure 8 show the comparison between FEST3D computation and those reported by Ghia et al. [21]. Figures 8a, 8c and 8e compares normalized x-component of velocity ( $U^*$ ) at the vertical geometric mid-plane, for different inviscid flux reconstruction schemes with [21]. A similar comparison is done for normalized y-component of velocity ( $V^*$ ) at horizontal geometric mid-plane as shown in Figs. 8b, 8d and 8f. Although all the schemes implemented in FEST3D gives similar results, a small mismatch is clearly visible between results from Ghia et al. [21] and the values computed with FEST3D.

#### 6. Turbulent boundary layer (TBL) on a flat plate.



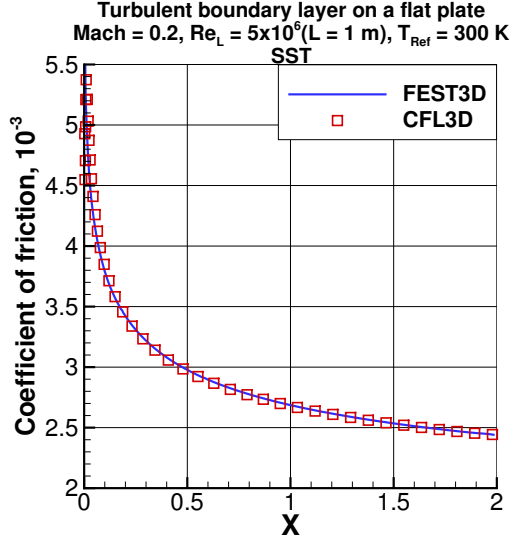
**Fig. 9 Domain and boundary conditions used for turbulent boundary layer simulation: subsonic flow over flat plate.**

The subsonic flow of Mach 0.2 and Reynolds number of 5 million, based on unit length of the plate, over a flat plate of 2 meters (with an inviscid region in front of leading edge of the plate) is simulated. The domain used and boundary conditions applied is shown in Fig. 9. The case definition and grid were obtained from Langley Research Center's turbulence modeling resource website<sup>¶</sup>. The simulation was run using AUSM scheme for inviscid flux formulation scheme, MUSCL ( $\kappa = 1/3$ ) as state reconstruction scheme, and 4<sup>th</sup> order Runge-Kutta time integration method.

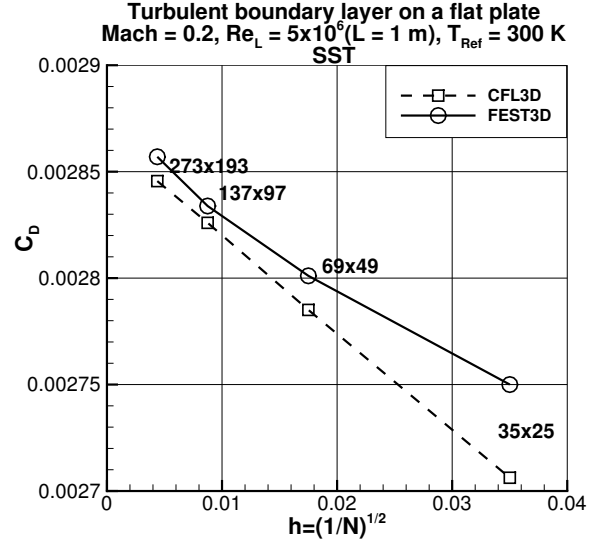
Both the SST and k-kL turbulence models provided in FEST3D solver were tested for this simple verification case. The code was run for a maximum of 2 million iterations on the finest grid of 273x193 (225 grid points on the plate), which gave a normalized density residual under  $10^{-11}$ . The very high number of time steps are needed for the simulation as the solver employs explicit time marching and the Mach number of the flow is low. Results obtained with FEST3D code for this case were compared with results given for CFL3D code on Langley Research Center's turbulence modeling resource website<sup>¶</sup> and in NASA technical report [22]. Figures 10a and 10b show that there is a good match between the FEST3D code and CFL3D code for coefficient of friction and drag obtained with the SST turbulence model. A similar conclusion can be inferred from Fig. 11a and 11b for the k-kL turbulence model. The results shown in figures 10a and 11a are for the finest mesh.

<sup>¶</sup>Data available online at <https://turbmodels.larc.nasa.gov/flatplate.html>

<sup>¶</sup>Data available online at [https://turbmodels.larc.nasa.gov/flatplate\\_sst.html](https://turbmodels.larc.nasa.gov/flatplate_sst.html) (SST) and [https://turbmodels.larc.nasa.gov/flatplate\\_kkl-meah.html](https://turbmodels.larc.nasa.gov/flatplate_kkl-meah.html) (k-kL)

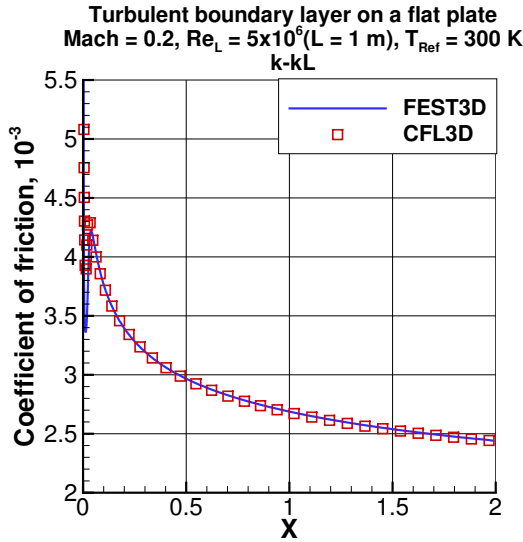


(a)

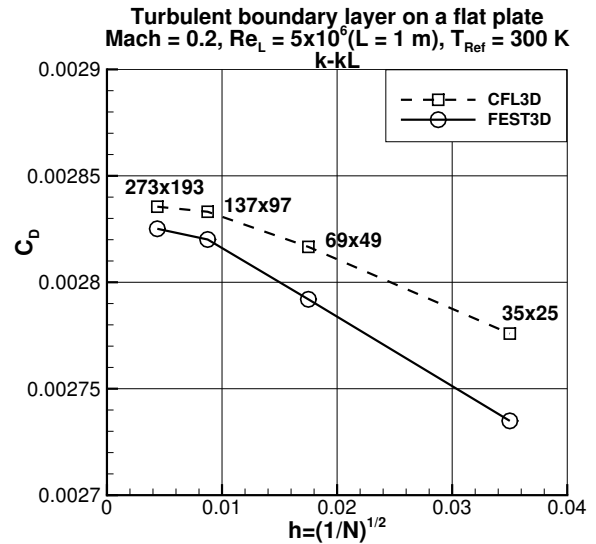


(b)

**Fig. 10** Results obtained for turbulent boundary layer simulation with SST turbulence model: a) coefficient of friction; b) grid convergence



