Spring 5-2019

Development of One-Equation ARSM-k-kL model and Extension of Wray-Agarwal Turbulence Model to Transitional and Rough Wall Flows

Tianshu Wen
Washington University in St. Louis

WASHINGTON UNIVERSITY IN ST. LOUIS

James McKelvey School of Engineering
Department of Mechanical Engineering and Material Science

Thesis Examination Committee:
Dr. Ramesh Agarwal, Chair
Dr. David Peters
Dr. Swami Karunamoorthy

Development of One-Equation ARSM-k-kL model and Extension of Wray-Agarwal Turbulence Model to Transitional and Rough Wall Flows

by
Tianshu Wen

A dissertation presented to the James McKelvey School of Engineering
of Washington University in St. Louis
in partial fulfillment of the requirements for the degree of
Master of Science

May 2019
St. Louis, Missouri

Table of Contents

List of Figures

List of Tables

Nomenclature

C_f	=	skin friction coefficient
C_l	=	lift coefficient
C_p	=	pressure coefficient
G	=	production of turbulent kinetic energy
L	=	turbulent length scale
L_{vk}	=	von Kármán length scale
Ma	=	Mach number
Re	=	Reynolds number
Reτ	=	friction Reynolds number
S	=	mean strain rate magnitude
S_{ij}	=	symmetric strain rate tensor
T	=	temperature
W	=	vorticity magnitude
W_{ij}	=	asymmetric vorticity rate tensor
c	=	chord length of the airfoil
d	=	distance normal to the nearest wall

k = turbulent kinetic energy

k_s = sand grain roughness height

l = length of the plate

t = time

u^+ = mean velocity normalized by the friction velocity

y = Cartesian coordinate

AOA = angle of attack

ARSM = algebraic Reynolds stress model

CFD = computational fluid dynamics

DNS = direct numerical simulation

LES = large eddy simulation

RANS = Reynolds Averaged Navier-Stokes

δ_{ij} = Kronecker Delta

κ = Karman constant

ν = kinematic viscosity

μ = bulk viscosity

μ_t = turbulent eddy viscosity

ω = dissipation rate per unit turbulent kinetic energy

ρ = density

τ_{ij} = Reynold stress tensor

Acknowledgments

First of all, I would like to thank Professor Ramesh Agarwal for his guidance and patience throughout my research. His talent, both industrial and academic, and rich CFD experience inspired and encouraged me to explore the academic world in fluid dynamics.

I would also like to thank my committee members, Dr. Peters and Dr. Karunamoorthy, for taking the time to read this thesis and attend the defense.

I would like to thank Dr. Xu Han for helping me when I first tried to use OpenFOAM. His excellent coding skills helped me a lot in implementation of turbulence models.

Finally, I would like to thank my family for their encouragement and financial support. Without their support, I would not be able to finish this thesis.

Tianshu Wen

Washington University in St. Louis

May 2019

Dedication

I would like to dedicate this thesis to my parents (Chengli Wen and Ruihong Li) for their unconditional support.

I will never succeed without their guidance, influence and encouragement.

ABSTRACT OF THE THESIS

Development of One-Equation ARSM-k-kL model and Extension of Wray-Agarwal Turbulence Model to Transitional and Rough Wall Flows

by

Tianshu Wen

Master of Science in Mechanical Engineering

Washington University in St. Louis, 2019

Research Advisor: Professor Ramesh K. Agarwal

In last five decades, Computational Fluid Dynamics (CFD) has become a mature technology and the CFD solvers are now regularly employed in the analysis and design of automobiles, aircrafts and a wide variety of other industrial applications. Despite of its wide usage, one of its building blocks, namely the 'Turbulence Modeling' still remains a pacing item in accurate computation of fluid flows; turbulence models are required in numerical simulation of turbulent flows using the Reynolds Averaged Navier-Stokes equations (RANS). Even though Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES) can provide better accuracy, the needed computing power at present is prohibitive for complex 3D applications. The goal of this research has been to develop accurate and efficient one-equation turbulence models to increase the accuracy of simulations for flow over rough wall flows and flows with mild separation. The development is based on recently proposed one-equation eddy viscosity RANS models which are known as the Wray-Agarwal (WA) model and the two-equation k-kL-ARSM model. The two proposed modified one-equation models are validated by NASA Turbulence Modeling Resource (TMR) benchmark test cases; both the models provide competitive results compared to the one-equation Spalart-Allmaras (SA) model and one-equation k-kL model.

