Spring 5-19-2017

Temperature Corrected Turbulence Models for High Speed Compressible Flows

Shuai Shuai
Washington University in St. Louis

WASHINGTON UNIVERSITY IN ST. LOUIS

SCHOOL OF ENGINEERING & APPLIED SCIENCE

Department of Mechanical Engineering and Material Science

Thesis Examination Committee

Ramesh Agarwal, chair

Qiulin Qu

Swami Karunamoorthy

Temperature Corrected Turbulence Models for High Speed Compressible Flows

by

Shuai Shuai

A thesis presented to School of Engineering and Applied Science
of Washington University in partial fulfillment of the
requirements for the degree of
Master of Science

May 2017

St. Louis, Missouri

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my sincere gratitude to my academic advisor Dr. Ramesh K. Agarwal for his guidance. I would acknowledge my friends in CFD lab, thanks to Ling Zhang and Xiao Zhang for their advice and help.

Shuai Shuai

Washington University in St. Louis

May 2017

ABSTRACT OF THE THESIS

Temperature Corrected Turbulence Models for High Speed Compressible Flows

by

Shuai Shuai

Master of Science in Mechanical Engineering

Washington University in St. Louis, 2017

Research Advisor: Ramesh Agarwal

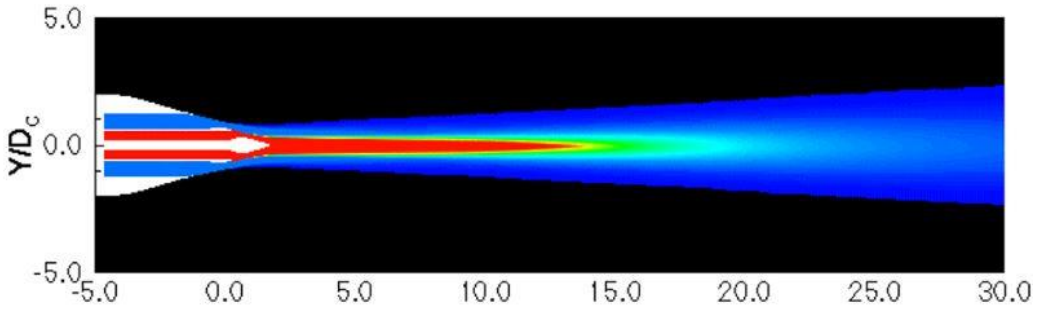
It is well known that the turbulence models were originally developed and applied to compute flows in subsonic and transonic regimes at relatively low Mach numbers and room temperature. This thesis employs new temperature gradient corrections to the eddy viscosity term in two-equation k-ε, and SST k-ω models and one-equation Wray-Agarwal (WA) model for computing flows at high Mach numbers and high temperature. The accuracy of the improved k-ε model with temperature correction is assessed by computing the supersonic/hypersonic boundary layer flow over a flat plate and the jet flows from supersonic nozzles. The validation of the temperature corrections to SST k-ω model and WA model is tested by computing supersonic jet flows. The results of computations using the three temperature corrected k-ε, SST k-ω and WA models are in better agreement with the available theoretical, numerical results, and experimental data compared to those obtained with the industry standard two-equation k-ε model, SST k-ω model and one-equation SA model.

Chapter 1: Introduction and Background

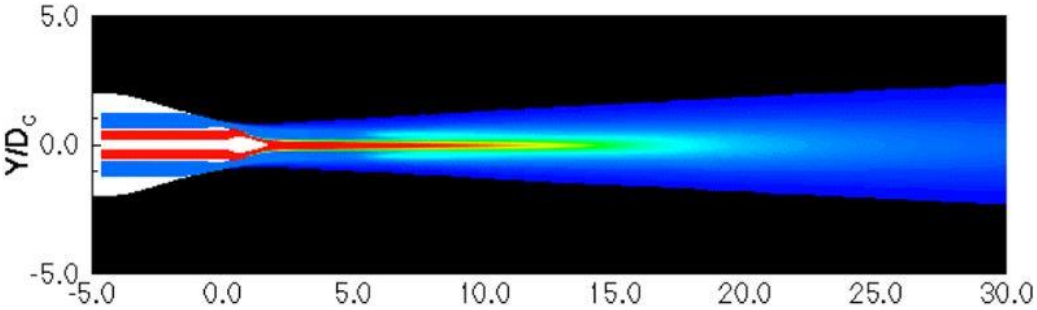
This chapter describes the specific application of turbulence models to hot jet existing from the nozzle of aircraft engines, their current deficiency in accurate prediction of the flow fields of high temperature high Mach number flows and the focus of this thesis on developing temperature corrected turbulence models to improve their prediction capability.

1.1 Application of Turbulence Model to Hot Jet Flow Simulations

Propulsion noise from an aircraft jet engine has a significant effect on both passengers and people on the ground. Thus the acoustic effect of propulsion and jet noise are of major concern in the development of aircraft engines. Many noise reduction devices are designed and the acoustics of these devices is studied and tested using computational simulations. For example, Chevron nozzle is one kind of device developed in last few decades to reduce the jet noise, the chevrons penetrate into the core flow and decrease the length of the exhaust jet plume. Figure 1 shows the total temperature contour of the jet exhaust plume. Figure 2 shows the total temperature contour in the plume cross sections at x/D = 2, 5, and 8. These two figures indicate that the chevrons around the engine cowl at the nozzle exit have a significant influence on the length of the exhaust jet plume. According to Seiner and Jansen [1], the jet noise has a positive correlation with the exhaust jet plume. To specify the design parameters for these chevrons, accurate turbulent flows field simulations of the jet plume are referred, which require a turbulence model that can accurately predict the high temperature high Mach number jet flows, using the compressible Reynolds Averaged Navier-Stokes (RANS) equations.

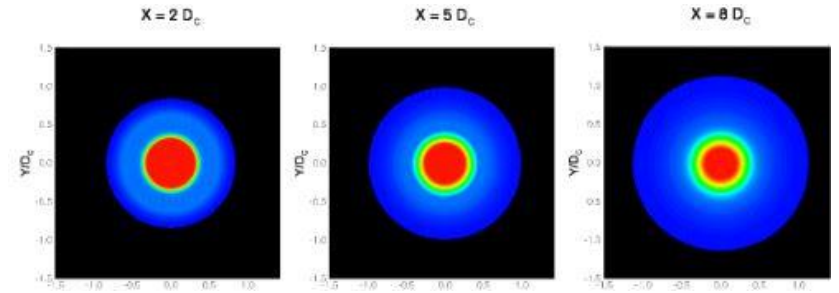


(a) Nozzle without the chevrons

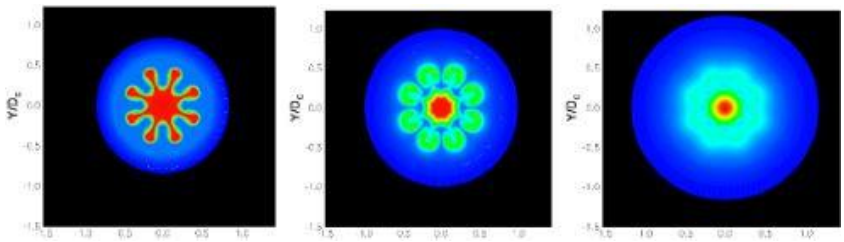


(b) Nozzle with the chevrons

Figure 1. Total temperature contours in the symmetry plane of the exhaust jet plume.



(a) Nozzle without the chevrons



(b) Nozzle with the chevrons

Figure 2. Total temperature contours in cross sections of a jet exhaust plume at x/D = 2, 5, and 8.

1.2 Deficiency in Standard Turbulence Models

High-speed high temperature compressible flows present a new challenge to turbulence modeling. Since Reynolds Averaged Navier-Stokes (RANS) equations are widely used to calculate the jet flows. One and two-equation eddy-viscosity models are developed first for incompressible flows (subsonic free-shear flows and incompressible boundary layers), then an additional correction is applied to extend their usefulness to compressible flows. These extensions are often derived from Morkovin's hypothesis, which states that the compressibility affects the turbulence through variations in mean density and that the density fluctuations have little effect on turbulence. Experimental and numerical simulations have largely confirmed this hypothesis for moderate Mach numbers. However, free-shear flows with $M > 3$ and wall bounded flows with $M > 4.5$ have been shown to be clearly outside the range of this hypothesis. The density gradient at high Mach number and high temperature increase the flow instability and thereby influence the turbulence in the flow. Therefore, accurate simulation of turbulent flows at supersonic and hypersonic Mach numbers with high temperatures and temperature gradients requires advanced turbulence models. The industry standard one- and two-equation turbulence models, namely the Spalart-Allmaras (SA) model, SST k-ω model and k-ε model to name a few lack the ability to simulate turbulent boundary layers and the increase in growth rate of mixing layers in high- temperature high Mach number flows.

1.3 Limitations of Many Proposed Modifications to Turbulence Models

Many researchers have proposed several modifications to turbulence models to address the above-mentioned deficiency in the models for computing high temperature high Mach number flows [2] [3] [4]. Majority of proposed modifications modify the heat flux term in the RANS equations and

some employ a temperature dependent eddy viscosity. However, these modifications change the basic formulation of RANS equations on an adhoc basis and have not been able to accurately predicted high temperature high Mach number flows.

1.4 Temperature Correction based Turbulence Models

Total temperature and total pressure are two main parameters in nozzle flow. Both pressure fluctuations and temperature fluctuations change the flow density and lead to nonzero dilatation; the dilatation then affects the turbulence kinetic energy and pressure-strain correlation. Many turbulence models have developed correction terms to account for the pressure fluctuations, however, in jet flows, temperature fluctuations play an important role in affecting turbulence. The experiment conducted by Seiner and Thomas has shown that total temperature gradient has a significant influence on the growth rate of mixing layer [1] [5]. In CFD prediction of nonisothermal flows, ignoring this effect leads to poor results. Thus a correction accounting for the temperature fluctuations becomes important in turbulence modeling. This thesis develops a temperature correction to several standard turbulence models namely the k-ε, SST k-ω and Wray-Agarwal (WA) [6] and assesses their accuracy by computing several cold and hot jet exhaust flows from supersonic nozzles.

1.5 User Defined Function (UDF) for Temperature Corrected Turbulence Models

A user-defined function or UDF is a function programmed in C and can be compiled with the FLUENT solver to customize the program to make the required changes. UDFs can be used to make changes to physical models, numerical algorithms, turbulence models, etc. In this thesis, UDF is developed to enhance existing FLUENT models by customizing boundary conditions,

developing source terms in making changes in transport equations including the turbulence model equation, and creating user-defined scalar (UDS), etc. Some key UDF code developed in this thesis is given in the Appendix.

Chapter 2: Temperature Corrected k-ε Model

Abdol-Hamid et al. [7] proposed a temperature corrected *k-ε* model. This chapter validates his temperature correction to *k-ε* model by computing the supersonic flat plate flows and supersonic jet exhaust flows from Eggers nozzle [8] and Seiner nozzle [9].

2.1 Two-equation k-ε Turbulence Model

The *k-ε* model is the most widely used two-equation turbulence model in which model transport equations are solved for the turbulent kinetic energy *k* and turbulent dissipation *ε*. From these two quantities, quantities can be calculated, e.g., the turbulent eddy viscosity.

The equations for *k* and *ε* are given by

$$\frac{Dk}{Dt} = \frac{\partial}{\partial x_j}\left[\frac{\nu_t}{\sigma_K}\frac{\partial k}{\partial x_j}\right] + \nu_t \frac{\partial \overline{u_i}}{\partial x_j}\left[\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}\right] - \varepsilon \tag{1}$$

$$\frac{D\varepsilon}{Dt} = \frac{\partial}{\partial x_j}\left[\frac{\nu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_j}\right] + C_1 \nu_t \frac{\varepsilon}{k}\frac{\partial \overline{u_i}}{\partial x_j}\left[\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}\right] - C_2 \frac{\varepsilon^2}{k} \tag{2}$$

The turbulent kinematic eddy viscosity is computed by the equation:

$$\nu_t = C_\mu \frac{k^2}{\varepsilon} \tag{3}$$

The five constants in the standard *k-ε* model are:

$$C_\mu = 0.09, \qquad C_1 = 1.44, \qquad C_2 = 1.92, \qquad \sigma_K = 1.0, \qquad \sigma_\varepsilon = 1.3 \tag{4}$$

2.2 Temperature Corrected k-ε Model

A modification to the standard *k-ε* model has been developed by Abdol-Hamid et al. [7]. For computing high temperature compressible flows by including a temperature gradient term in the kinematic eddy viscosity which makes the kinematic eddy viscosity ν_t a function of the total temperature gradient *Tt*. Instead of deriving this correction analytically, this correction is based on the experimental data. A non-dimensional variable *Tg* is defined as a function of total temperature as follows:

$$T_g = \frac{k^{3/2}}{\varepsilon}\frac{|\nabla T_t|}{T_t} \tag{5}$$

To make the new model applicable to high speed flows, a turbulent Mach number is also included in the modified definition of kinematic eddy viscosity. The turbulent Mach number is given by the equation:

$$M_\tau = \frac{\sqrt{2k}}{a}\frac{Dk}{Dt} \tag{6}$$

where a is the local speed of sound and k is the turbulent kinetic energy. The kinematic eddy viscosity is given as a function of total temperature gradient as:

$$\nu_t = 0.09\left[1 + \frac{{T_g}^3}{0.041 + f(M_\tau)}\right]\frac{k^2}{\varepsilon}\nu_t = C_\mu\frac{k^2}{\varepsilon} \tag{7}$$

where

$$f(M_\tau) = \left({M_\tau}^2 - {M_{\tau 0}}^2\right) H(M_\tau - M_{\tau 0}) \tag{8}$$

And *H(x)* is the Heaviside step function. $M_{\tau 0}$ = 0.1 is an empirical constant. It should be noted that the term in the square bracket in Eq. [7] is the modification by Abdol-Hamid et al. to the standard *k-ε* model; there are no other changes. It turns out that in case of some free shear layer and jet flows at high Mach numbers, ν_t given by Eq. [7] can acquire very high values which can lead to divergence of the solution. Therefore, an upper bound on the value of ν_t may be required.