(a)



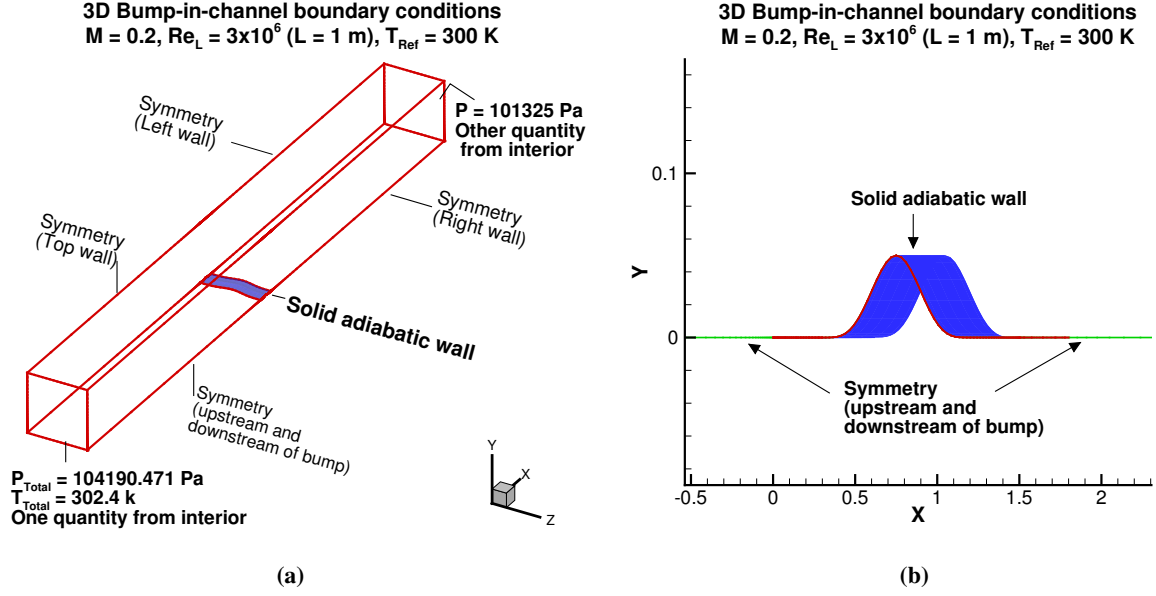
(b)

**Fig. 11** Results obtained for turbulent boundary layer simulation with k-kL turbulence model: a) coefficient of friction; b) grid convergence

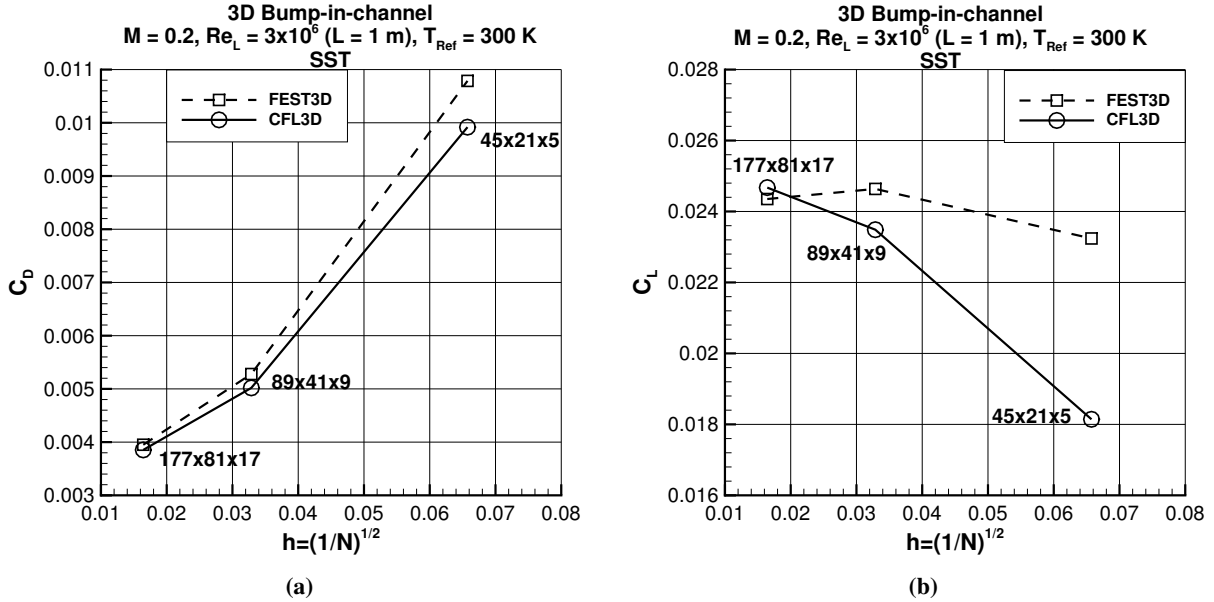
### 7. 3D bump in Channel

The subsonic flow of Mach 0.2 and Reynolds number of 3 million, based on unit length of the plate, over a smooth 3D bump in a channel with an inviscid region upstream and downstream of the bump is simulated. The domain used and boundary condition applied is shown in Fig. 12. The case definition and grid were obtained from Langley Research Center's turbulence modeling resource website<sup>\*\*</sup>. The simulation was run using AUSM scheme for inviscid flux formulation scheme, MUSCL ( $\kappa = 1/3$ ) as state reconstruction scheme, 4<sup>th</sup> order Runge-Kutta time integration method

<sup>\*\*</sup>Data available online at <https://turbmodels.larc.nasa.gov/bump3d.html>



**Fig. 12** Domain and boundary conditions used 3D bump-in-channel test case a) three dimensional view and b) closeup of 3D bump.



**Fig. 13** Grid convergence study for 3D bump-in-channel with a) coefficient of drag and b) coefficient of lift.

and SST turbulence model. Three different grid levels were used for the study: I) 45x21x5, II) 89x41x9, III) 177x81x17 as shown in Fig 13a