Chapter 1: Introduction

1.1 Background and Motivation

Turbulent flow is a type of fluid motion that undergoes unsteady, and irregular fluctuations. It can be observed in everyday surroundings e.g. in smoke rising from chimney or water flowing in a river. Despite of their pervasiveness, prediction of turbulent flow still remains an unsolved problem in classical physics due to the random variations of flow variables with infinite number of length and time scales. In past few decades, Computational Fluid Dynamics (CFD) has emerged as an effective tool to obtain reasonably accurate solutions of turbulent flows encountered in many industrial applications. Even though the instantaneous flow properties in a turbulent flow are very sensitive to initial and boundary conditions, the time-averaged properties are quite regular on length and time scales of interest. With decades of effort, the CFD technology for solution of Reynolds Averaged Navier-Stokes (RANS) equations with turbulence models has been developed and is now widely used in industrial applications. To close the time-averaged Navier-Stokes or RANS equations, the Reynolds stresses or turbulent stresses are modeled; it is the so-called closure problem. Turbulence modeling is modeling of the turbulent stresses. Based on their complexity, the turbulence models in terms of strain rate tensor range from the simple algebraic or zero-equation model to full Reynolds stress model, with seven transport equations. In categorizing a turbulence model, *n-equation* model implies that *n* time-averaged partial differential equations are used to define the eddy viscosity in the turbulence model which must be added to the time-averaged continuity and momentum equations. The one-equation Spalart-Allmaras model for eddy-viscosity and two-equation k-ω (k = turbulent kinetic

energy and ω = specific turbulent dissipation rate) Shear Stress Transport (SST) model are the most widely used models in the industry. In addition, there are two approaches called the Direct Numerical Simulation (DNS) and the Large Eddy Simulation (LES) that are used for computation of turbulent flows which are more accurate compared to the RANS with turbulence models. However, both DNS and LES can currently be used for very simple applications due to their very high computation cost and CPU requirements.

1.2 Objectives

The overall objective of this research is to extend and develop one-equation turbulence models, which include extension of Wray-Agarwal (WA) model to rough wall flows and transitional flow, and development of a new one equation Algebraic Reynolds Stress Model (ARSM) model based on k-kL closure, and transitional flow prediction

The principal tasks to be accomplished are:

1. Implement surface roughness corrections in WA-2017m, WA-2018, SA models.
2. Derive a new one-equation turbulence model base on k-kL closure and algebraic Reynolds stresses.
3. Combine the WA2018 model with an algebraic intermittency term γ term for prediction of laminar-turbulent transition process.
4. Validate the newly proposed models for wide range of incompressible and compressible benchmark flows.

1.3 Outline

The goal of this research can be divided into three parts: (1) extend the WA-2017m and WA-2018 models for flow over rough surfaces; (2) to develop a new one-equation Algebraic Reynolds Stress Model based on k-kL closure to improve prediction of flow separation.; and (3) to improve the baseline WA2018 model for laminar-turbulent transition prediction by including a simple algebraic transition model.

A brief summary of each chapter and its contents is given below:

Chapter 2: Introduction to Turbulence Modeling: This chapter introduces turbulent flows and turbulence modeling. The main approach used in Computational Fluid Dynamics (CFD), namely the solution of Reynolds Averaged Navier-Stokes (RANS) equations is briefly described. The linear eddy viscosity turbulence models are explained and three turbulence models, namely the Spalart-Allmaras, k-ω Shear Stress Transport (SST) and WA2018 are introduced.

Chapter 3: Extension of Wray-Agarwal Turbulence Model for Rough Wall Flows: This chapter introduces extensions of WA 2017m and WA2018 models for flow over rough surfaces. The WA models with the rough-wall extensions are validated through several benchmark cases.

Chapter 4: Development of a One-Equation Algebraic Reynolds Stress Model based on k-kL Closure: This chapter provides the derivation of a new one-equation turbulence model based on k-kL closure. Instead of using the Boussinesq assumption, the newly proposed model utilizes algebraic Reynold stresses to improve the prediction of flows with mild separation. The new model is validated by several benchmark cases from NASA TMR [1].

Chapter 5: Development of a New Algebraic Transitional Flow Model: This chapter describes an extension of one-equation WA2018 model by developing and including an algebraic

transitional flow model. The extension is accomplished by coupling the one-equation WA model with an algebraic intermittency γ term. The new model is validated by 2D benchmark cases.