2.3 Temperature Correction Method

The first step in the correction method is to develop a non-dimensional local total temperature gradient by employing the local turbulence length scale:

$$T_g = L \frac{|\nabla T_t|}{T_t} \tag{9}$$

In the two-equation *k-ε* model, the length scale *L* is given by:

$$L = \frac{k^{3/2}}{\varepsilon} \tag{10}$$

As Eq. [5] shows, the non-dimensional total temperature gradient T_g can be derived from length scale, total temperature and the absolute value of its gradient. The influence of compressibility at high temperature and high Mach number is accounted for by Eq. [6] and Eq. [8]. Thus, Eq. [7] consists of compressibility factor and non-dimensional total temperature gradient factor.

2.4 Test Cases and Results

The commercial CFD solver ANSYS FLUENT is used for simulations. Grids are generated using ICEM CFD. The convection terms are discretized using a third-order upwind scheme. The viscous terms are central differenced using a second-order scheme. Time discretization is achieved by a first-order implicit scheme. Reynolds-Averaged Navier-Stokes (RANS) equation are solved in conjunction with the modified k-ε turbulence model described in section 2.1. High temperature modifications to standard k-ε model, described in section 2.2 are included by a User Defined Function (UDF). All the computations presented in this section have been performed on grids which ensure the grid independence of the solution.

2.4.1 Supersonic Flow past a Flat Plate

Supersonic flow past a flat plate is computed for the following flow conditions:

Table 1. Flow condition for supersonic flow past a flat plate

Case #	Mach number	Ambient Temperature	Wall Temperature
1	3.0	288K	300K
2	3.0	288K	500K
3	3.0	288K	900K

The computational domain grid used in the simulations is shown in Figure 3. A highly clustered fine grid of size 545×385 is employed to obtain almost an exact solution to compare the calculations with the benchmark solutions reported by van Driest [10].

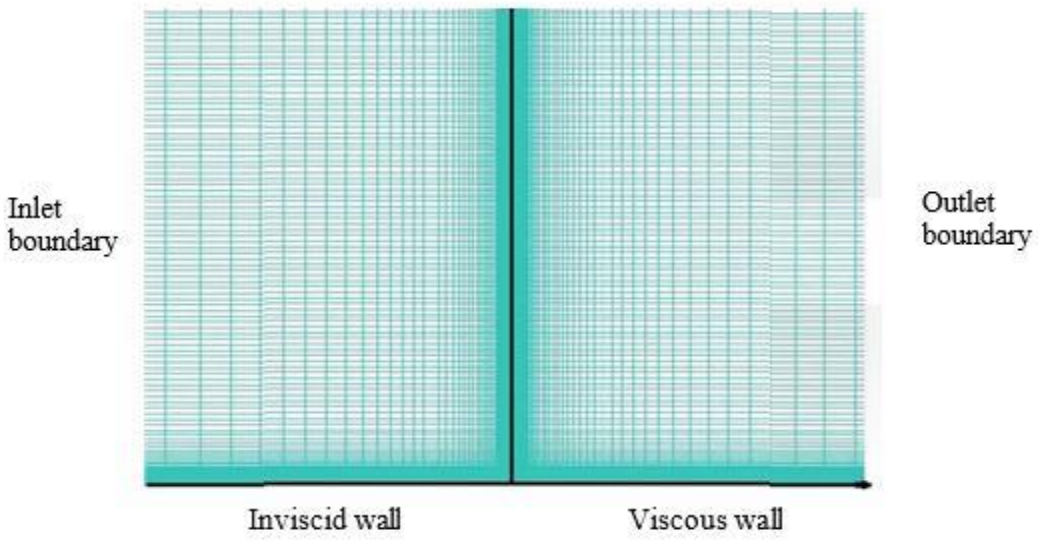


Figure 3. Computational domain, grid, and boundary conditions for computing supersonic flow past a flat plate.

Figure 4. Shows the comparison of skin friction coefficient between the present simulations and van Driest' benchmark solutions [10] that are considered as good as the exact solutions.

O'Donnell [11] has measured the velocity profile in a compressible turbulent boundary layer shown in Figure 5. The measured profile is shown as a plot of u/U_∞ versus y/δ_2 for $M_\infty = 2.4$. δ_2 is the momentum thickness defined as:

$$\delta_2 = \int_0^\delta \frac{T_1}{T}\frac{u}{U}\left(1 - \frac{u}{U}\right) dy \tag{11}$$

where δ is the boundary layer thickness and T_1 is the temperature at the outer edge of the boundary layer. In Figure 5, the solid black line represents the seventh power law for compressible boundary layer given by Eq. (11) which fits the data of O'Donnell [11] :

$$\frac{u}{U_\infty} = 0.683\left(\frac{u}{\delta_2}\right)^{1/7} \tag{12}$$

Figure 5 also shows the results of present computations at various locations on the plate represented in terms of local Reynolds number *Rex*. The calculations show a similarity profile as expected. However, they do not match the seventh power law given by Eq. (12) representing the experiment of O'Donnell [11].

In conclusion, the computations with the temperature corrected k-ε model agree with the van Driest's formula which is considered highly accurate for compressible boundary layer flows but are not able to predict the profiles based on the seventh power law.

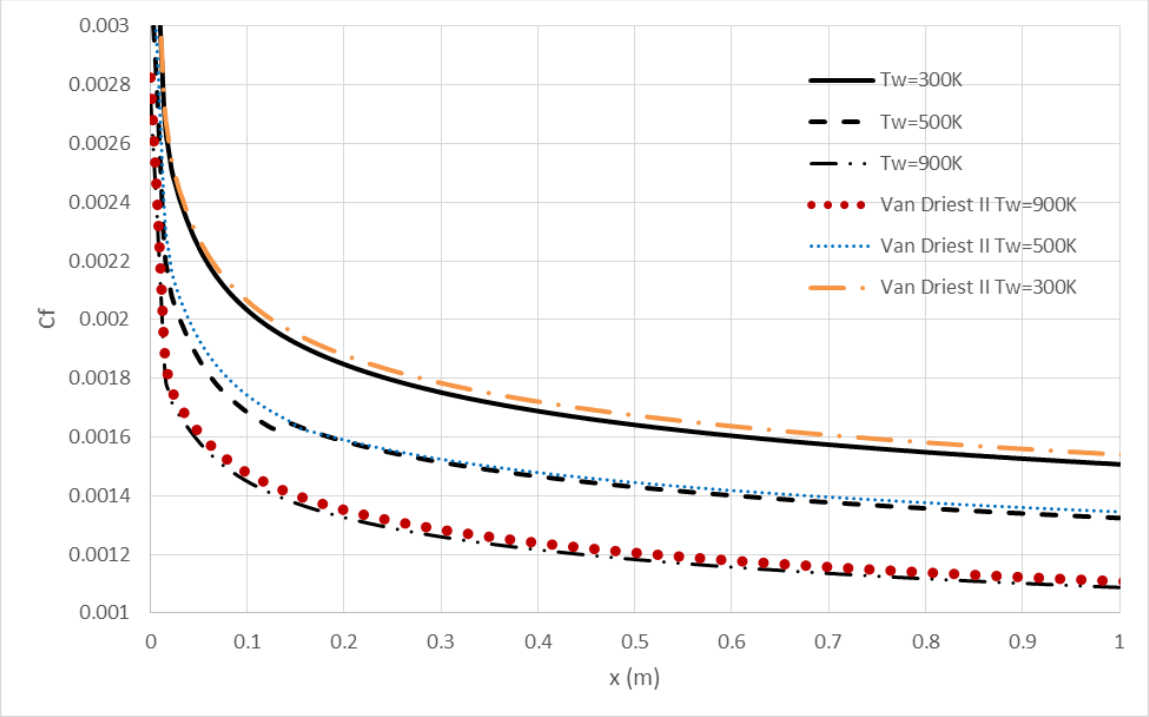


Figure 4. Comparison of present computations with van Driest solution [10] for skin friction coefficient for supersonic flow past a flat plate, M∞ =3.0, T∞ = 288K.

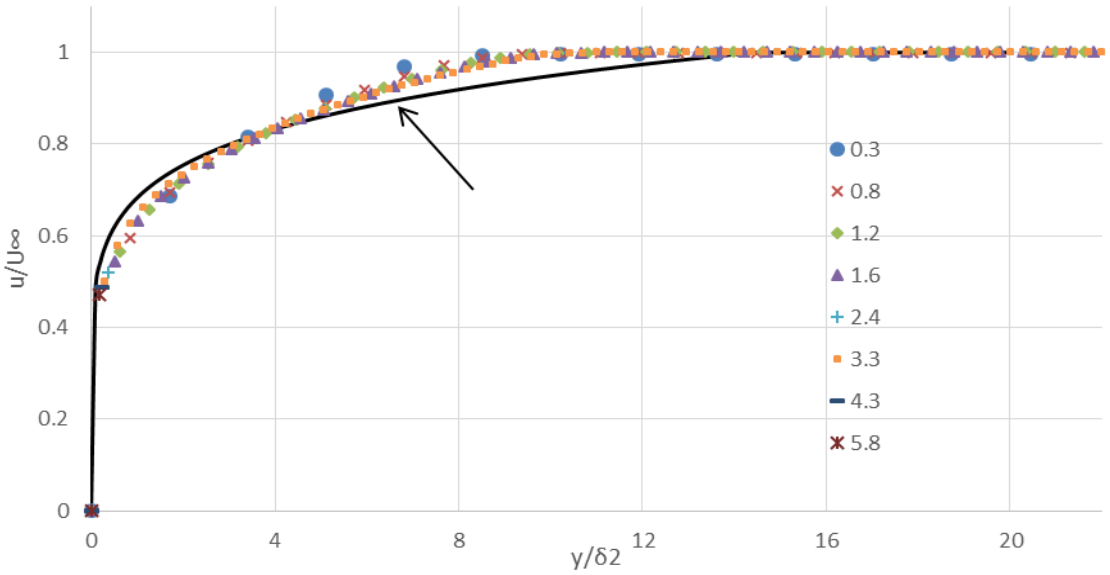


Figure 5. Comparison of the computed self-similar velocity profile and seventh power law for a supersonic turbulent boundary layer on a flat plate.

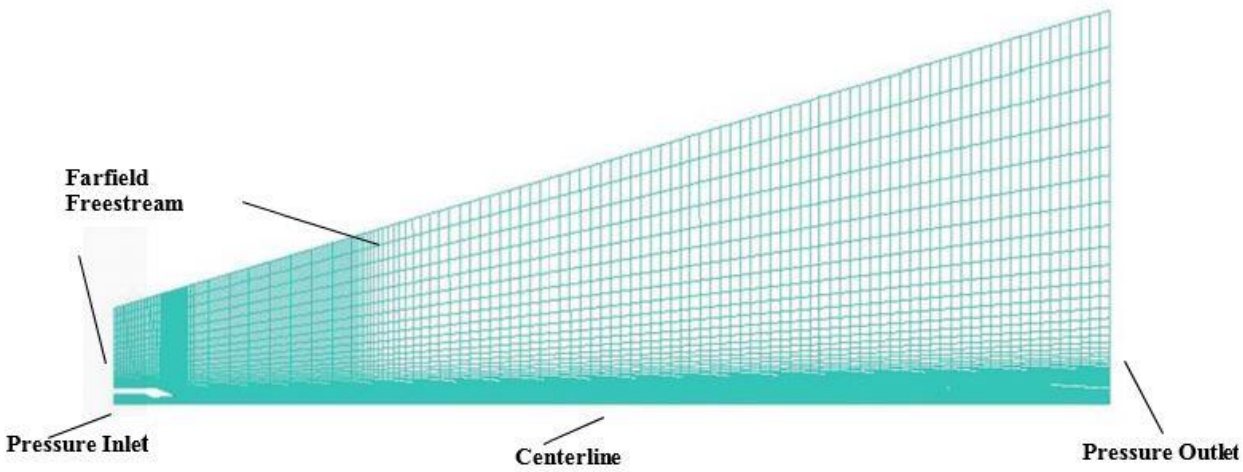
2.4.2 Jet Exhaust Flow from a Supersonic Nozzle – Eggers' Experiment [8]

In this section, the results of computation for exhaust flow from an axisymmetric C-D supersonic nozzle (Eggers Nozzle) using the compressible RANS equations with temperature corrected *k-ε* model are presented. The experiments for this nozzle have been performed by Eggers [8]. The geometry of the nozzle is obtained from Shoemaker [12]. The computational domain, grid and the boundary conditions employed in the computations are shown in Figure 6. The grid is generated using ANSYS ICEM. The features of the CFD solver ANSYS FLUENT used in the simulation have already been described in the first paragraph of section 2.4. The exit Mach number of the nozzle is M = 2.2.