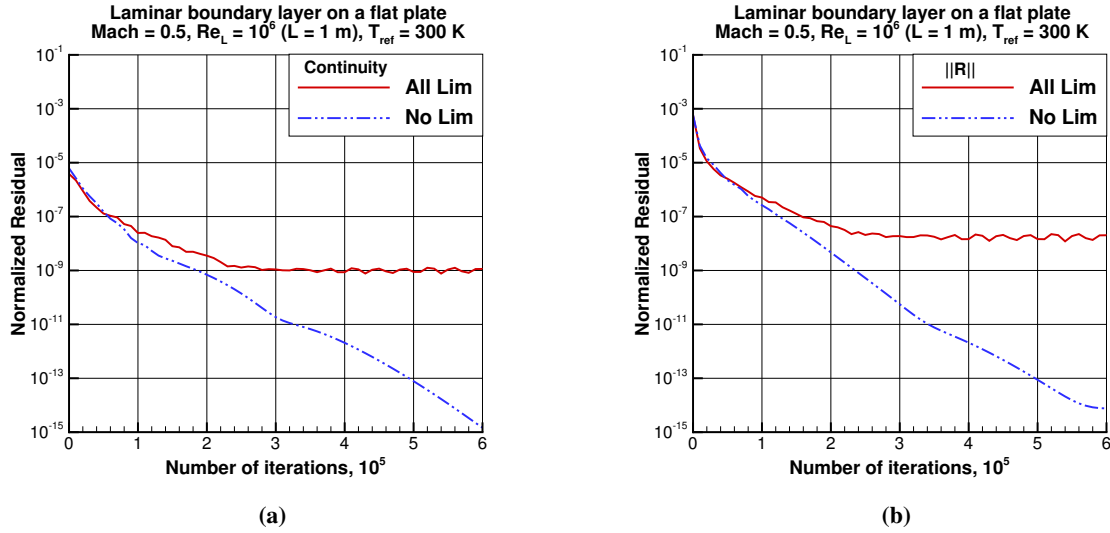
Similar to turbulent flat plate case, results obtained with FEST3D code, for this case, were compared with results given for CFL3D code on Langley Research Center's turbulence modeling resource website<sup>††</sup> and in NASA technical report [22]. Figures 13a and 13b shows that, for the fine grid, there is a good match between the coefficient of drag and lift, respectively, obtained with FEST3D code, and the same quantities obtained with CFL3D code for SST turbulence model.

<sup>††</sup>Data available online at <https://turbmodels.larc.nasa.gov/bump3d-sstv.html>



## B. Convergence study

This is a study on the effect of limiter on the convergence of solution to steady state, for laminar and turbulent flow simulations, is performed with MUSCL [15] face reconstruction scheme, AUSM flux formulation scheme, and explicit Euler time integration. The basic idea behind the use of the limiter is to prevent oscillations: undershoots and overshoots, and maintain monotonic behavior within the cell when using higher order state reconstruction [23]. However, application of these limiters to smooth flows (without shock), like subsonic laminar flow past a flat plate, described in section V.A.3, limits the convergence of the solution. This effect can be seen clearly in Figs. 14a and 14b, where application of the limiter stalled the convergence of the solution. The simulation was run on a fine grid of 225x161 grid points. In Figs. 14a and 14b, “All Lim” means that limiter was used for all the state variables and “No Lim” means the limiter was not used for any of the state variables.



**Fig. 14** Solution convergence for laminar boundary layer simulation: a) normalized residual for continuity equation; b) normalized residual (summed) of continuity, momentum, and energy equation

A similar study was conducted for the turbulent boundary-layer simulation, described in section V.A.4, which is also a smooth flow without any discontinuity (shock). The simulation was performed for three different grids, having  $69 \times 49$ ,  $137 \times 97$  and  $293 \times 193$  grid points, for both SST and  $k - kL$  turbulence models. Results similar to that of the laminar boundary layer simulation were expected: stalled convergence with limiter and good convergence without any limiter. In addition to the cases of ‘All Lim’ and ‘No Lim’, a third variant (of limiting) was considered for the turbulent boundary-layer simulations, wherein only the reconstruction of the turbulence variables is limited - this is termed as ‘Partial Limiting’. Results for the  $69 \times 49$  and  $293 \times 193$  are shown here in Figs. 15 and 16., for the simulations with SST and  $k - kL$  turbulence models respectively, which throws up something interesting.

Firstly, although the solution without the use of limiters (‘No Lim’) for the simulation with the SST turbulence model on the coarse grid ( $69 \times 49$ ) showed that the residual for the Navier-Stokes equations converge to low values, however, it also showed a stalled convergence for the equations for the turbulence variables,  $k$  and  $\omega$ , as shown in Fig. 15. It can be further observed from Figs. 15b and 15c that although the residuals for the turbulence equations on the fine mesh ( $273 \times 193$ ) without the use of limiters (‘No Lim’) also stall, it stalls at a lower value as compared to the coarse mesh solution.

The addition of limiting for all the variables (‘All Lim’) improves the convergence of the residuals for the turbulence equations, but degrades the convergence of the Navier-Stokes equations for both the coarse- and fine- grid simulations.

The coarse and fine grid simulations with the  $k - kL$  turbulence model (Fig. 16) showed behavior similar to SST turbulence model. As seen in Figs. 16b and 16c, stalled convergence for the turbulence variables’ ( $k$  and  $kL$ ) equations, for ‘No Lim’ case improves (but still remains stalled) with the application of limiter to all the variables (‘All Lim’). For both turbulence models,  $k - kL$  and SST, the residual for the Navier-Stokes equations with ‘No Lim’ case stalls at value lower than that for ‘All Lim’, as seen in Figs. 15 and 16.