Chapter 6: Summary: This chapter summarizes the results of the thesis.

Chapter 2: Introduction to Turbulence Modeling

2.1 Turbulent Flow

Turbulent flows can be observed in our everyday surroundings, e.g. in smoke from a chimney or in water flowing down a river. Turbulent flows occur in almost all practical engineering problem of interests, for example flow past an airplane, a ship, or an automobile etc. Furthermore, turbulence can even play a role in applications that often involve nearly laminar flow e.g. blood in veins or drug delivery to the lung. Turbulence always occurs at large Reynolds numbers in both the external and internal flows.

The most commonly used definition of Turbulence was proposed by Hinze (1975) and later modified by Bradshaw (1974), which states that:

"Turbulent fluid motion is an irregular condition of flow in which the various quantities show a random variation with time and space coordinates, so that statistically distinct average values can be discerned. Turbulence has a wide range of scales."

By Fourier analysis of the time history of a turbulent flow, it can be found that time and length scales of turbulence can be represented by frequencies and wavelength, respectively. Compared to laminar flows, turbulent flows have the properties of instability, nonlinearity, vortex stretching, violent mixing etc. Turbulence is a continuum phenomenon. Due to its complex nature and properties, turbulence still remains an unsolved problem of classical physics.

2.2 Turbulence Modeling

2.2.1 Introduction

The time-dependent, three-dimensional continuity, Navier-Stokes and energy equations describe all the physics of a given turbulent flow. However, finding the exact solutions of Navier-Stokes equations even for the simplest of turbulent flow (in very simple geometries) remains an unsolvable has now become problem. Hence, other than the experiment, the numerical approach---the Computational Fluid Dynamics (CFD) has now become the state-of-the-art methodology to analyze the behavior of turbulent flows at high Reynolds numbers. There are three computational methods that have been developed in last five decades for solving the Navier-Stokes equations, namely the Direct Numerical Simulation (DNS), Large Eddy Simulation (LES) and Reynolds averaged Navier-Stokes (RANS). At present, DNS and LES are limited to very simple applications due to their very high computational cost, which cannot be sustained by industry in the product design. The most commonly used approach in industry is to utilize the RANS equations that require modeling of the turbulent stresses (also known as Reynolds stresses). In RANS equations, the turbulent stresses must be modeled to remove any reference to the fluctuation part of the velocity components. This is called the closure problem. The modeling of turbulent stresses in RANS equations is known as the turbulence modeling. Turbulence models relate the turbulent stresses with the strain tensor via an eddy-viscosity μ_t in an analogy to the Stokes' hypothesis which relates the shear stress in a laminar flow to strain via the molecular viscosity of the fluid μ. The complexity of turbulence models ranges from the simplest algebraic models or zero-equation models to the full Reynolds-stress model with seven transport equations. Using the most commonly used terminology in general, a *n-equation*

turbulence model means that *n* partial differential equations for some turbulence variables (for example turbulent kinetic energy, turbulent dissipation, turbulent length scale etc.) must be added to the time-averaged continuity and momentum equations (RANS) to model the turbulent stresses. The one-equation Spalart-Allmaras (SA) model for eddy-viscosity and the two-equation k-ω (k = turbulent kinetic energy and ω = specific dissipation rate) Shear Stress Transport (SST) model are the most commonly used models in the industry. The next two subsections describe the three models---SA, SST k-ω and a newly proposed Wray-Agarwal (WA2018) model.

2.2.2 The Spalart-Allmaras (SA) Turbulence Model

The Spalart-Allmaras (SA) one equation turbulence model is one of the most widely used models in industry. It is an eddy-viscosity model developed for applications to wall-bounded flows at high Reynolds number [2]. The SA model introduces a transport variable, $\tilde{\nu}$, which is proportional to the eddy viscosity μ_t as:

$$\mu_t = \rho \tilde{\nu} f_{v1} \tag{2.1}$$

The transport equation is derived by empiricism and arguments of dimensional analysis. Being a one-equation model, the SA model has both the good stability and efficiency; however it may lack of accuracy compared to two-equation models in some applications. $\tilde{\nu}$ in the standard SA model is given by the equation:

$$\begin{aligned}\frac{\partial \tilde{\nu}}{\partial t} + \frac{\partial u_j \tilde{\nu}}{\partial x_j} &= c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} - \left[c_{\omega 1} f_\omega - \frac{c_{b1}}{\kappa^2} f_{t2}\right]\left(\frac{\tilde{\nu}}{d}\right)^2 \\ &+ \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu + \tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right) + c_{b2}\frac{\partial \tilde{\nu}}{\partial x_i}\frac{\partial \tilde{\nu}}{\partial x_i}\right]\end{aligned} \tag{2.2}$$