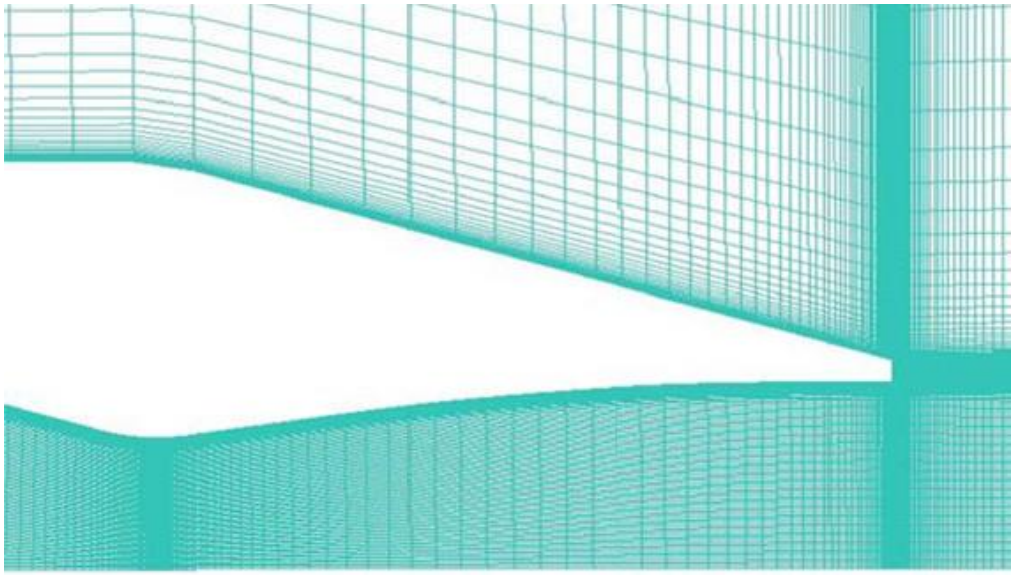
Three levels of the grid from coarse, medium to fine consisting of 32092, 62362 and 126494 nodes are used in the simulation to determine their effect on the solution. Figure 7(a) shows the effect of three levels of grids on axial velocity along the centerline and Figure 7(b) shows the effect of three

levels of the grid on velocity profile at x/r = 45.94. The results from the medium and fine grid are very close. Therefore the medium grid is employed in the computations.

Figure 8 shows the comparison of axial velocity profile computed using the SST k-ω model, the standard *k-ε* model, and the temperature corrected *k-ε* model with the experimental data of Eggers. Figure 9 shows the comparison of velocity profiles at various x/r locations in the exhaust computed using the SST *k-ω* model, the standard *k-ε* model and the temperature corrected *k-ε* model with the experimental data of Eggers. Figure 8 and Figure 9 show that both the standard *k-ε* model and temperature corrected *k-ε* model give a good prediction in the core region of the exhaust flow as well as in regions near the exit of the nozzle. However, the temperature corrected *k-ε* model is more accurate than the standard *k-ε* model in the downstream locations of the jet flow. The results of computations using the SST *k-ω* model do not match the experimental data at all.

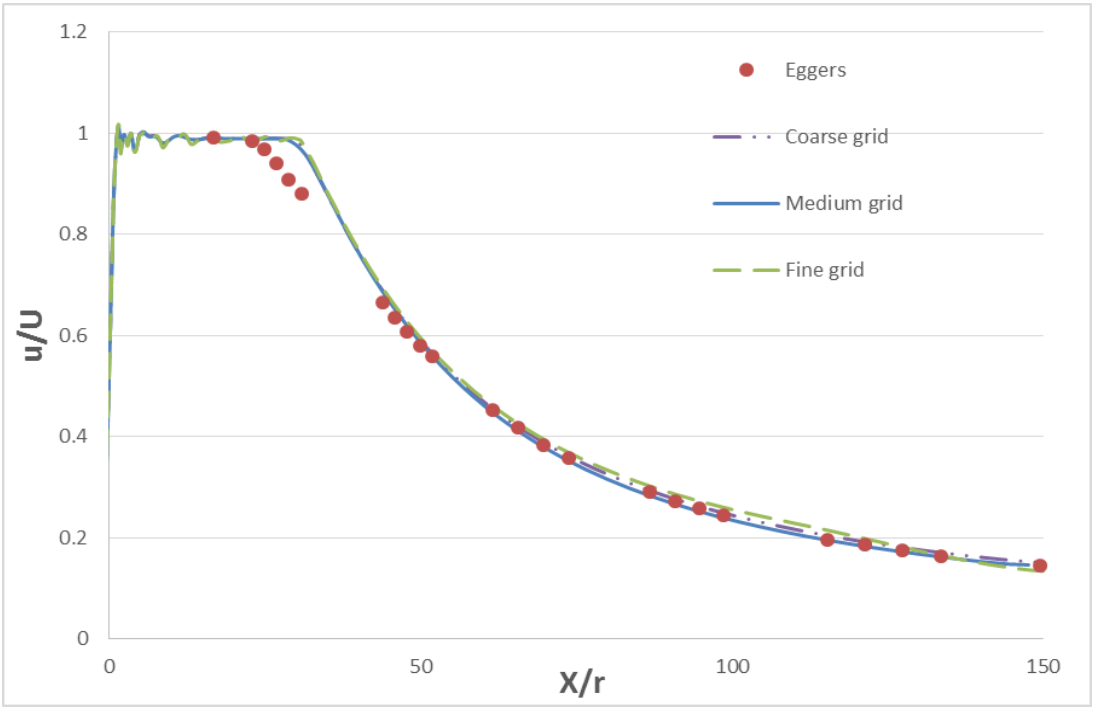


(a) Computational domain, grid and the boundary conditions for Eggers nozzle.

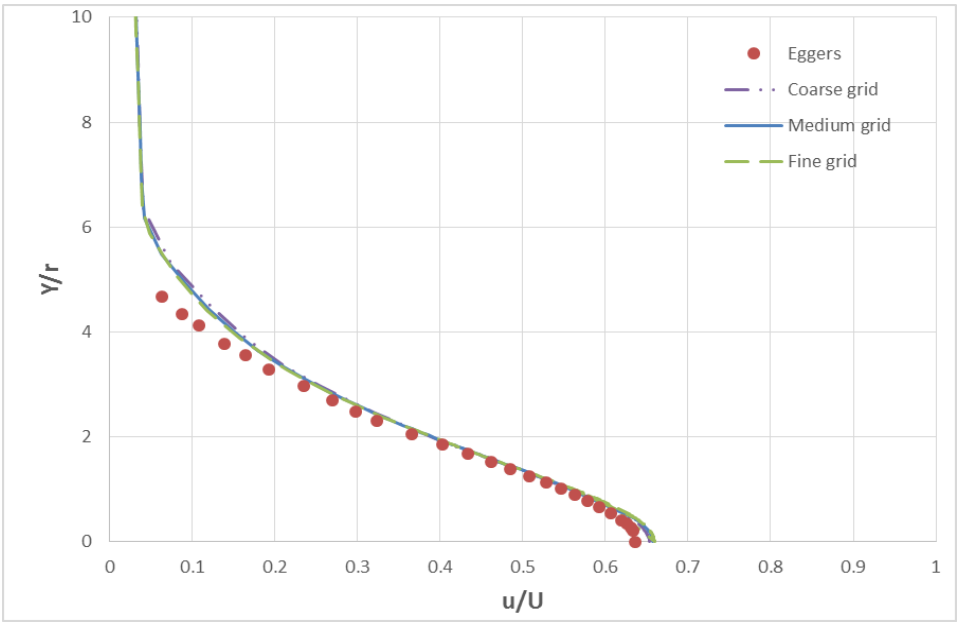


(b) Zoomed-in-View of the mesh near the inlet

Figure 6. Computational domain, grid, and the boundary conditions.



(a) Comparison of computed centerline velocity using three levels of the grid with experimental data.



(b) Comparison of computed velocity profile at x/r = 45.9 using three levels of the grid with experimental data.

Figure 7. Grid refinement study for the computation of Eggers's nozzle exhaust.

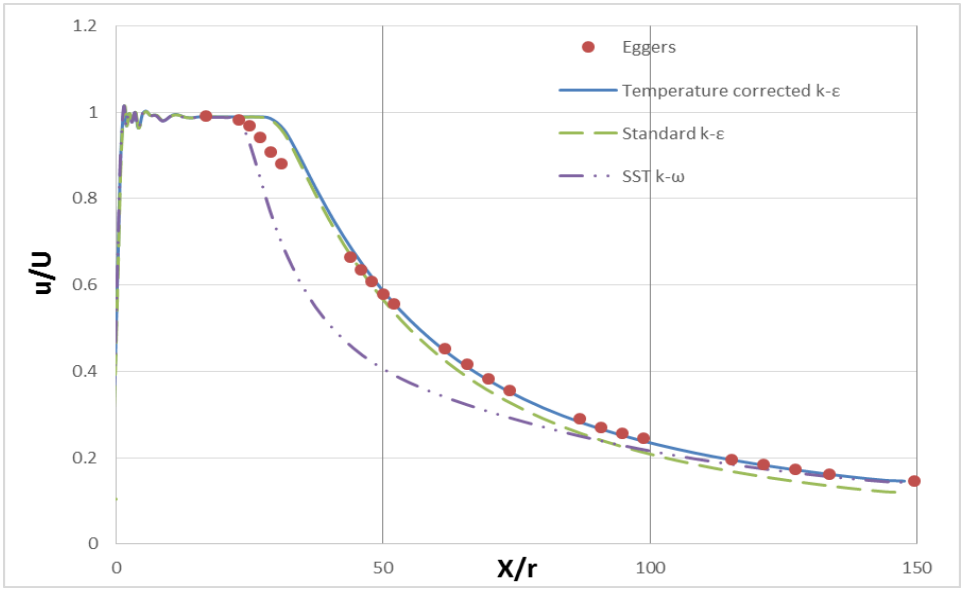
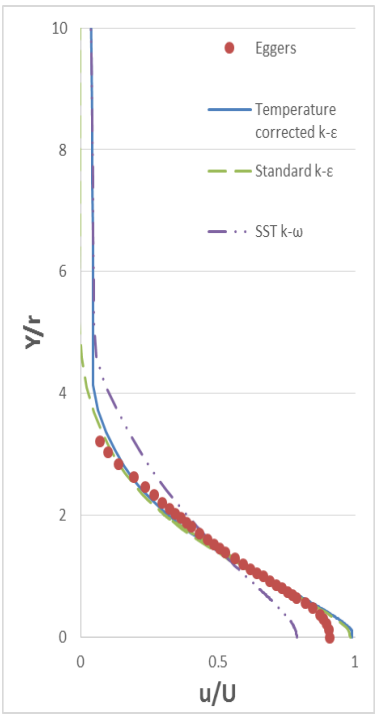
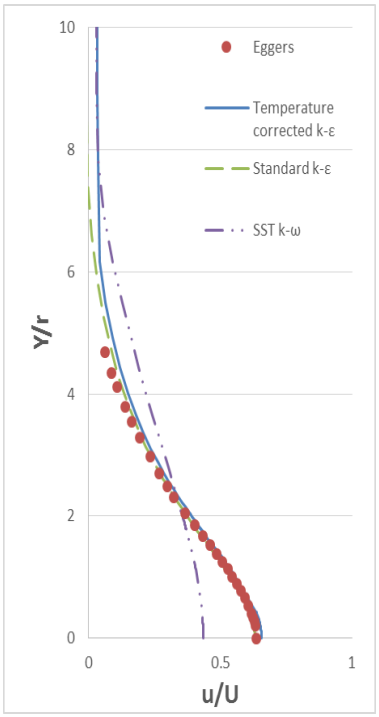


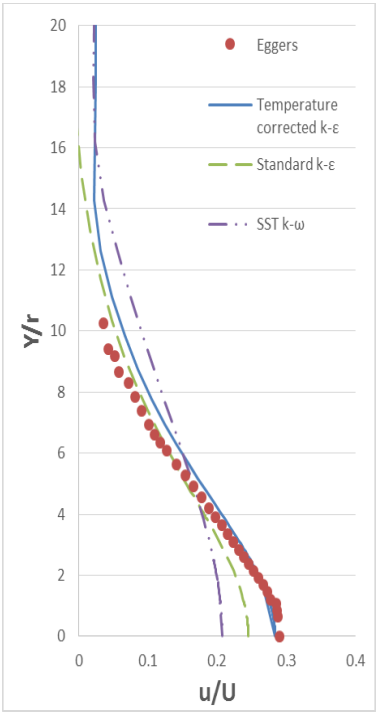
Figure 8. Comparison of computed centerline velocity using the three turbulence models with experimental data.