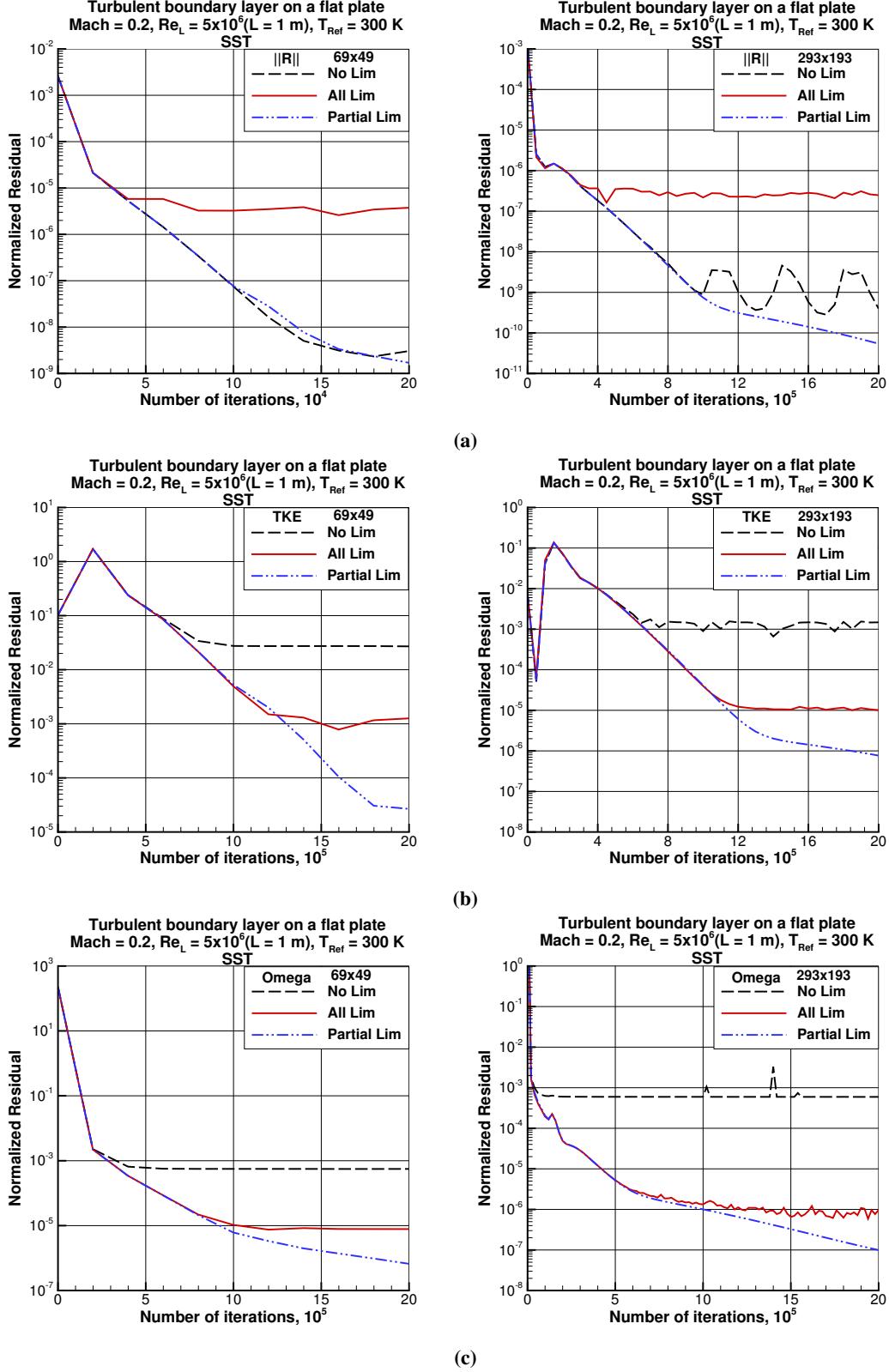
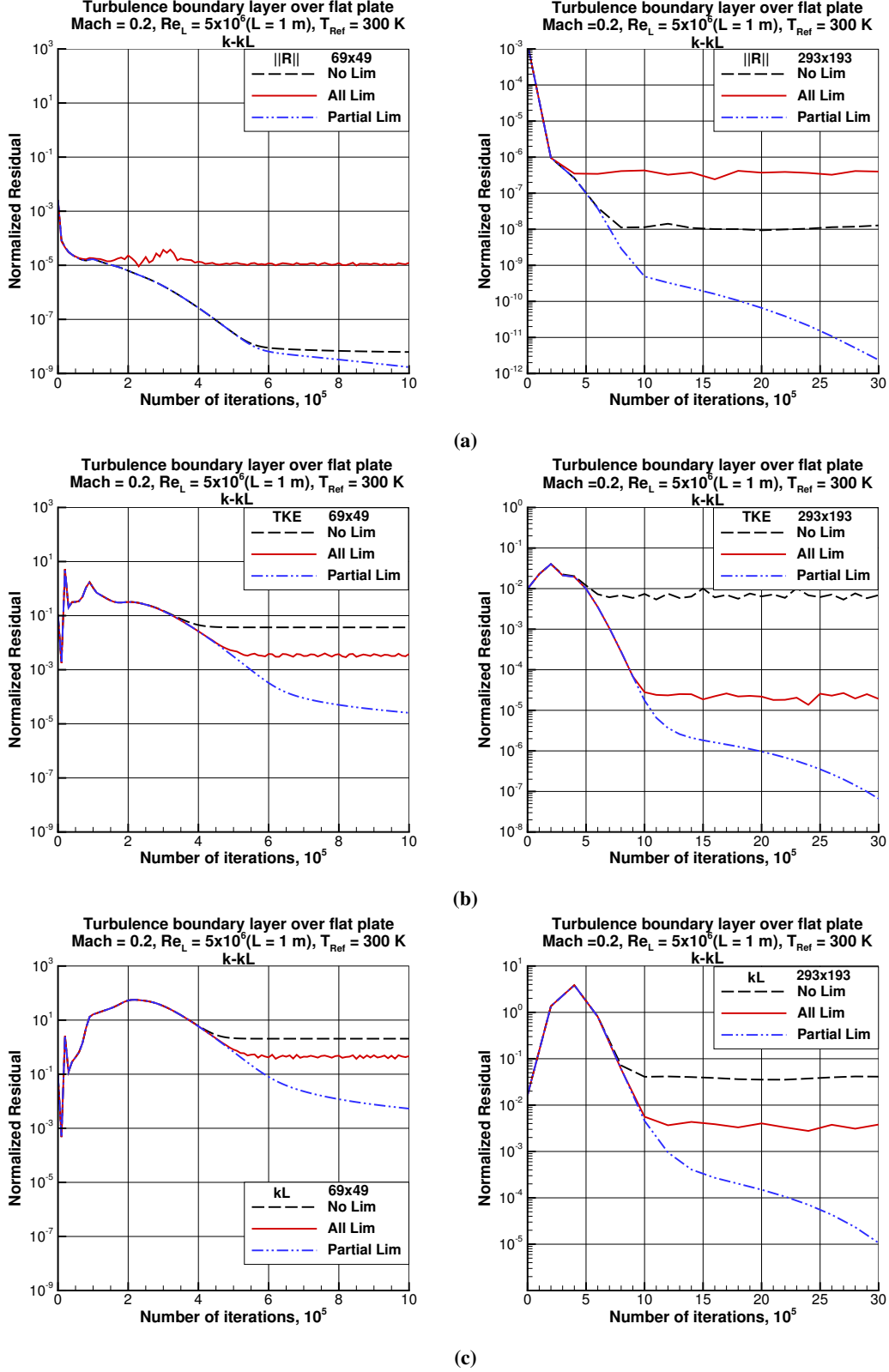
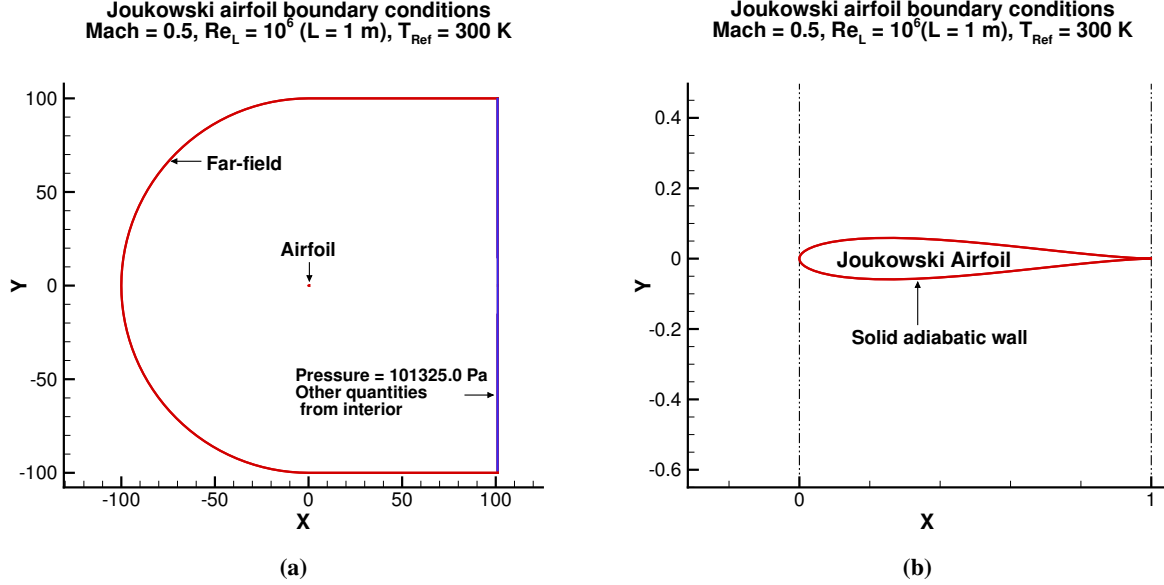