The damping function f_{v1} is used to account for near wall blocking and is given by the following equation:

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \qquad \chi = \frac{\tilde{\nu}}{\nu} \tag{2.3}$$

The remaining functions are defined as follows:

$$\tilde{S} = \Omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \qquad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \tag{2.4}$$

where $\Omega = \sqrt{2W_{ij}W_{ij}}$ is the magnitude of the vorticity, and *d* is the distance from the field point to the nearest wall. W_{ij} is defined by the following equation:

$$W_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) \tag{2.5}$$

$$f_\omega = g\left[\frac{1 + c_{\omega 3}^6}{g^6 + c_{\omega 3}^6}\right]^{\frac{1}{6}}, \qquad g = r + c_{\omega 2}(r^6 - r), \qquad r = min\left[\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, 10\right] \tag{2.6}$$

$$f_{t2} = c_{t3} \exp(-c_{t4}\chi^2) \tag{2.7}$$

The model constants are:

$$c_{b1} = 0.1355, \qquad \sigma = \frac{2}{3}, \qquad c_{b2} = 0.622, \qquad \kappa = 0.41, \qquad c_{\omega 2} = 0.3$$

$$c_{\omega 3} = 2, \qquad c_{v1} = 7.1, \qquad c_{t3} = 1.2, \qquad c_{t4} = 0.5, \qquad c_{\omega 1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}$$

2.2.3 Menter's *k*-*ω* Shear Stress Transport (*k*-*ω* SST) Turbulence Model

Menter's *k*-*ω* Shear Stress Transport (SST) model is obtained by combining some features of *k*-*ε* model with *k*-*ω* model [3]. It is a widely used eddy viscosity turbulence model in industry.

The model includes the transport equations for turbulent kinetic energy k and the specific dissipation rate ω. It has been established by the researcher that Wilcox's k-ω [4] model is more accurate near the solid boundaries while the k-ε model is more accurate in the free-stream and other shear regions. The switching function F_1 allows the model to switch between the k-ω type for the near wall treatment, and k-ε type in the freestream region. This characteristic avoids the k-ω model being too sensitive to the inlet freestream turbulence properties and ensures the accuracy beyond the wall. The transport equations of SST k-ω model are given by:

$$\frac{\partial k}{\partial t} + \frac{\partial u_j k}{\partial x_j} = P - \beta^* k\omega + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_k \nu_t)\frac{\partial k}{\partial x_j}\right] \tag{2.8}$$

$$\frac{\partial \omega}{\partial t} + \frac{\partial u_j \omega}{\partial x_j} = \frac{\gamma}{\nu_t} P - \beta\omega^2 + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_\omega \nu_t)\frac{\partial \omega}{\partial x_j}\right] + 2(1 - F_1)\,\frac{\sigma_{\omega 2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} \tag{2.9}$$

The production term P is given by:

$$P = \tau_{ij}\frac{\partial u_i}{\partial x_j} \tag{2.10}$$

where shear stress term τ_{ij} is given by the Boussinesq assumption:

$$\tau_{ij} = \nu_t\left(2S_{ij} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}k\delta_{ij} \tag{2.11}$$

and the shear strain rate S is given by:

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.12}$$

The turbulent eddy viscosity is given by:

$$\nu_t = \frac{a_1 k}{\max(a_1\omega, \Omega F_2)} \tag{2.13}$$

where Ω is the magnitude of the vorticity computed by $\Omega = \sqrt{2W_{ij}W_{ij}}$.

Each of the constant is a blend of an inner (1) and outer (2) constant, given by:

$$\phi = F_1\phi_1 + (1 - F_1)\phi_2 \tag{2.14}$$

where the constants (1) and (2) are represented by ϕ_1 and ϕ_2, respectively.