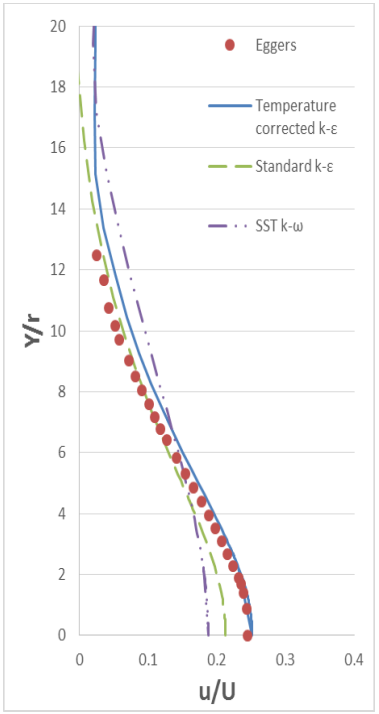
x/r=28.93



x/r=45.94



x/r=89.90



x/r=98.89

Figure 9. Comparison of computed velocity profiles at various x/r locations using the three turbulence models with experimental data.

2.4.3 Jet Exhaust Flow from a Supersonic Nozzle – Seiner's Experiment [9]

In this section, the results of computation for exhaust flow from another axisymmetric C-D supersonic nozzle (Seiner Nozzle) using the RANS equations with temperature corrected k-ε model are presented. The experiments for this nozzle have been performed by Seiner et al. [9]. In this case, the mixing of the free stream with the nozzle exhaust flow was considered in the experiment; the geometry of the nozzle is obtained from the NASA Turbulence Modeling Resource (TMR) website. The computational domain, grid and the boundary conditions employed in the computations are shown in Figure 10. The grid is generated using ANSYS ICEM. The features of the CFD solver ANSYS FLUENT used in the simulation have already been described in the first paragraph of section 2.4. The nozzle exit Mach number is M = 2.0.

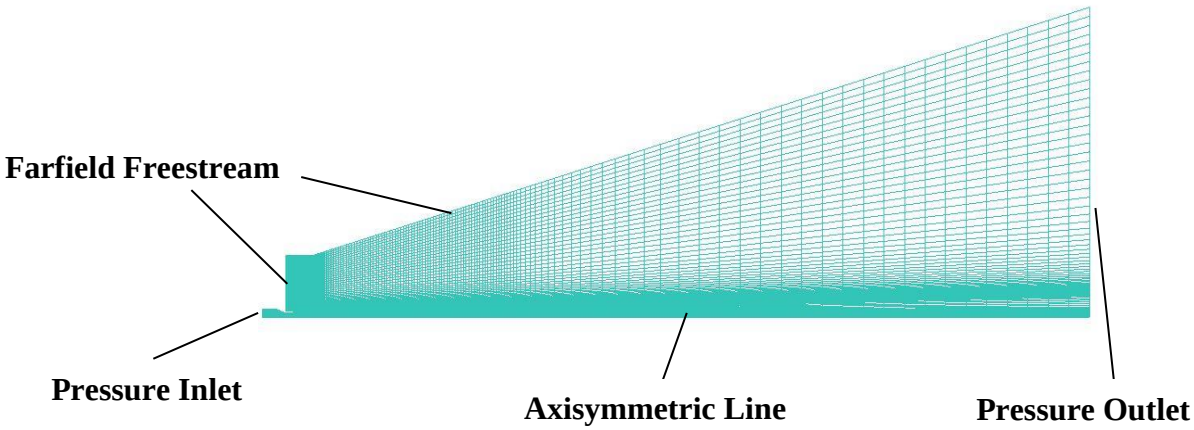
Three levels of the grid from coarse, medium to fine consisting of 17914, 43712 and 61168 nodes are used in the simulation to determine their effect on the solution. Figure 11 shows the effect of three levels of grids on axial velocity along the centerline. The results from the medium and fine grid are very close. Therefore the medium grid is employed in the computations.

Three simulations are conducted by varying the total temperature Tc of the reservoir at 313K, 755K, and 1116K; the total pressure Pc of the reservoir is kept the same at 792,678 Pa along with the free stream conditions. The freestream temperature T_0 = 313K, freestream pressure P_0 = 101,325 Pa and freestream Mach number M_0 = 0.02. Figures 12(a) and 12(b) respectively show the comparison of variation in Mach number and total pressure at Tc = 313K along the centerline of the nozzle computed using SST k-ω model, the standard k-ε model and the temperature corrected k-ε model with the experimental data of Seiner et al. [9]. In this figure, the variation in total temperature along the centerline is not shown since Tc = T_0 = 313K. Figures 13 (a) – (c) respectively show the comparison of variation in Mach number, total pressure, and total

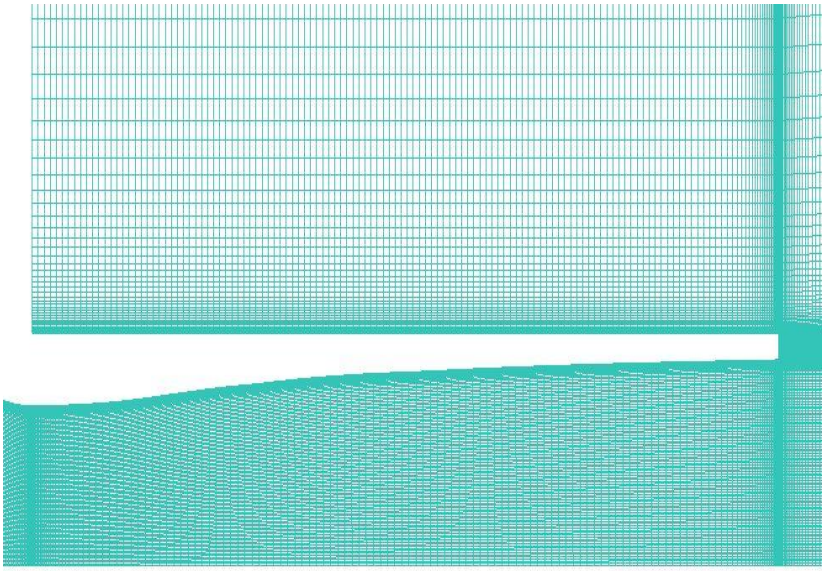
temperature at Tc = 755K along the centerline of the nozzle computed using the SST *k-ω* model, the standard *k-ε* model and the temperature corrected *k-ε* model with the experimental data of Seiner et al. [9]. Figures 14 (a) – (c) respectively show the comparison of variation in Mach number, total pressure, and total temperature at Tc = 1116K along the centerline of the nozzle computed using the SST *k-ω* model, the standard *k-ε* model and the temperature corrected *k-ε* model with the experimental data of Seiner et al. [9]. Figures 12-14 show that both the standard *k-ε* model and temperature corrected *k-ε* model give reasonably good prediction compared to the experimental data; however, the temperature corrected *k-ε* model is more accurate in predicting the experimental results at all Tc. The results of computations using the SST *k-ω* model do not match the experimental data at all.

Table 2. Flow conditions for Seiner Nozzle

Case #	Tc [K]	Pc [pa]	T_0 [K]	P_0 [Pa]	M_0
1	313	792678	313	101325	0.02
2	755	792678	313	101325	0.02
3	1116	792678	313	101325	0.02



(a) Computational domain, grid and the boundary conditions for Seiner nozzle.



(b) Zoomed-in View of mesh near the inlet.

Figure 10. Computational domain, grid, and the boundary conditions.

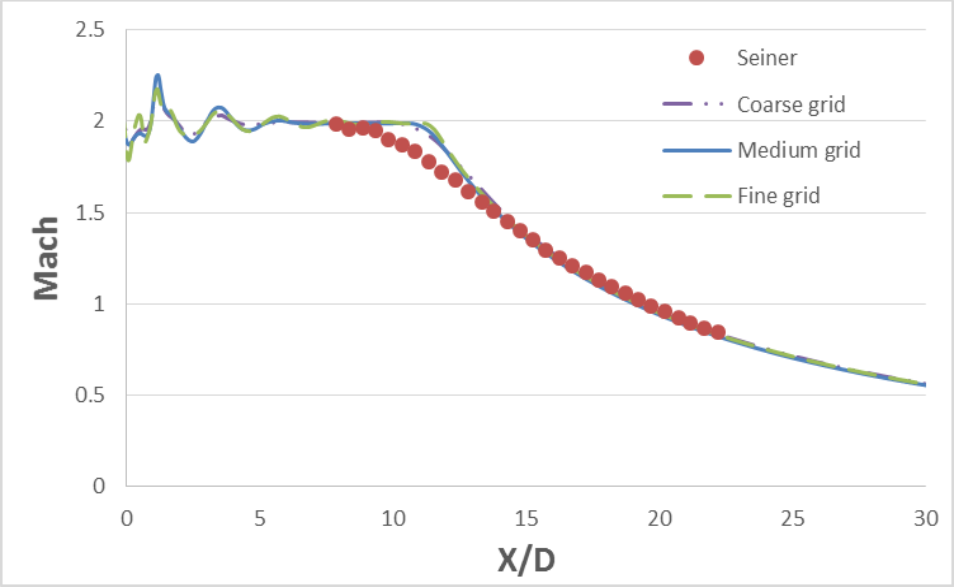
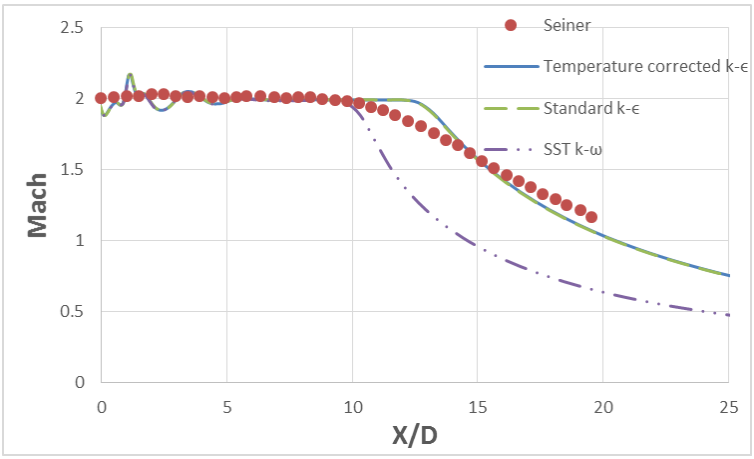
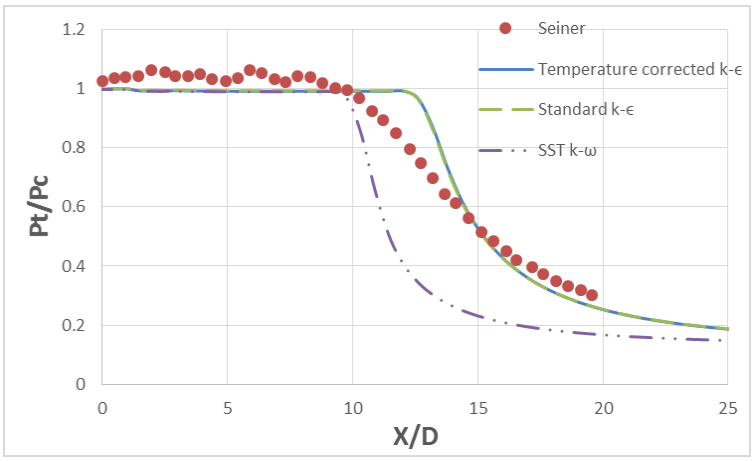


Figure 11. Comparison of computed centerline velocity using three levels of grid with experimental data at Tc = 313K

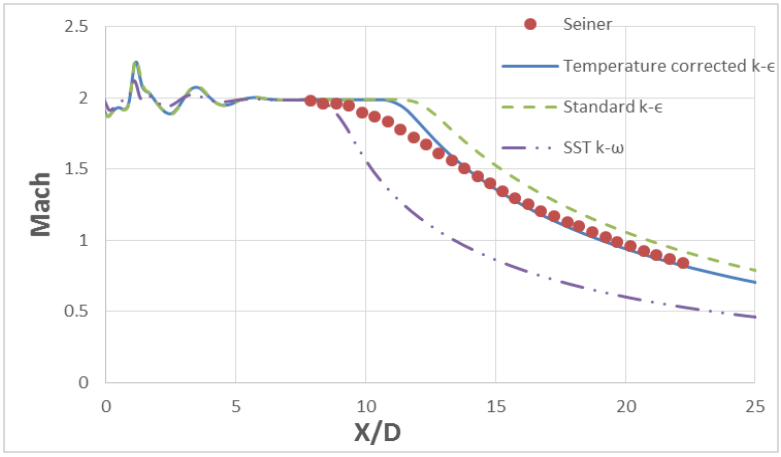


(a) Centerline Mach number

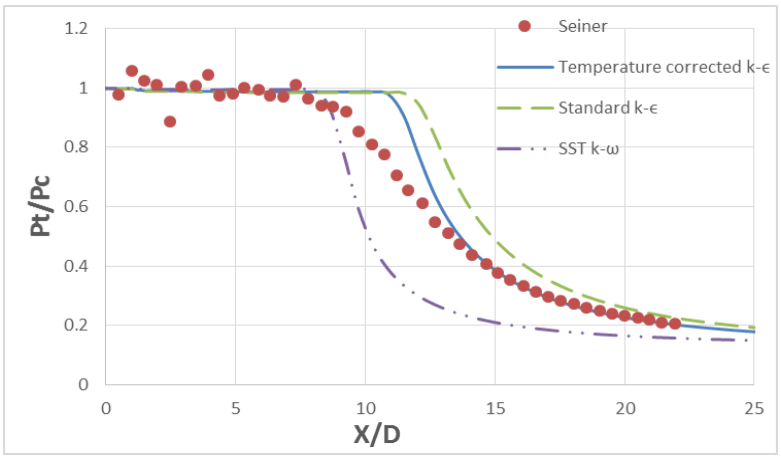


(b) Centerline total pressure

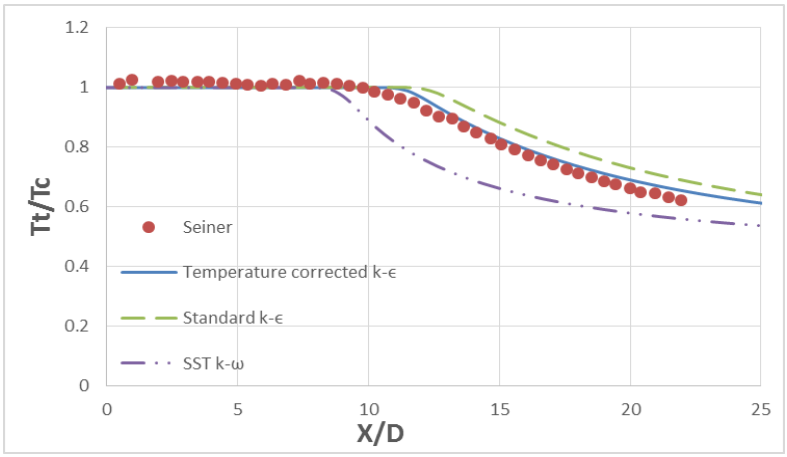
Figure 12. Comparison of computed centerline Mach number and total pressure using the three turbulence models with experimental data at Tc = 313K.