Fig. 15 Turbulent boundary layer simulation over flat plate using SST turbulence model on mesh of  $69 \times 49$  grid points (left column) and mesh of  $273 \times 193$  grid points (right column): a) normalized residual (summed) of continuity, momentum, and energy equations; b) normalized residual for turbulent kinetic energy equation; c) normalized residual for  $\omega$  equation.



**Fig. 16** Turbulent boundary layer simulation over flat plate using  $k - kL$  turbulence model and mesh of  $69 \times 49$  grid points (left column) and mesh of  $273 \times 193$  grid points (right column): a) normalized residual (summed) of continuity, momentum, and energy equations; b) normalized residual for turbulent kinetic energy equation; c) normalized residual for  $kL$  equation.

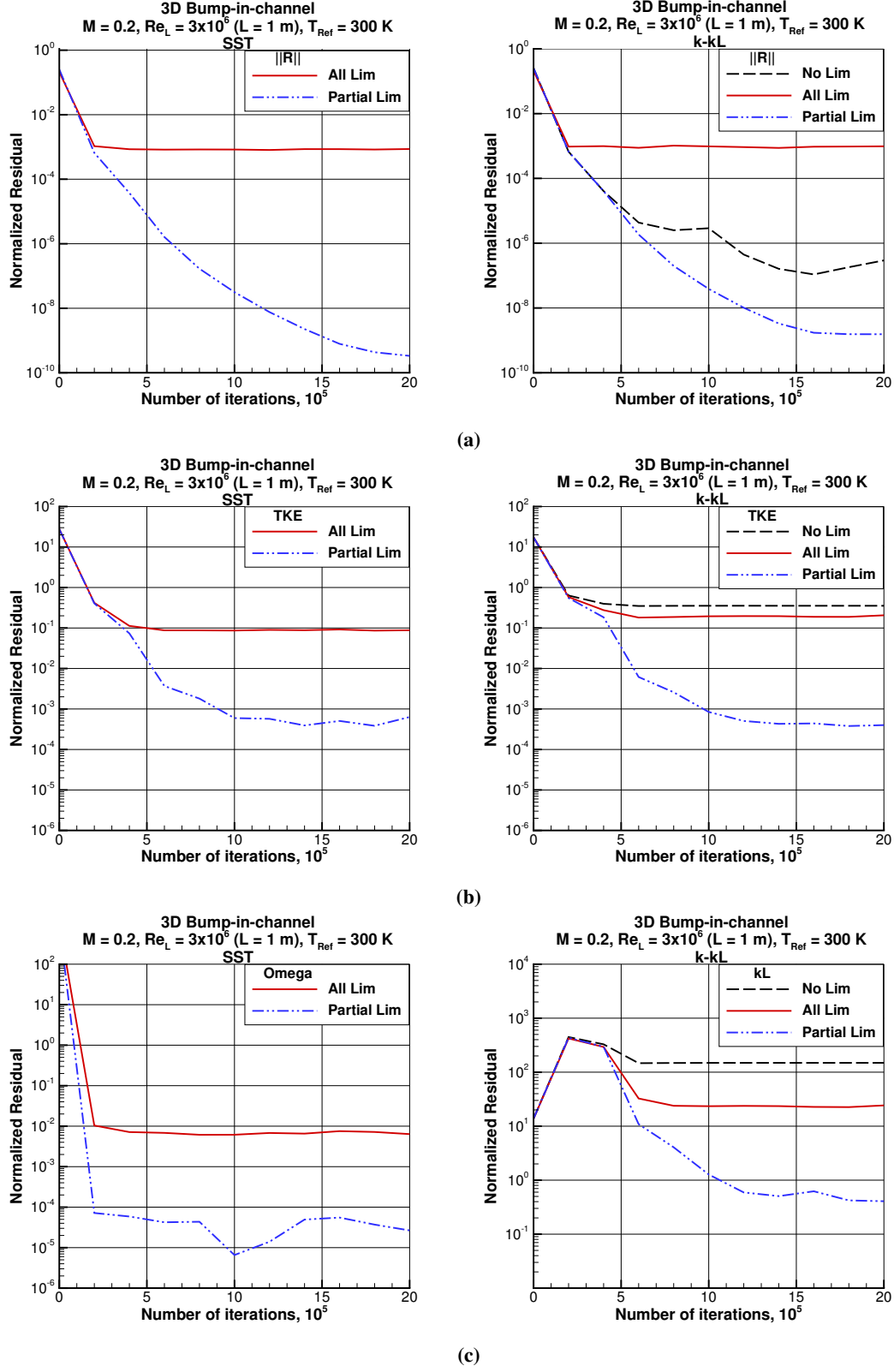
As far as the 'Par Lim' case is concerned, it was observed that convergence of the residuals for the turbulence equations obtained with partial limiting, using SST turbulence model, was better than what was achieved when limiter was applied to all state and turbulence variables as seen in Figs. 15b and 15c. The same trend was seen for k-kl turbulence model. As seen in Figs. 16b and 16c, improved convergence was obtained for  $k$  and  $kL$  equations with limiting used only for  $k$  and  $kL$  variables as compared to when limiting was applied to all state and turbulence variables. Moreover, solution convergence for the Navier-Stokes equations, with either of the turbulence models, was better with partial limiting, compared to the case where no limiting was applied to any of the variables, as seen in Figs. 15a (SST) and 16a (k-kl). The best convergence of the residuals, as such, for both the Navier-Stokes equations and the turbulence equations was obtained with the 'Partial Limiting'.