The remaining functions are given by the following equations:

$$F_1 = tan\,h(arg_1^4) \tag{2.15}$$

$$arg_1 = min\left[max\left(\frac{\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{d^2\omega}\right), \frac{4\rho\sigma_{\omega 2}k}{CD_{k\omega}d^2}\right] \tag{2.16}$$

$$CD_{k\omega} = \max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}, 10^{-20}\right) \tag{2.17}$$

$$F_2 = tan\,h(arg_2^2) \tag{2.18}$$

$$arg_2 = max\left(2\frac{\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{d^2\omega}\right) \tag{2.19}$$

The model constants are given as follows:

$$\gamma_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1}\kappa^2}{\sqrt{\beta^*}}, \qquad \gamma_2 = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2}\kappa^2}{\sqrt{\beta^*}}$$

$$\sigma_{k1} = 0.85, \qquad \sigma_{\omega 1} = 0.5, \qquad \beta_1 = 0.075$$

$$\sigma_{k2} = 1.0, \qquad \sigma_{\omega 2} = 0.856, \qquad \beta_2 = 0.0828$$

$$\beta^* = 0.09, \qquad \kappa = 0.41, \qquad a_1 = 0.31$$

2.2.4 Wray-Agarwal (WA2018) Turbulence Model

The WA one equation turbulence model was first proposed by Wray and Agarwal [5-7]; it is a one-equation linear eddy viscosity model; “WA” in the model stands for the two authors’ last

names. The latest version of WA model is WA2018 developed by Han et al [6]. The WA2018 is a wall-distance-free model, which has been shown to improve the accuracy near curved surfaces. The model solves for the variable $R = k/\omega$, and its transport equation is given as follows:

$$\frac{\partial R}{\partial t} + \frac{\partial u_j R}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\sigma_R R + \nu)\frac{\partial R}{\partial x_j}\right] + C_1 RS + f_1 C_{2kw}\frac{R}{S}\frac{\partial R}{\partial x_j}\frac{\partial S}{\partial x_j} - (1 - f_1)min\left[C_{2k\omega}R^2\left(\frac{\frac{\partial S}{\partial x_j}\frac{\partial S}{\partial x_j}}{S^2}\right), C_m\frac{\partial R}{\partial x_j}\frac{\partial R}{\partial x_j}\right] \quad (2.20)$$

The turbulent eddy viscosity is given by:

$$\mu_t = \rho f_\mu R \quad (2.21)$$

where ρ is the density. S is the magnitude of the strain rate:

$$S = \sqrt{2S_{ij}S_{ij}}\,, \qquad S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad (2.22)$$

To ensure that there is no division by zero, S is bounded by:

$$S = \max(S, 10^{-16}S^{-1}) \quad (2.23)$$

The damping function f_μ is used to account for wall blocking:

$$f_\mu = \frac{\chi^3}{\chi^3 + C_\omega^3}\,, \quad \chi = \frac{R}{\nu} \quad (2.24)$$

The kinematic viscosity ν is defined as μ/ρ. The switching function f_1 is defined by:

$$f_1 = \tanh(arg_1^4), \quad arg_1 = \frac{\nu + R}{2}\frac{\eta^2}{C_\mu k\omega} \quad (2.25)$$

where

$$k = \frac{\nu_t S}{\sqrt{C_\mu}} \tag{2.26}$$

$$\omega = \frac{S}{\sqrt{C_\mu}} \tag{2.27}$$

$$\eta = S \max\left(1, \left|\frac{W}{S}\right|\right) \tag{2.28}$$

$$W = \sqrt{2W_{ij}W_{ij}}, \quad W_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) \tag{2.29}$$

where W is the magnitude of vorticity. The model constants are:

$$C_{1k\omega} = 0.0829, \quad C_{1k\varepsilon} = 0.1284$$

$$C_1 = f_1(C_{1k\omega} - C_{1k\varepsilon}) + C_{1k\varepsilon}$$

$$\sigma_{kw} = 0.72, \quad \sigma_{k\varepsilon} = 1.0$$

$$\sigma_R = f_1(\sigma_{k\omega} - \sigma_{k\varepsilon}) + \sigma_{k\varepsilon}$$

$$C_{2k\omega} = \frac{C_{1k\omega}}{\kappa^2} + \sigma_{kw}, \quad C_{2k\varepsilon} = \frac{C_{1k\varepsilon}}{\kappa^2} + \sigma_{k\varepsilon}$$

$$\kappa = 0.41, \quad C_\omega = 8.54$$

$$C_\mu = 0.09, \quad C_m = 8.0$$

Chapter 3: Extension of Wray-Agarwal Turbulence Model to Rough Wall Flows

3.1 Introduction

The analysis of the effect of surface roughness due to manufacturing, erosion or cavitation is very important in many real-world applications since roughness can significantly affect the performance of industrial products. The accurate roughness modification to a turbulence model is especially important since it can affect the computational simulation results of all industrial devices and products influenced by fluid flow; these results are important in the design and optimization of products.