(a) Centerline Mach number

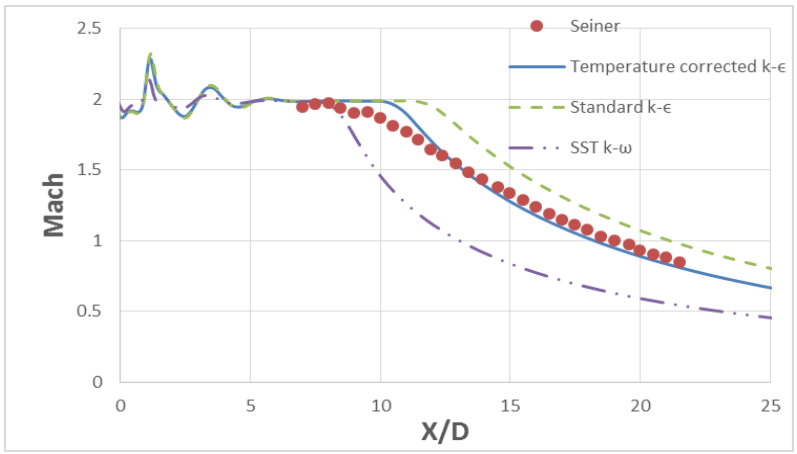


(b) Centerline total pressure

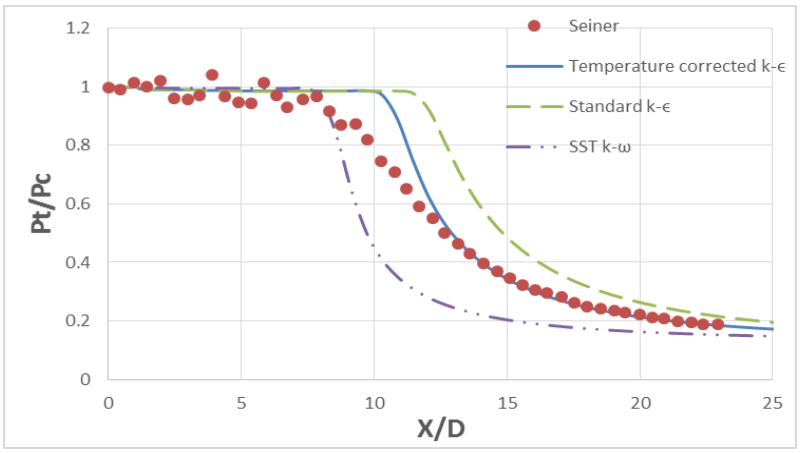


(c) Centerline total temperature

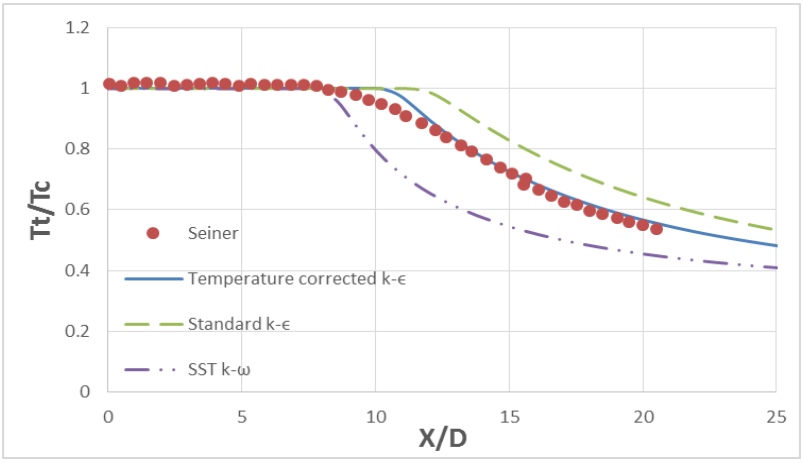
Figure 13. Comparison of computed centerline Mach number, total pressure and total temperature using the three turbulence models with experimental data at Tc = 755K.



(a) Centerline Mach number



(b) Centerline total pressure



(c) Centerline total temperature

Figure 14. Comparison of computed centerline Mach number, total pressure and total temperature using the three turbulence models with experimental data at Tc = 1116K.

2.5 Summary

The objective of this work was to verify and validate the temperature corrected *k-ε* turbulence model proposed by Abdol-Hamid et al. [7] by computing the supersonic turbulent boundary layers on a flat plate and jet flows from supersonic nozzles. The accuracy of the temperature corrected *k-ε* model was assessed by comparing the computations using the temperature corrected *k-ε* turbulence model with those obtained using the standard *k-ε* and SST *k-ω* models and the

experimental data. It was found that for supersonic flat boundary layers, all the three models gave results in good agreement with the experimental data; however for two well documented jet flows exiting from supersonic nozzles, the temperature corrected k-ε model gave the best predictions compared to the experimental data. The results from the standard k-ε model were also very close to those predicted by the temperature corrected k-ε model with somewhat larger variation with respect to the experimental data. On the other hand, the results of computations using the SST k-ω model did not match the experimental data at all especially in the core region of the jet and near the exit of the nozzle. The results presented in this chapter demonstrate that the temperature corrected k-ε model should be considered for computing the supersonic high temperature wall bounded and free shear flows. However further investigations are needed using this model by computing additional 3D complex supersonic turbulent flows.

Chapter 3: Temperature Corrected SST k-ω Model

The temperature corrected *k-ε* model gave good predictions of high temperature and high Mach number jet flows. This chapter describes the formulation of temperature corrected SST *k-ω* model following the approach for temperature corrected *k-ε* model.

3.1 Two-equation SST k-ω Turbulence Model

SST *k-ω* model is one of the most widely used two-equation models, SST *k-ω* model is considered superior to *k-ε* model in predicting viscous flow near the wall region, the ω is defined as:

$$\omega \equiv \epsilon / k \tag{13}$$

The shear stress transport (SST) *k-ω* turbulence model is a combination of *k-ε* model and standard *k-ω* model and is more accurate than the standard *k-ω* model.

The transport equations for *k* and ω are:

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = P_k - \beta^* k\omega + \frac{\partial}{\partial x_j}\left[(\upsilon + \sigma_k \upsilon_T)\frac{\partial k}{\partial x_j}\right] \tag{14}$$

$$\frac{\partial \omega}{\partial t} + U_j \frac{\partial \omega}{\partial x_j} = \alpha S^2 - \beta \omega^2 + \frac{\partial}{\partial x_j}\left[(\upsilon + \sigma_\omega \upsilon_T)\frac{\partial \omega}{\partial x_j}\right] + 2(1 - F_1)\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i} \tag{15}$$

where

$$\nu_t = C_\mu \frac{a_1 k}{\max(a_1\omega, \Omega F_2)}, \qquad \Omega = \sqrt{2W_{ij}W_{ij}}, \qquad W_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}) \tag{16}$$

$$\varphi = F_1\varphi_1 + (1 - F_1)\varphi_2 \tag{17}$$

$$\nu_t = C_\mu \frac{a_1 k}{max(a_1\omega, \Omega F_2)}, \qquad \Omega = \sqrt{2W_{ij}W_{ij}}, \qquad W_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}) \tag{18}$$

$$F_1 = \tanh(arg_1{}^4) \tag{19}$$

$$arg_1 = \min\left[\max\left(\frac{\sqrt{k}}{\beta^*\omega d}, \frac{500\upsilon}{d^2\omega}\right), \frac{4\rho\sigma_{\omega 2}k}{CD_{k\omega}d^2}\right] \tag{20}$$

3.2 Temperature Corrected SST k-ω Model

Accounting for the effect of temperature fluctuations on turbulence, the temperature correction can also be applied to SST *k-ω* model in a similar way as for the *k-ε* model. According to the description in section 2.3, to develop the temperature corrected equations, the formulation of a non-dimensional total temperature variable and a compressibility factor are needed. The non-dimensional total temperature variable can be derived using the length scale of the two-equation SST *k-ω* model given by:

$$L = \frac{k^{1/2}}{\omega} \tag{21}$$

Substitute Eq. [21] into Eq. [9], the dimensionless total temperature variable T_g is given by:

$$T_g = \frac{k^{1/2}}{\omega}\frac{|\nabla T_t|}{T_t} \tag{22}$$

The SST *k-ω* model has a well calibrated compressibility correction which affects directly the transport equations, thus the compressibility factor is not necessary to appear in temperature corrected SST *k-ω* model. After calibration against the experimental data, the temperature corrected kinematic turbulent eddy viscosity can be written as:

$$\nu_t = 0.09\left[1 + 340 \times {T_g}^3\right]\frac{k}{\omega} \tag{23}$$

However, if the standard SST *k-ω* model is used without the compressibility correction of Ref [13], the temperature kinematic turbulent eddy viscosity can be expressed as:

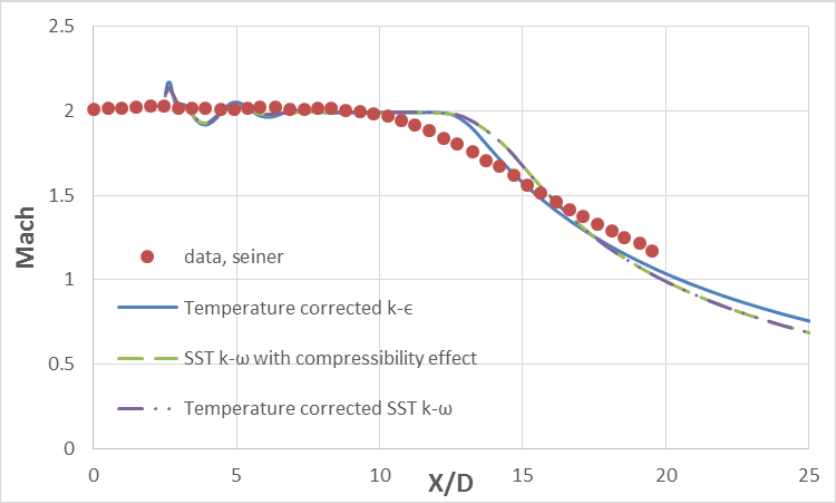
$$\nu_t = 0.09\left[1 + \frac{13.5{T_g}^3}{0.039 + f(M_\tau)}\right]\frac{k}{\omega} \tag{24}$$

Where $f(M_\tau)$ has been defined in section 2.2. It should be noticed that when the compressibility effect is small, $f(M_\tau)$ is small and Eq. [24] reduces to Eq. [23]. The two corrections Eq. [23] and Eq. [24] give similar results in simulation; the results from the Eq. [23] are presented in section 3.3.

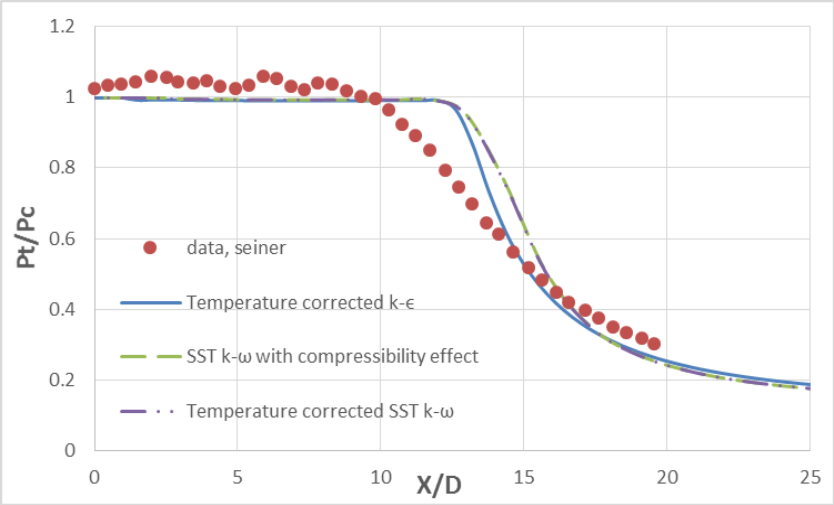
As mentioned in section 2.2, an upper bound on the value of ν_t may be required in calculations to avoid divergence of the solution.