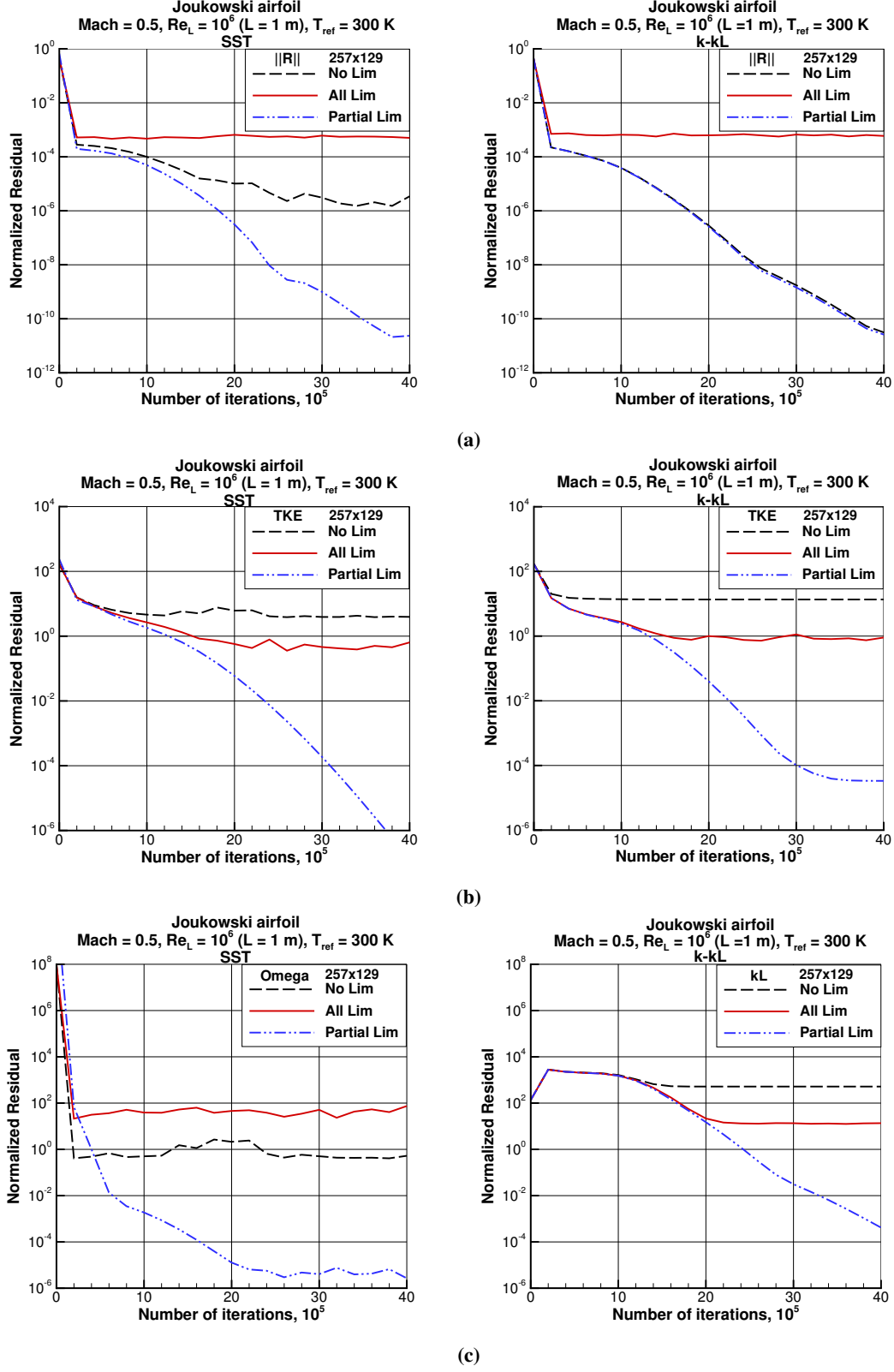
**Fig. 17 Joukowski airfoil test case: a) Full Domain and boundary conditions used; b) Close-up view of Joukowski airfoil**

To investigate whether the convergence behavior of the residuals observed in turbulent flow past a flat plate simulation is present in other turbulent flow cases, two more test cases: Mach 0.2 flow past a 3D bump in a channel, and Mach 0.5 ( $Re_L = 10^6$ ,  $T_{Ref} = 300$  K) flow past a Joukowski airfoil was simulated. Domain, boundary conditions, and numerical methods used for 3D bump case are same as defined in section V.A.7. Domain and boundary conditions for Joukowski airfoil is given in Fig. 17. The airfoil simulation was performed for a computational domain that included only the upper-half using symmetry (as airfoil is symmetric and angle of attack is zero) which was meshed with a grid of size  $129 \times 129$  ( $257 \times 129$  for full domain). For the simulation, AUSM Scheme was used for inviscid flux reconstruction, MUSCL ( $\kappa = 1/3$ ) for higher-order face reconstruction and RK4 for time integration. Figures 18 and 19 shows results for the convergence study performed for the 3D bump-in-channel test case and Joukowski airfoil test case respectively. Both, 3D bump-in-channel and Joukowski test cases show similar patterns of residual convergence for "All Lim", "No Lim" and "Partial Lim" limiting cases as was observed for the flat-plate turbulent boundary-layer simulation.