This chapter extends the Wall-Distance-Free (WDF) one equation Wray-Agarwal (WA) model (WA2018) to rough wall flows. As shown by Han et al. [6], WA-WDF (WA2018) model has several advantages compared to WA2017 model [7]: (a) it is accurate and robust in nearly zero-strain rate flow field encountered in some applications and (b) the wall distance free nature of the WA model enhances its accuracy near curved surfaces [6]. Hence, to take advantage of WA2018 model, a new version of WA model that includes the effect of surface roughness is developed in this thesis. The validation and verification of WA2018-Rough includes two cases: (a) flow past a rough flat plate with various roughness heights and (b) flow past a rough S809 airfoil. It is shown that WA2018-Rough can accurately predict the flow past objects with surface roughness.

3.2 The New WA Model with Roughness Extension

3.2.1 The WA-2017 Roughness Model

A. The Original Model – WA2017

The original WA2017 turbulence model is also used in this study; it is the listed on the NASA Turbulence Modeling Resource (TMR) website [1]. The WA one-equation model solves for the variable $R = k/\omega$.

$$\frac{\partial R}{\partial t} + \frac{\partial u_j R}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\sigma_R R + \nu)\frac{\partial R}{\partial x_j}\right] + C_1 RS + f_1 C_{2kw}\frac{R}{S}\frac{\partial R}{\partial x_j}\frac{\partial S}{\partial x_j} - (1 - f_1)C_{2k\varepsilon}R^2\left(\frac{\frac{\partial S}{\partial x_j}\frac{\partial S}{\partial x_j}}{S^2}\right) \quad (3.1)$$

The turbulent eddy viscosity is given by:

$$\mu_t = \rho f_\mu R \quad (3.2)$$

where ρ is the density. S is strain given by:

$$S = \sqrt{2S_{ij}S_{ij}}\,, \qquad S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad (3.3)$$

To ensure there is no division by zero, S is bounded by:

$$S = \max(S, 10^{-16}S^{-1}) \quad (3.4)$$

The damping function f_μ is used to account for wall blocking:

$$f_\mu = \frac{\chi^3}{\chi^3 + C_\omega^3}\,, \quad \chi = \frac{R}{\nu} \quad (3.5)$$

The kinematic viscosity ν is defined as μ/ρ. The switching function f_1 is defined by:

$$f_1 = \min(\tanh(arg_1^4), 0.9), \quad arg_1 = \frac{1 + \frac{d\sqrt{RS}}{\nu}}{1 + \left[\frac{\max\left(d\sqrt{RS}, 1.5R\right)}{20\nu}\right]^2} \tag{3.6}$$

where *d* is the minimum distance to the nearest wall. The constants are defined as:

$$C_{1k\omega} = 0.0829, \quad C_{1k\varepsilon} = 0.1127$$

$$C_1 = f_1(C_{1k\omega} - C_{1k\varepsilon}) + C_{1k\varepsilon}$$

$$\sigma_{kw} = 0.72, \quad \sigma_{k\varepsilon} = 1.0$$

$$\sigma_R = f_1(\sigma_{k\omega} - \sigma_{k\varepsilon}) + \sigma_{k\varepsilon}$$

$$C_{2k\omega} = \frac{C_{1k\omega}}{\kappa^2} + \sigma_{kw}, \quad C_{2k\varepsilon} = \frac{C_{1k\varepsilon}}{\kappa^2} + \sigma_{k\varepsilon}$$

$$\kappa = 0.41, \quad C_\omega = 8.54$$

B. Roughness Modified Version of WA2017 Model

Nikuradse has shown that the idealized physical roughness can be represented by the equivalent sand grain approach with empirical correlations [8]. The basic idea to get the roughness effect on turbulent flow is to increase the eddy viscosity as a function of the roughness height near the wall. The velocity will have a normal shift in the boundary layer under fully rough surface condition. The velocity profile is given by:

$$u^+ = \frac{1}{\kappa}\ln\frac{y}{k_s} + 8.5 \tag{3.7}$$

The WA2017-Rough model follows the approach of SA-Rough model. The wall distance d is replaced by d_{new} at all occurrences of the distance d in the original WA2017 model. d_{new} is given by:

$$d_{new} = d + 0.03k_s \tag{3.8}$$

The viscous damping function, Eq. (3.5), must also be modified to get the accurate representation of viscous sublayer and buffer layer profiles. The modification is given by:

$$f_\mu = \frac{\chi^3}{\chi^3 + C_\omega^3}, \quad \chi = \frac{R}{\nu} + C_{r1}\frac{k_s}{d_{new}} \tag{3.9}$$

where $C_{r1} = 0.5$, and C_ω remains 8.54.

Since the modification of boundary condition does not give a large enough eddy viscosity near the wall, the coefficient $C_{2k\omega}$ of destruction term in k-ω is modified based on Wray and Agarwal's work [9]. It is given by:

$$(C_{2k\omega})_{new} = C_{2k\omega}\frac{d}{d_{new}} \tag{3.10}$$

Eq (3.10) is used to replace the $C_{2k\omega}$ coefficient in the original WA equation in Eq. (3.1).

3.2.2 The WA-Wall Distance Free (WDF) Roughness Model

A. The Original Wall Distance Free WA Model – WA2018

Recall Eq. (2.20), the transport equation of WA2018 model is given by:

$$\frac{\partial R}{\partial t} + \frac{\partial u_j R}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\sigma_R R + \nu)\frac{\partial R}{\partial x_j}\right] + C_1 RS + f_1 C_{2kw}\frac{R}{S}\frac{\partial R}{\partial x_j}\frac{\partial S}{\partial x_j} - (1 - f_1)min\left[C_{2k\omega}R^2\left(\frac{\frac{\partial S}{\partial x_j}\frac{\partial S}{\partial x_j}}{S^2}\right), C_m\frac{\partial R}{\partial x_j}\frac{\partial R}{\partial x_j}\right] \tag{2.20}$$

B. Roughness Modified Version of WA2018 Model

The current version of roughness modification to WA2018 is shown below:

$$k_{new} = \frac{\nu_t S}{\sqrt{C_\mu}} Cr_1 \tag{3.11}$$

where $Cr_1 = \frac{1}{1+\frac{Uks}{\nu}}$. Note that the term $\frac{Uks}{\nu}$ is a non-dimensional roughness height such that if ks $\rightarrow$0, then $Cr_1 \rightarrow 1$, and roughness k keeps the original form as in the WA2018 model. Obviously, Cr_1 is adapted to roughness condition; if the roughness height is infinitesimal, this roughness extension will perform as if the surface is smooth.

The boundary condition $R_{wall} = 0$ is replaced by an equation:

$$R_{wall} = 18133k_s^3 - 58.4k_s^2 + 0.0999k_s + 0.0000354 \tag{3.12}$$

Note that Eq. (3.12) should be set at a fixed value on the boundary after substituting the value of k_s.

3.3 Validation Cases

3.2.1 Flow past a 2D Rough Flat Plate

This is a 2D flat plate verification and validation test case from NASA Turbulence Modeling Resource (TMR) [1]. Figure 3.1 shows the boundary conditions. In this case, a two-meter-long flat plate is employed. The Mach number is Ma= 0.2 and Reynolds number at x = 1m is $Re_L = 5 \times 10^6$. A velocity boundary condition of 66.3 m/s at inlet is used in this case.

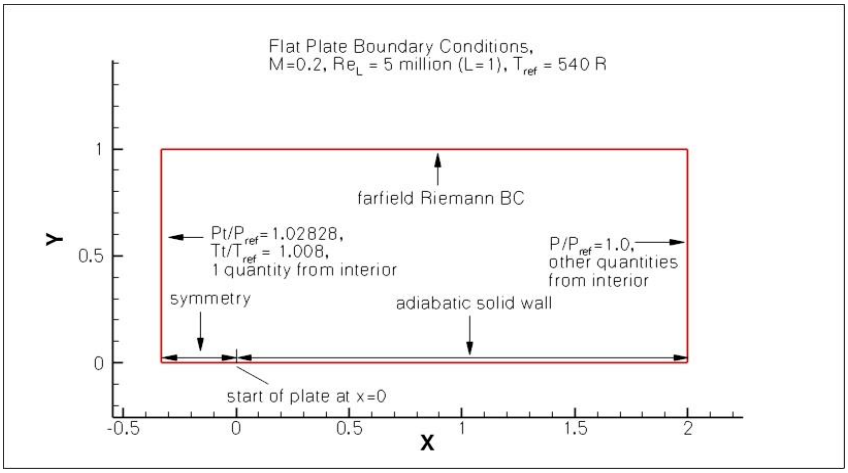

Figure 3.1 Flat plate geometry and boundary conditions [1].