3.3 Test Cases and Results

In this section, the computational results for jet flow exhaust from Seiner nozzle, employing the new temperature correction based on two-equation SST *k-ω* model, are presented. The computational results are compared to the temperature corrected *k-ε* model and standard SST *k-ω* model with compressibility effect.

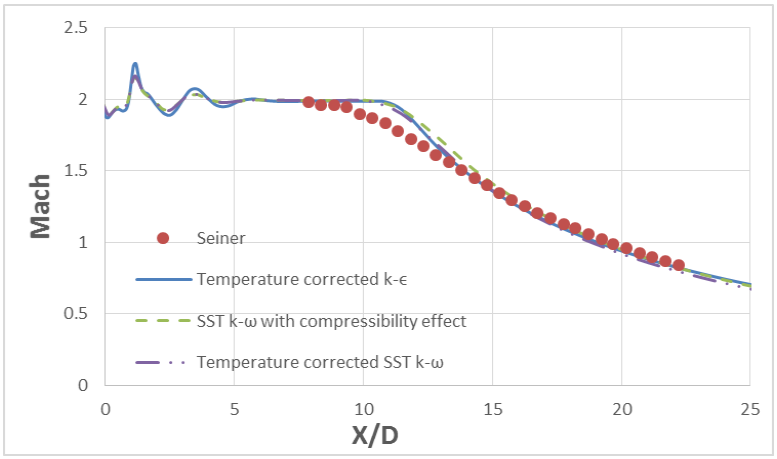


(a) Centerline Mach number

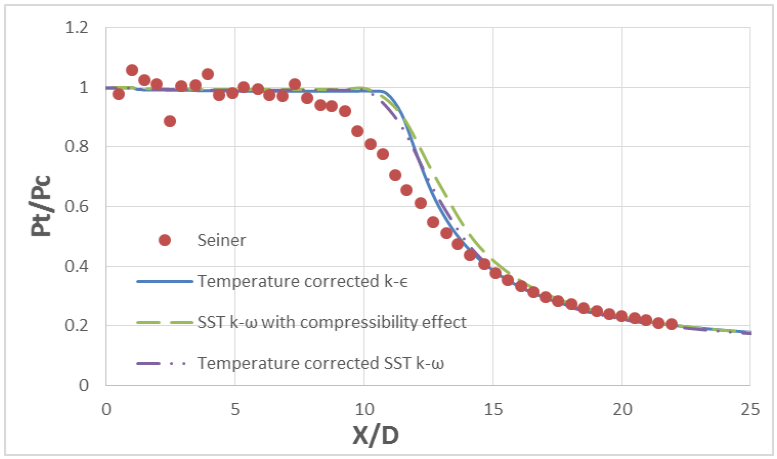


(b) Centerline total pressure

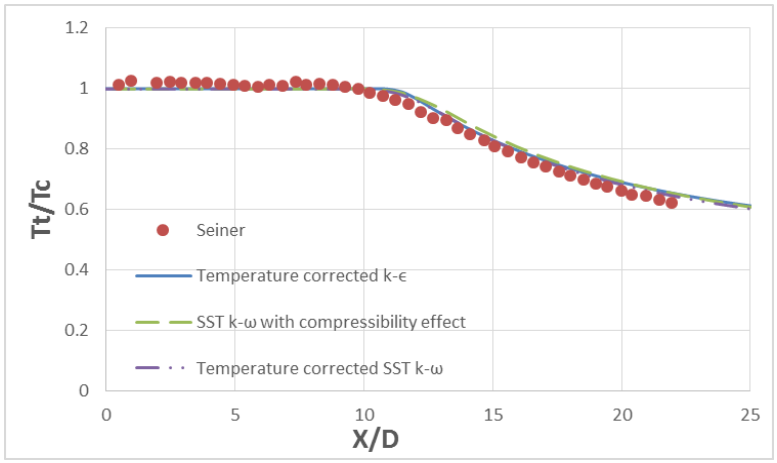
Figure 15. Comparison of computed centerline Mach number and total pressure using the three turbulence models with experimental data at Tc = 313K.



(a) Centerline Mach number

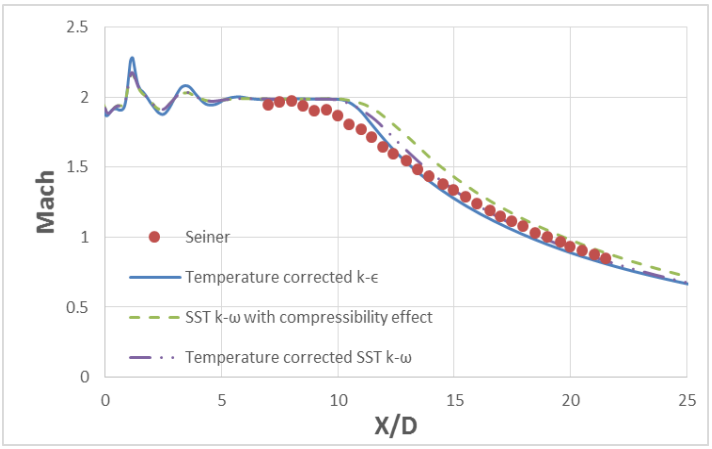


(b) Centerline total pressure

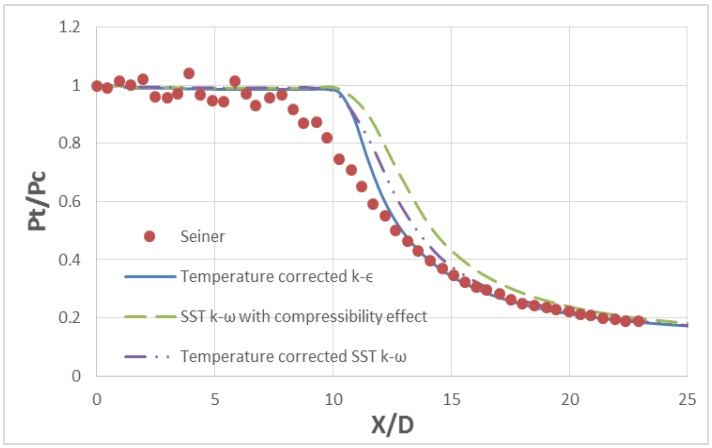


(c) Centerline total temperature

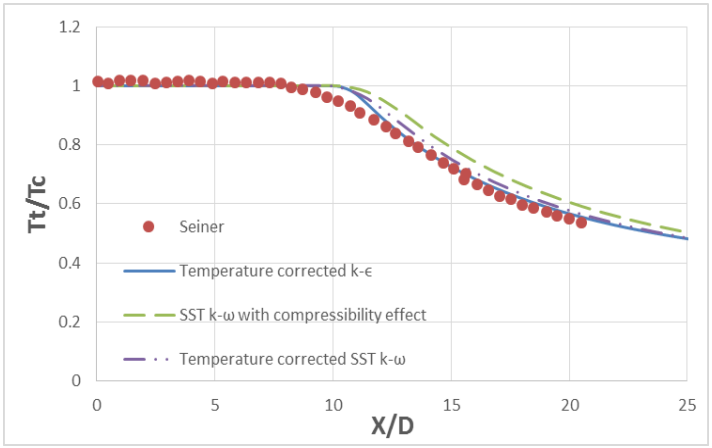
Figure 16. Comparison of computed centerline Mach number, total pressure and total temperature using the three turbulence models with experimental data at Tc = 755K.



(a) Centerline Mach number



(b) Centerline total pressure



(c) Centerline total temperature

Figure 17. Comparison of computed centerline Mach number, total pressure and total temperature using the three turbulence models with experimental data at Tc = 1116K.

3.4 Summary

The objective of this work was to verify and validate the temperature corrected SST k-ω turbulence model by computing the supersonic jet exhaust flow from Seiner nozzle. The accuracy of the temperature corrected SST k-ω model was assessed by comparing the computations using the temperature corrected SST k-ω turbulence model with those of the temperature corrected k-ε model, standard SST k-ω model with compressibility effect and the experimental data. It was found, the temperature corrected SST k-ω model gave better predictions compared to standard SST k-ω with compressibility effect. The results from the temperature corrected k-ε model gave the best prediction with respect to the experimental data. The results presented in this section show that the temperature corrected SST k-ω model improved the accuracy compared to standard SST k-ω model; however, it was not as good as temperature corrected k-ε model.

Chapter 4: Temperature Corrected WA Model

In previous two chapters, the temperature corrected *k-ε* and SST *k-ω* models show their ability to improve the accuracy in computation of high temperature high Mach number jet flows. In this chapter, the formulation of temperature corrected one-equation WA model [6] is presented along with the simulation results for Eggers and Seiner nozzle.

4.1 One-equation WA Model

WA model is a one-equation model which has exhibited with good numerical accuracy and faster computation speed for computing turbulent flows [6]. It is competitive in accuracy of the SST *k-ω* and *k-ε* model. It has been shown that this model is more accurate than the one-equation Spalart-Allmaras (SA) model. For the WA model the transport equation for modified eddy viscosity $R=k/\omega$ can be written as:

$$\frac{DR}{Dt} = \frac{\partial}{\partial x_j}\left[(\sigma_R R + \upsilon)\frac{\partial R}{\partial x_j}\right] + C_1 RS + f_1 C_{2k\omega}\frac{R}{S}\frac{\partial R}{\partial x_j}\frac{\partial S}{\partial x_j} - (1-f_1)C_{2k\epsilon}R^2\left(\frac{(\frac{\partial S}{\partial x_j})(\frac{\partial S}{\partial x_j})}{S^2}\right) \tag{25}$$

$$\nu_t = f_\mu R \tag{26}$$

The quantity *R* is defined as:

$$R \equiv \frac{k}{\omega} \tag{27}$$

4.2 Temperature Corrected Equations to WA Model

According to the description in section 2.2 and 2.3, development of temperature corrected formulation of a turbulent model requires a nondimensional total temperature gradient factor T_g derived from the length scale and a compressibility factor. In the transport equation of WA model Eq. [25], two quantities, *R* and *S*, can be used to form a length scale:

$$L = (\frac{R}{S})^{1/2} \tag{28}$$

However, since WA model has a blend of *k*-ε model and *k*-ω model, T_g should not be directly derived from the length scale, instead the switch function should be applied:

$$T_g = (\frac{\sigma_R R}{S})^{1/2} \frac{|\nabla T_t|}{T_t} \tag{29}$$

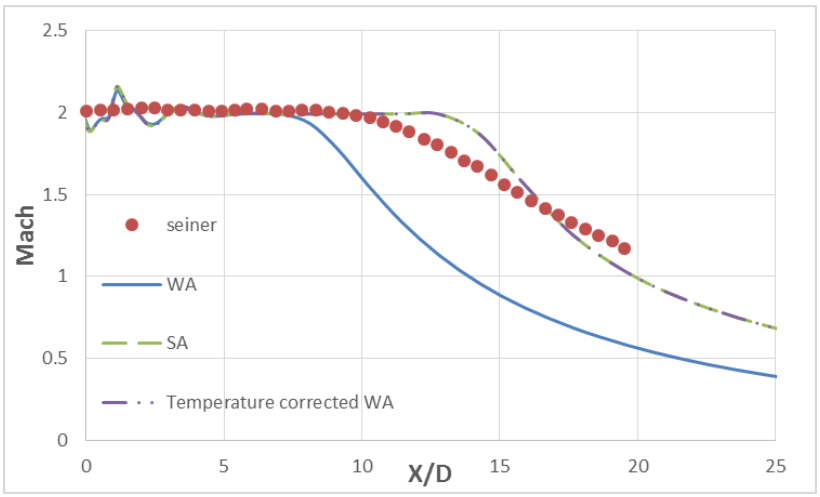
Wray and Agarwal have tested the WA model for several highly compressible flows [6]. They have demonstrated that the WA model can simulate compressible flows without the need for a compressible correction [6]. Thus, the compressibility term is not necessary for the formulation of temperature corrected equation.

After calibration against the experimental data, the temperature corrected kinematic turbulent eddy viscosity can be written as:

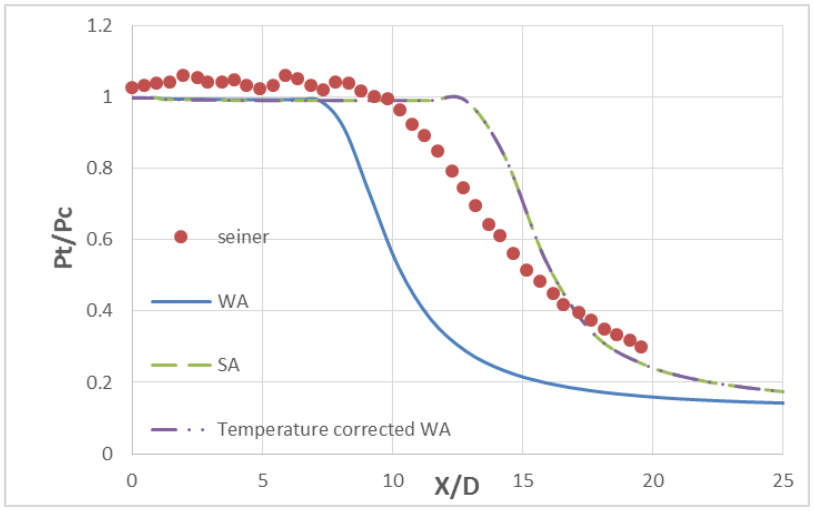
$$\nu_t = f_\mu R(1 + 18.0 \times {T_g}^3) \tag{30}$$

4.3 Test Cases and Results

In this section, the computational results for jet exhaust flow from seiner nozzle, employing the temperature corrected WA model, are presented. The computed results from temperature corrected WA model are compared to those from WA model, SA model, and experimental data.