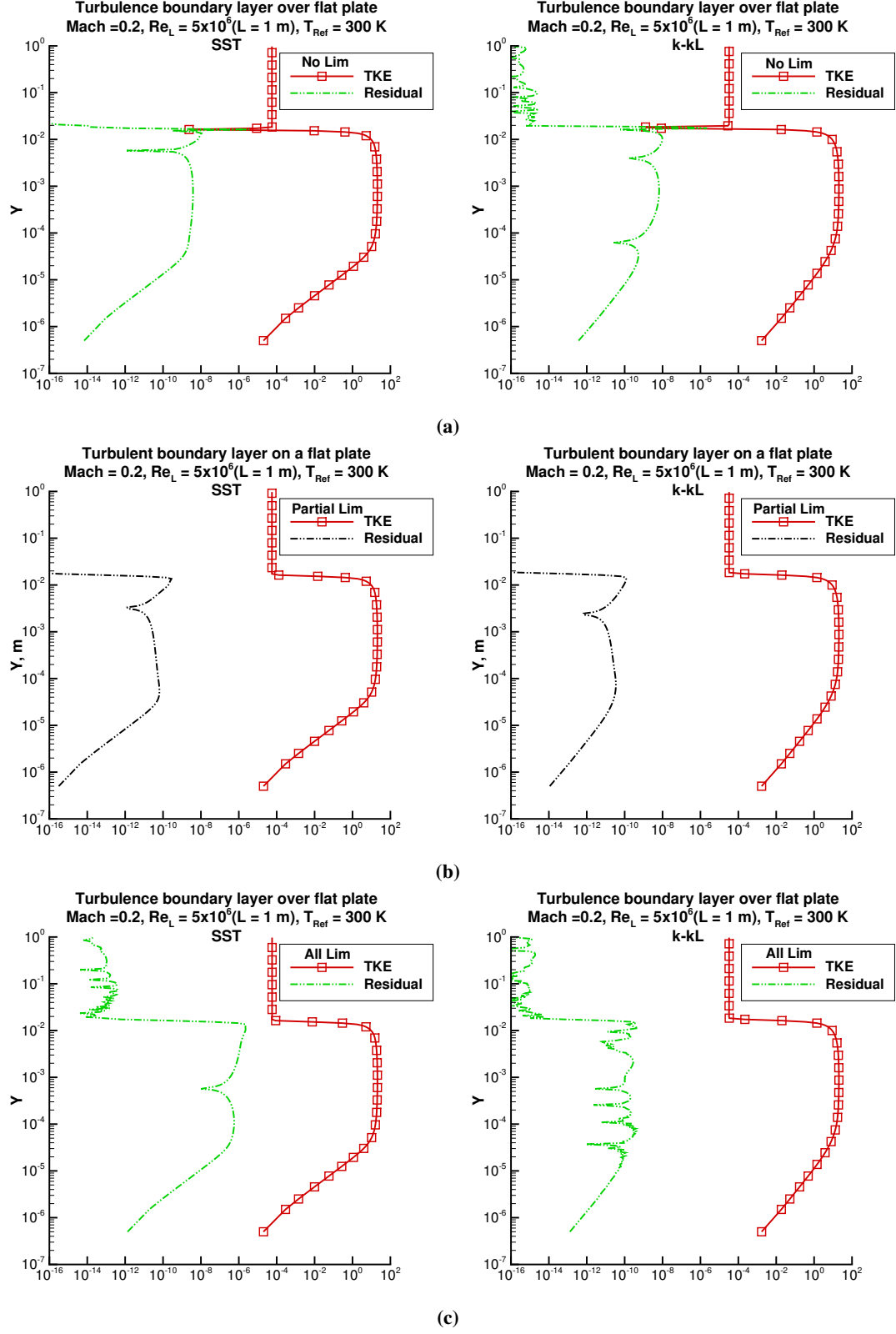
To further investigate the observed trends in the behaviour of the residuals for the Navier-Stokes and turbulence variable equations, the profiles of  $k$  and residual of the  $k$ -equation are extracted at  $x = 1.0$  m for the turbulent flow past a flat plate test case for both the turbulence model: SST and  $k - kL$ . The extracted data is shown in Fig. 20, where the left column presents data for SST turbulence model and right column presents data for  $k - kL$  turbulence model. For both turbulence model in the "No Lim" case, a sharp jump in turbulent kinetic energy and its residual is present at location where TKE reaches its free-stream value. This oscillation can be attributed to the higher order face reconstruction method used in this case. Using limiter for only the turbulence variables ("Par Lim"), removes this oscillations and results in smooth solution. For the "All Lim" case, although no jump is observed in the value of the residual, the values are large compared to the "Par Lim" case, and also observed to contain more noise.



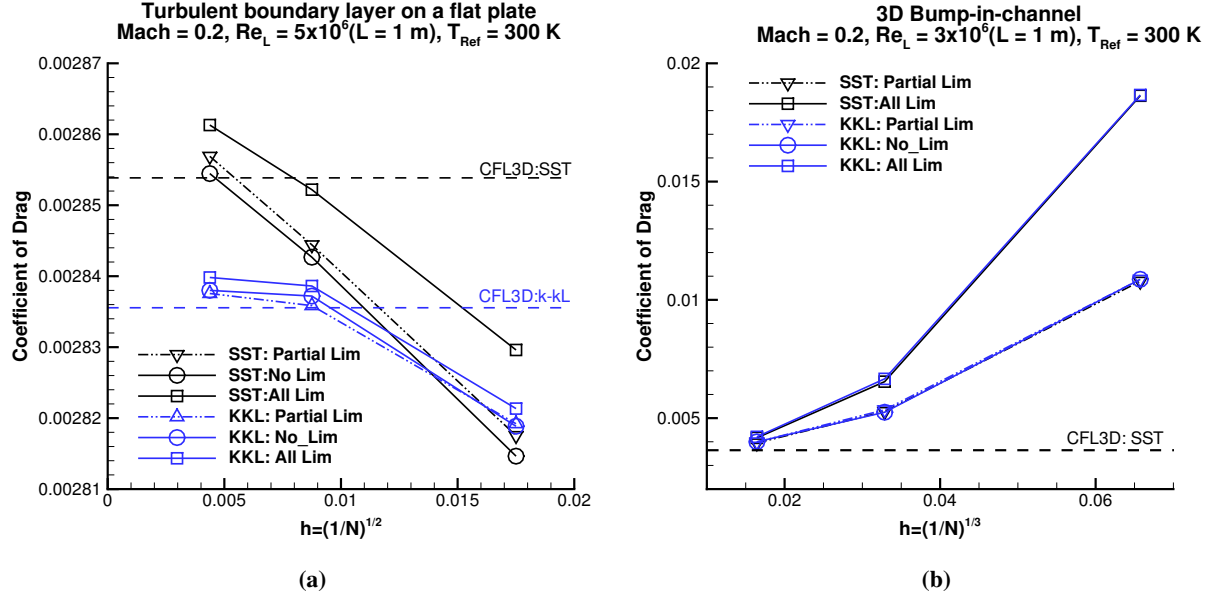
**Fig. 18** Turbulent flow over 3DBump-in-channel simulation using SST (left column) and  $k-kL$  (right column) turbulence model and mesh of  $177 \times 81 \times 17$  grid points: a) normalized residual (summed) of continuity, momentum, and energy equations; b) normalized residual for turbulent kinetic energy equation; c) normalized residual for Omega (left) and  $kL$  (right) equation.



**Fig. 19** Joukowski airfoil simulation using SST (left column) and  $k - kL$  (right column) turbulence model and mesh of  $257 \times 129$  grid points: a) normalized residual (summed) of continuity, momentum, and energy equations; b) normalized residual for turbulent kinetic energy equation; c) normalized residual for Omega (left) and  $kL$  (right) equation.



**Fig. 20** Variation of Turbulent kinetic energy and its residual with  $y$  at  $x = 1.0$  for turbulent flow over a flat plate simulation using SST (left column) and  $k - kL$  (right column) turbulence model with mesh of  $293 \times 193$  grid points: a) No limiting; b) Partial limiting; c) All limiting.