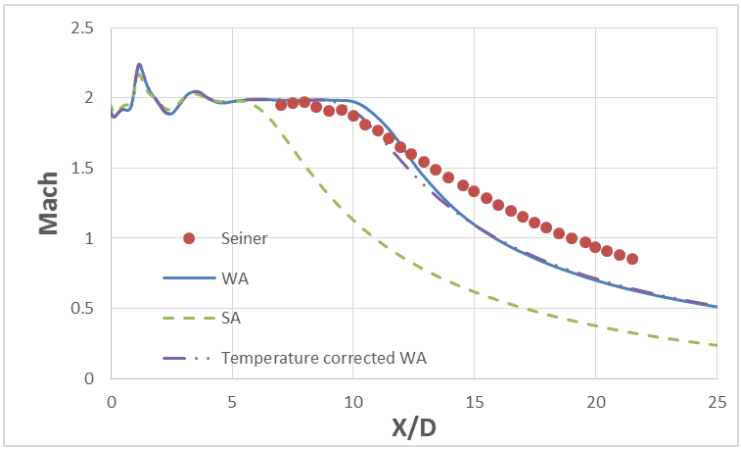


(a) Centerline Mach number

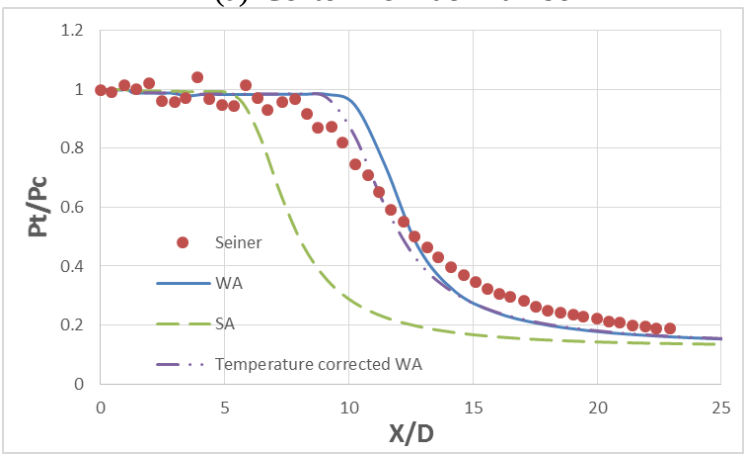


(b) Centerline total pressure

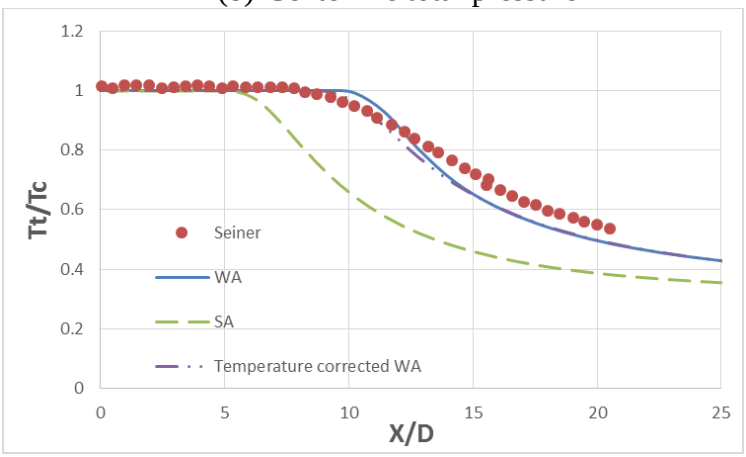
Figure 18. Comparison of computed centerline Mach number and total pressure using the three turbulence models with experimental data at Tc = 313K.



(a) Centerline Mach number

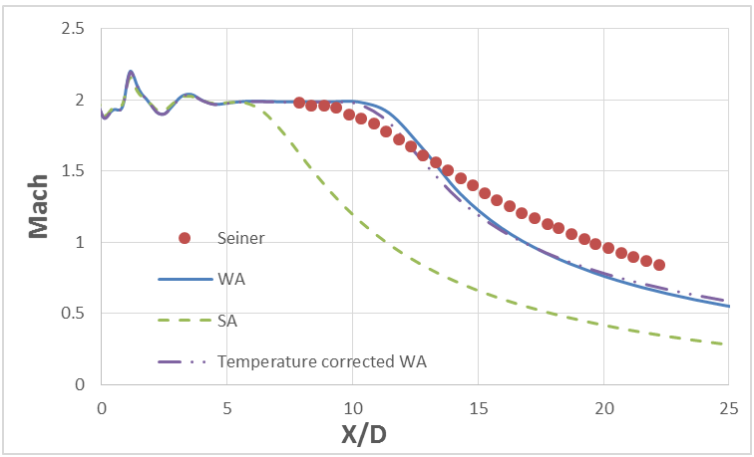


(b) Centerline total pressure

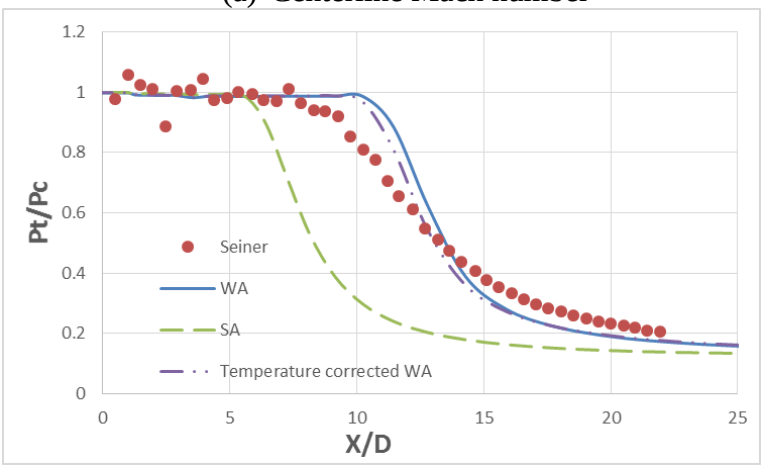


(c) Centerline total temperature

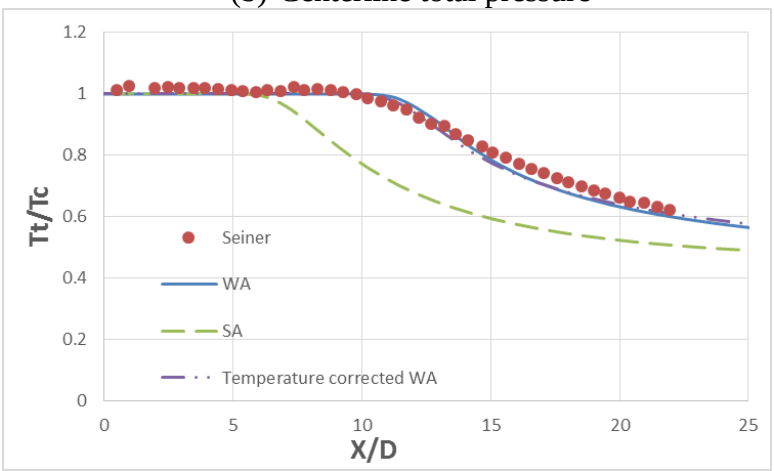
Figure 19. Comparison of computed centerline Mach number and total pressure using the three turbulence models with experimental data at Tc = 755K.



(a) Centerline Mach number



(b) Centerline total pressure



(c) Centerline total temperature

Figure 20. Comparison of computed centerline Mach number and total pressure using the three turbulence models with experimental data at Tc = 1116K.

4.4 Summary

The objective of this work was to verify and validate the temperature corrected WA model by computing the supersonic jet exhaust flow from Seiner nozzle. The accuracy of the temperature corrected WA model was assessed by comparing the computations using the temperature corrected WA turbulence model with those of the standard WA model, SA model, and the experimental data. It was found that the WA model and its temperature corrected version are not as accurate as the two-equation k-ε model and the SST k-ω model. However, the temperature corrected WA model showed better predictions compared to the standard WA model. The results of computations using the SA model did not match the experimental data at all especially in the core region of the jet and near the exit of the nozzle.

Chapter 5: Conclusion

In previous chapters, it was shown that the computations using the temperature corrected turbulence models improved the accuracy compared to the results from the original standard models. It can be concluded that the temperature corrected turbulence models are more accurate for simulating the high temperature high Mach number jet exhaust flows with large temperature fluctuations. The temperature correction method was applied to the two-equation k-ε model, two-equation SST k-ω model, and the one-equation WA model. It was demonstrated that the temperature correction methodology can be applied to any turbulence model. The temperature correction equations consist of a non-dimensional total temperature gradient factor and a compressibility factor. Compressibility factor can be eliminated if the standard turbulence model already has a compressibility correction and has the ability to predict compressible flow. The dimensionless total temperature gradient factor can be derived from the turbulence length scale based on the turbulent transport equations.

References/Bibliography/Works Cited

[1] Seiner, J. M., Ponton, M. K., Jansen, B. J. & Lagen, N. T., "The Effects of Temperature on Supersonic Jet Noise Emission," DGLR/AIAA Paper 92-02-045,1992

[2] So, R. M. C., and Sommer, T. P., "An Explicit Algebraic Heat-Flux Model for the Temperature Field," International Journal of Heat and Flow, 1995, Vol. 7, pp. 455-465.

[3] Ronki, M., and Gatski, Thomas B., "Predicting Turbulent Convective Heat Transfer in Fully Developed Duct Flows," International Journal of Heat and Flow, Vol. 22, 2001, pp. 381-392.

[4] Abe, K., Kondoh, T., and Nagano, Y., "A Two-equation Heat Transfer Model Reflecting Second-Moment Closures for Wall and Free Turbulent Flows," International Journal of Heat and Flow, Vol. 17, 1996, pp. 228-237.

[5] Thomas, R. H., Kinzie, K. W. and Pao, S. Paul, "Computational Analysis of a PylonChevron Core Nozzle Interaction," AIAA Paper 2001-2185, May 2001.

[6] Wray, T. and Agarwal, R. K., "Application of the Wray-Agarwal Model to Compressible Flows," AIAA paper 2015-3083, 45th AIAA Fluid Dynamics Conference, AIAA Aviation Forum, June 2015.

[7] Abdol-Hamid, K. S., Pao, S. P., Massey, S. J., and Elmiligui, A., "Temperature Corrected Turbulence Model for High Temperature Jet Flows," Journal of Fluids Engineering, Vol. 126, No. 5, 2004, pp. 844– 850.

[8] Eggers, J. M., "Velocity Profiles and Eddy Viscosity Distributions Downstream of a Mach 2.22 Nozzle Exhausting to Quiescent Air," NASA TN D-3601, 1966.

[9] Seiner, J. M., Dash, S. M., and Wolf, D. E., "Analysis of Turbulent Under-expanded Jets, Part II: Shock Noise Features Using SCIPVIS," AIAA Journal, Vol.23, No 5, 1985, pp. 699-677.

[10] Van Driest, E. R., "Turbulent Boundary Layers in Compressible Fluids," Journal of Spacecraft and Rockets, Vol. 40, No. 6, 2003, pp. 1012-1028, http://dx.doi.org/10.2514/2.7048.

[11] O'Donnell, R. M., "Experimental Investigation at a Mach number of 2.41 of Average Skin Friction Coefficients and Velocity Profiles for Laminar and Turbulent Boundary Layers and an Assessment of Probe Effects," NACA TN 3122, 1954

[12] Shoemaker, C. J., "Performance of Seven Semicircular Lift-Producing Nozzles," NASA TN D-2731, 1965.

[13] Rumsey, C. L., "Compressibility Considerations for k-w Turbulence Models in Hypersonic Boundary-Layer Applications," Journal of Spacecraft and Rockets, Vol. 47, No. 1, 2010, pp. 11-20.

Appendix

UDF CODE:

Temperature Correction for k-ε Model

```
#include "udf.h"


DEFINE_TURBULENT_VISCOSITY(user_mu_t, c, t)
{
        real mu_t;
        real rho = C_R(c, t);
        real k = C_K(c, t);
        real d = C_D(c, t);
        real cp = C_CP(c, t);
        real R = C_RGAS(c, t);
        real T = C_T(c, t);
        real u = C_U(c, t);
        real v = C_V(c, t);
        real mu = C_MU_L(c, t);
        real dudx = C_U_RG(c, t)[0];
        real dudy = C_U_RG(c, t)[1];
        real dvdx = C_V_RG(c, t)[0];
        real dvdy = C_V_RG(c, t)[1];
        real dTdx = (C_T_RG(c, t)[0]);
        real dTdy = (C_T_RG(c, t)[1]);
        real M_0 = 0.1;
```

```
real gamma;
real f_mu;
real T_g;
real F_M_u;
real a;
real M_tou;
real T_t;
real G_T_t;
real tp;

gamma = cp / (cp - R);

a = sqrt(gamma*R*T);
M_tou = (sqrt(2.0 * k) / a);

T_t = T + (SQR(u) + SQR(v)) / (2 * cp);

G_T_t = sqrt(SQR(dTdx + (u / cp)*dudx + (v / cp)*dvdx) + SQR(dTdy + (u / cp)*dudy + (v / cp)*dvdy));

T_g = (G_T_t*(pow(k, 1.5) / d)) / T_t;
```

```
        if (M_tou >= M_0) {
                F_M_u = SQR(M_tou) - SQR(M_0);
        }
        else {
                F_M_u = 0;
        }


        f_mu = 1 + (pow(T_g, 3.0) / (0.041 + F_M_u));

        tp = M_keCmu*rho*SQR(k)*f_mu / d;


        if (tp <= 500000 * mu) {
                mu_t = tp;
        }
        else {
                mu_t = 500000 * mu;
        }


        return mu_t;
}
```

Temperature Correction for SST k-ω Model

```
#include "udf.h"


DEFINE_TURBULENT_VISCOSITY(user_mu_t, c, t)
{
        real mu_t;
        real rho = C_R(c, t);
        real k = C_K(c, t);
      real o = C_O(c, t);
        real cp = C_CP(c, t);
        real R = C_RGAS(c, t);
        real T = C_T(c, t);
        real u = C_U(c, t);
        real v = C_V(c, t);
        real mu = C_MU_L(c, t);
        real dudx = C_U_RG(c, t)[0];
        real dudy = C_U_RG(c, t)[1];
        real dvdx = C_V_RG(c, t)[0];
        real dvdy = C_V_RG(c, t)[1];
        real dTdx = (C_T_RG(c, t)[0]);
        real dTdy = (C_T_RG(c, t)[1]);
        real M_0 = 0.1;
        real gamma;
        real f_mu;
```

```
real T_g;
real F_M_u;
real a;
real M_tou;
real T_t;
real G_T_t;
real tp;

gamma = cp / (cp - R);

a = sqrt(gamma*R*T);
M_tou = (sqrt(2.0 * k) / a);

T_t = T + (SQR(u) + SQR(v)) / (2 * cp);

G_T_t = sqrt(SQR(dTdx + (u / cp)*dudx + (v / cp)*dvdx) + SQR(dTdy + (u / cp)*dudy + (v / cp)*dvdy));

T_g = (G_T_t*(pow(k, 0.5) / o)) / T_t;

if (M_tou >= M_0) {
    F_M_u = SQR(M_tou) - SQR(M_0);
```

```
        }
        else {
                F_M_u = 0;
        }


        f_mu = 1.0 + 340 * pow(T_g, 3.0);

        tp = rho*k*f_mu/ o;


        if (tp <= 500000 * mu) {
                mu_t = tp;
        }
        else {
                mu_t = 500000 * mu;
        }


        return mu_t;
}
```

```
//Wray-Agarwal Turbulence Model

#include "udf.h"
#include "mem.h"
#include "math.h"

#define Kappa                0.41
#define C1kw          0.0829          //kwConstant
#define Sigmakw              0.72            //kwdiffusion
#define C2kw          (C1kw/Kappa/Kappa+Sigmakw) //kwConstant
#define C1keps               0.1127 //ProdConstant
#define Sigmakeps     2.0             //kepsdiffusion
#define C2keps               (C1keps/Kappa/Kappa+Sigmakeps) //1.86//kepsConstant
#define Cv1           8.541426

#define MYSMALL           1e-8
#define C_UDSI_RG(c,t,i)C_STORAGE_R_NV(c,t,SV_UDS_I(i)+SV_UDS_0_RG-
SV_UDS_0)
//#define  C_UDSI_RG(c,t,NuTilda)        C_UDSI_G(c,t,NuTilda)
//#define  C_UDSI_RG(c,t,SRM)            C_UDSI_G(c,t,SRM)

enum{
   NuTilda,
        SRM,
   N_REQUIRED_UDS,
```

```
        Eke,
        Ekw,
        Switch,
        eta,
        nu
};

DEFINE_ON_DEMAND(setnames)
{
   Set_User_Scalar_Name(NuTilda,"NuTilda");
        Set_User_Scalar_Name(SRM,"SRM");

        Set_User_Memory_Name(Eke,"Eke");
        Set_User_Memory_Name(Ekw,"Ekw");
        Set_User_Memory_Name(Switch,"Switch");
        Set_User_Memory_Name(eta,"eta");
        Set_User_Memory_Name(nu,"nu");
}

DEFINE_ON_DEMAND(initialize)
{
        //TODO add check that the data exists to avoid crash
        Domain *d;
        Thread *t;
```

```
        cell_t c;


        d = Get_Domain(1);


        //thread loop
        thread_loop_c(t,d)
        {
                //cell loop
                begin_c_loop(c,t)
                {
                        C_UDSI(c,t,NuTilda) = C_MU_T(c,t)/C_R(c,t);
                        C_UDSI(c,t,SRM) = C_STRAIN_RATE_MAG(c,t);
                }//end cell loop
                end_c_loop(c,t)
        }//end thread loop
}

//////////////////////////////////////////
///////////////// FUNCTIONS /////////////////
//////////////////////////////////////////
DEFINE_ADJUST(adjust, d)
{
        Thread *t;
        cell_t c;
```

```
if (! Data_Valid_P())
    {
        Message("\nNO DATA!");
        return;
    }
    thread_loop_c(t,d)
    {
        begin_c_loop(c,t)
        {
            //Bound NuTilda and SRM to avoid divide by zero
            C_UDSI(c,t,NuTilda) = MAX(C_UDSI(c,t,NuTilda),MYSMALL);
            C_UDSI(c,t,SRM) =
MAX(C_STRAIN_RATE_MAG(c,t),MYSMALL);


            //Compute the switch function.
            C_UDMI(c,t,nu) = C_MU_L(c,t)/C_R(c,t);
            C_UDMI(c,t,eta) =
C_WALL_DIST(c,t)*sqrt(C_UDSI(c,t,NuTilda)*C_UDSI(c,t,SRM))/(20.0*C_UDMI(c,t,nu)
);
            C_UDMI(c,t,Switch) =
(1.0+20.0*C_UDMI(c,t,eta))/(1.0+SQR(C_WALL_DIST(c,t)*MAX(sqrt(C_UDSI(c,t,NuTil
da)*C_UDSI(c,t,SRM)),1.5)/(20.0*C_UDMI(c,t,nu))));
            C_UDMI(c,t,Switch) = tanh(pow(C_UDMI(c,t,Switch), 4.0));
            C_UDMI(c,t,Switch) = MIN(C_UDMI(c,t,Switch), 0.9);
        }
```

```
        end_c_loop(c,t)
    }

    //Compute the reconstruction gradients
    Alloc_Storage_Vars(d, SV_UDSI_RG(NuTilda),SV_UDSI_G(NuTilda),SV_NULL);
    Scalar_Reconstruction(d, SV_UDS_I(NuTilda), -1, SV_UDSI_RG(NuTilda),
NULL);
    Scalar_Derivatives(d, SV_UDS_I(NuTilda), -1, SV_UDSI_G(NuTilda),
SV_UDSI_RG(NuTilda), NULL);

    Alloc_Storage_Vars(d, SV_UDSI_RG(SRM),SV_UDSI_G(SRM),SV_NULL);
    Scalar_Reconstruction(d, SV_UDS_I(SRM), -1, SV_UDSI_RG(SRM), NULL);
    Scalar_Derivatives(d, SV_UDS_I(SRM), -1, SV_UDSI_G(SRM),
SV_UDSI_RG(SRM), NULL);

    //Compute destruction terms based on reconstruction gradients
    thread_loop_c(t,d)
    {
        begin_c_loop(c,t)
        {
            C_UDMI(c,t,Eke) =
MAX(SQR(C_UDSI(c,t,NuTilda)/C_UDSI(c,t,SRM))*NV_MAG2(C_UDSI_RG(c,t,SRM)),
MYSMALL);
            C_UDMI(c,t,Ekw) =
C_UDSI(c,t,NuTilda)/C_UDSI(c,t,SRM)*NV_DOT(C_UDSI_RG(c,t,NuTilda),C_UDSI_R
G(c,t,SRM));
        }
```

```
            end_c_loop(c,t)
        }

        //Free memory
        Free_Storage_Vars(d, SV_UDSI_RG(NuTilda),SV_UDSI_G(NuTilda), SV_NULL);
        Free_Storage_Vars(d, SV_UDSI_RG(SRM),SV_UDSI_G(SRM), SV_NULL);
}

real chi(cell_t c, Thread *t)
{
        return C_UDSI(c,t,NuTilda)/(C_MU_L(c,t)/C_R(c,t));
}

real fv1_15(cell_t c, Thread *t)
{
   return pow(chi(c,t),3.0)/(pow(chi(c,t),3.0)+pow(Cv1,3.0));
}

DEFINE_TURBULENT_VISCOSITY(mut_15,c,t)
{
        real cp = C_CP(c, t);
        real r = C_RGAS(c, t);
        real T = C_T(c, t);
        real u = C_U(c, t);
```

```
        real v = C_V(c, t);
        real dudx = C_U_RG(c, t)[0];
        real dudy = C_U_RG(c, t)[1];
        real dvdx = C_V_RG(c, t)[0];
        real dvdy = C_V_RG(c, t)[1];
        real dTdx = (C_T_RG(c, t)[0]);
        real dTdy = (C_T_RG(c, t)[1]);
        real gamma;
        real f_mu;
        real T_g;
        real a;
        real T_t;
        real G_T_t;
real sig;

        gamma = cp / (cp - r);

        a = sqrt(gamma*r*T);

        sig = (C_UDMI(c, t, Switch)*(Sigmakw - Sigmakeps) + Sigmakeps);
        sig = MIN(sig, 0.8);

        T_t = T + (SQR(u) + SQR(v)) / (2 * cp);
```

```
        G_T_t = sqrt(SQR(dTdx + (u / cp)*dudx + (v / cp)*dvdx) + SQR(dTdy + (u /
cp)*dudy + (v / cp)*dvdy));


        T_g = (G_T_t*18.0*pow((sig*C_UDSI(c,t,NuTilda)/C_UDSI(c,t,SRM)), 0.5)) / T_t;


        f_mu = 1.0 + pow(T_g, 3.0);


   return C_R(c,t)*fv1_15(c,t)*C_UDSI(c,t,NuTilda)*f_mu;

}


//////////////////////////////////////////

//////////// TRANSPORT TERMS ////////////////

//////////////////////////////////////////

DEFINE_SOURCE(source_prod,c,t,dS,eqn)

{

        dS[eqn] = C_R(c,t)*(C_UDMI(c,t,Switch)*(C1kw-
C1keps)+C1keps)*C_UDSI(c,t,SRM);

   return  C_R(c,t)*(C_UDMI(c,t,Switch)*(C1kw-
C1keps)+C1keps)*C_UDSI(c,t,NuTilda)*C_UDSI(c,t,SRM);

}


DEFINE_SOURCE(source_dest,c,t,dS,eqn)

{

        return  C_R(c,t)*(

                                        C_UDMI(c,t,Switch)*C2kw*C_UDMI(c,t,Ekw)
```

```
                                                    -(1-
C_UDMI(c,t,Switch))*C2keps*C_UDMI(c,t,Eke)

                                                );

}


DEFINE_DIFFUSIVITY(diff_WAblend,c,t,eqn)

{

   return C_MU_L(c,t)+C_R(c,t)*C_UDSI(c,t,NuTilda)*(C_UDMI(c,t,Switch)*(Sigmakw-
Sigmakeps)+Sigmakeps);

}


//////////////////////////////////////////////

////////// Boundary Conditions //////////////

//////////////////////////////////////////////

DEFINE_PROFILE(inlet, t, i)

{

        face_t f;

        cell_t c0;

        Thread *t0 = t->t0;


        begin_f_loop(f,t)

        {

                c0 = F_C0(f,t);

                F_PROFILE(f,t,i) = 3*C_MU_L(c0,t0)/C_R(c0,t0);

        }
```

```
        end_f_loop(f,t)
}

DEFINE_PROFILE(outlet, t, i)
{
        //TODO add check for reversed flow, better definition of derivative.
        face_t f;
        cell_t c0;
        Thread *t0 = t->t0;
        int revFlowFaces = 0;

        begin_f_loop(f,t)
        {
                if(F_FLUX(f,t) < 0)
                {
                        revFlowFaces = revFlowFaces++;
                        c0 = F_C0(f,t);
                        F_PROFILE(f,t,i) = 3*C_MU_L(c0,t0)/C_R(c0,t0);
                }else
                {
                        c0 = F_C0(f,t);
                        F_PROFILE(f,t,i) = C_UDSI(c0,t0,NuTilda);//looks like
dNuTilda/dn=0 for orthogonal meshes
                }
        }
```

```
        end_f_loop(f,t)


        if(revFlowFaces > 0)
        {
                //Message("\nReversed flow on %i faces",revFlowFaces);
        }
}
```

Vita

Shuai Shuai

Degree

M.S Mechanical Engineering, May 2017, Washington University in St. Louis

B.S Mechanical Engineering, August 2014, Shanghai JiaoTong University

Birth of place

Wuhan, Hubei, China

Publication

Shuai Shuai, Tim Wray, and Ramesh K. Agarwal. "Turbulence Modeling of High Speed Compressible Flows",

55th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum, (AIAA 2017-0317)

May 2017