**Fig. 21** Comparison of coefficient of drag for “Partial Lim”, “No Lim” and “All Lim” cases using both turbulence models, SST and k-kL, for a) Turbulent boundary layer on a flat plate test case and b) 3D bump in a channel test case

To get a quantitative comparison between “No Lim”, “Partial Lim” and “All Lim” cases, coefficient of drag ( $C_d$ ) with three different grid sizes ( $h$ ) for both turbulence models: SST and k-kL, is shown in Fig. 21. For both test cases, turbulent boundary layer on a flat plate (Fig. 21a) and 3D bump in a channel (Fig. 21b), a considerable difference in the predicted drag coefficients can be seen between the “Partial Lim” and “All Lim” cases on the coarse grid, whereas the values for the “Partial Lim” and “No Lim” cases almost overlap. For the simulations on the fine mesh, the difference in the predicted coefficient of drag between “Partial Lim” (or “No Lim”) case and “All Lim case” reduces.

## VI. Conclusion

A finite-volume solver, FEST3D on block-structured is presented with verification against multiple standard test cases for inviscid, laminar, and turbulent flows. The solver shows good agreement with analytic solutions and solutions from benchmark codes (CFL3D) for these case. Additionally, a convergence study of the residual of the Navier-Stokes equations and evolution equations for the turbulence variables for the two turbulence models, SST and  $k - kL$ , integrated with the solver, shows that when higher order state reconstruction method is used for turbulence variables a stalling of the residual occurs when no limiting is done. This is shown to happen due to the presence of strong oscillations in the value of the turbulent variable (TKE) and the residual of the turbulence kinetic energy equation at the boundary-layer edge. The convergence of the residual for the turbulence (variable) equations is improved with the application of solution limiting to all the variables, but this results in worse residual convergence for the Navier-Stokes equations. However, limiting only the turbulence variables leads to the lowest values of the residuals for all the equations. It is shown that this partial limiting - application of limiter to turbulence variables - removes this oscillation in the solution (profile of turbulence kinetic energy) and results in better residual convergence and smooth solution.

## Acknowledgments

We would thanks, P.G. Senapathy Center for Computing Resource at IIT Madras, Chennai, for providing time on VIRGO supercluster to perform some of the simulations using FEST3D code.



## References

- [1] Wilcox, D., *Turbulence Modeling for CFD*, 3<sup>rd</sup> ed., DCW Industries, 2006.
- [2] Menter, F. R., “Two-equation eddy-viscosity turbulence models for engineering applications,” *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 1598–1605. doi:10.2514/3.12149.
- [3] Abdol-Hamid, K. S., Carlson, J.-R., and Rumsey, C. L., “Verification and Validation of the k-kL Turbulence Model in FUN3D and CFL3D Codes,” *46th AIAA Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2016, pp. 1–27. doi:10.2514/6.2016-3941.
- [4] Krist, S. L., Biedron, R. T., and Rumsey, C. L., “CFL3D user’s manual (version 5.0),” *NASA Langley Research Center, Hampton, VA*, 1998.
- [5] Z.J. Wang, e. a., Krzysztof Fidkowski, “High-Order CFD Methods: Current Status and Perspective,” *International Journal For Numerical methods In Fluids*, Vol. 72, No. 8, 2013, pp. 1–42. doi:10.1002/fld.3767.
- [6] Menter, F. R., “Improved two-equation k-omega turbulence models for aerodynamic flows,” *NASA Ames Research Center, Moffett Field, CA*, 1992.
- [7] Van Leer, B., “Flux-vector splitting for the Euler equations,” *Eighth international conference on numerical methods in fluid dynamics*, Springer, 1982, pp. 507–512. doi:10.1007/3-540-11948-5\_66.
- [8] Liou, M.-S., and Christopher J Steffen, J., “A New Flux Splitting Scheme,” *Journal of Computational Physics*, Vol. 107, No. 1, 1993, pp. 23–39. doi:10.1006/jcph.1993.1122.
- [9] Edwards, J. R., “A low-diffusion flux-splitting scheme for Navier-Stokes calculations,” *Journal of Computers and Fluids*, Vol. 26, No. 6, 1997, pp. 635–659. doi:10.1016/S0045-7930(97)00014-5.
- [10] Shima, E., and Kitamura, K., “Parameter-Free Simple Low-Dissipation AUSM-Family Scheme for All Speeds,” *AIAA Journal*, Vol. 49, No. 8, 2011, pp. 1693–1709. doi:10.2514/1.J050905.
- [11] Einfeldt, B., “On Gudonov Type Methods for Gas Dynamics,” *SIAM Journal on Numerical Analysis*, Vol. 25, No. 2, 1988, pp. 294–318. doi:10.2514/1.40049.
- [12] Praveen, C., “FVM on unstructured grids,” <http://math.tifrbng.res.in/~praveen/notes/acfd2013/fvm.pdf>, 2013.
- [13] Shu, C.-W., “High-order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD,” *International Journal of Computational Fluid Dynamics*, Vol. 17, No. 2, 2003, pp. 107–118. doi:10.1080/1061856031000104851.
- [14] Colella, P., and Woodward, P. R., “The piecewise parabolic method (PPM) for gas-dynamical simulations,” *Journal of computational physics*, Vol. 54, No. 1, 1984, pp. 174–201. doi:10.1016/0021-9991(84)90143-8.
- [15] Van Leer, B., “Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method,” *Journal of computational Physics*, Vol. 32, No. 1, 1979, pp. 101–136. doi:10.1016/0021-9991(77)90095-X.
- [16] Vreugdenhil, C., and Koren, B., *Numerical Methods for Advection - Diffusion Problems*, Notes on numerical fluid mechanics and multidisciplinary design, Vieweg Friedr. + Sohn Ver, 1993.
- [17] Hoffmann, K., and Chiang, S., *Computational Fluid Dynamics for Engineers*, No. v. 1 in Computational Fluid Dynamics for Engineers, Engineering Education System, 1993.
- [18] Jameson, A., and Baker, T. J., “Solution of the euler equations for complex Problems,” *AIAA*, Vol. 1929, 1983, pp. 293–302. doi:10.2514/6.1983-1929.
- [19] Sod, G. A., “A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws,” *Journal of Computational Physics*, Vol. 27, No. 1, 1978, pp. 1–31. doi:10.1016/0021-9991(78)90023-2.
- [20] Fryxell, B., “Cococubed.com Exact Riemann Solver,” [http://cococubed.asu.edu/code\\_pages/exact\\_riemann.shtml](http://cococubed.asu.edu/code_pages/exact_riemann.shtml), 2018.
- [21] Ghia, U., Ghia, K. N., and Shin, C. T., “High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method,” *Journal of Computational Physics*, Vol. 48, No. 3, 1982, pp. 387–411. doi:10.1016/0021-9991(82)90058-4.
- [22] Childs, M. L., Pulliam, T. H., Jespersen, D. C., Box, P. O., and Field, M., “Turbulence Model Resource Verification Results,” *NASA Tech. Rep.*, 2014.
- [23] Hirsch, C., *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics*, Elsevier Science, 2007.