Since Spalart-Allmaras (SA) model is also one of the most widely used one equation turbulence model in aerodynamics, computations from WA–Rough model are also compared with SA-Rough model. The results from the two turbulence models are compared with a semi-empirical equation for the skin friction coefficient C_f on a rough flat plate. Based on Mills and Hang's work [10], the following equation is accurate within 1 percent of experimental values when $150 < x/k_s < 1.5 \times 10^7$:

$$C_f = \left(3.476 + 0.707 \ln \frac{x}{k_s}\right)^{-2.46} \tag{3.13}$$

Figure 3.2 shows the comparison of computed results obtained by WA2018-Rough model, WA2017-Rough model, SA-Rough model and Eq. (3.13). As the sand grain roughness height k_s increases, the error in results obtained from each model increases. When k_s is as small as 0.00025m, the flat plate has very small roughness, therefore the three turbulence models accurately predict the skin friction coefficient C_f. For $k_s = 0.0005$m, the SA-Rough model's predictions are more accurate compared to those from WA2017-Rough and WA2018-Rough models at the leading edge, especially in the range $0 \leq x \leq 0.4$m. When $x > 0.4$m, the two WA models show a better agreement with Eq. (3.13). For $k_s = 0.0010$m, WA2018-Rough model

shows the best agreement overall among the three models, while the SA-Rough model still has the best agreement in a very limited range near the leading edge ($x \leq 0.4$m). At this high level of roughness, it is obvious that WA2017-Rough model cannot have the same result as WA2018-Rough model in the range $x \leq 0.4$m, but the two WA-Rough models are still much better overall than the SA-Rough model. For $k_s = 0.0015$m, WA2018-Rough model gives good result near the leading edge, and has the best agreement near the trailing edge of the flat plate. The overall results from WA2018-Rough model are most accurate compared to the results from WA2017-Rough and SA-Rough models.

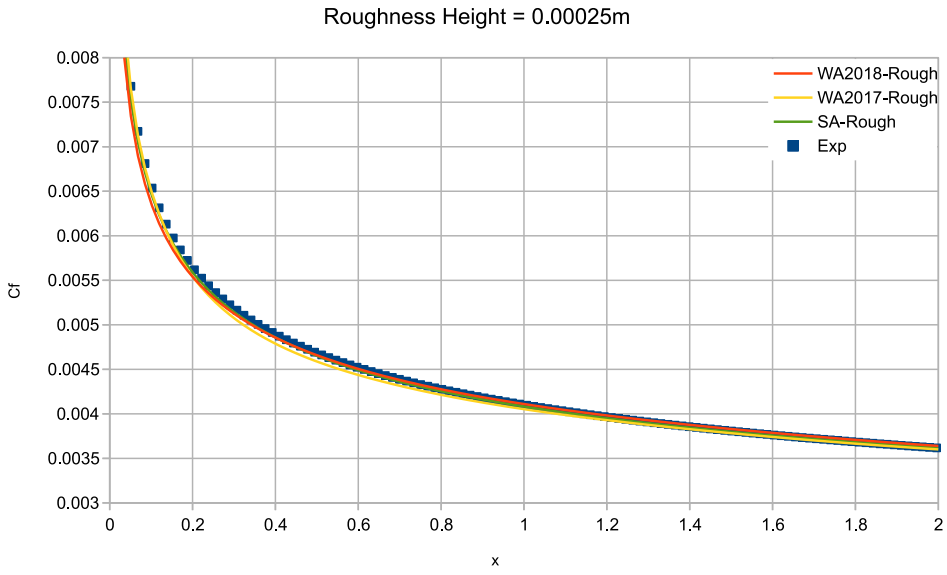

Figure 3.2 (a) Comparison of C_f for k_s=0.00025 m.

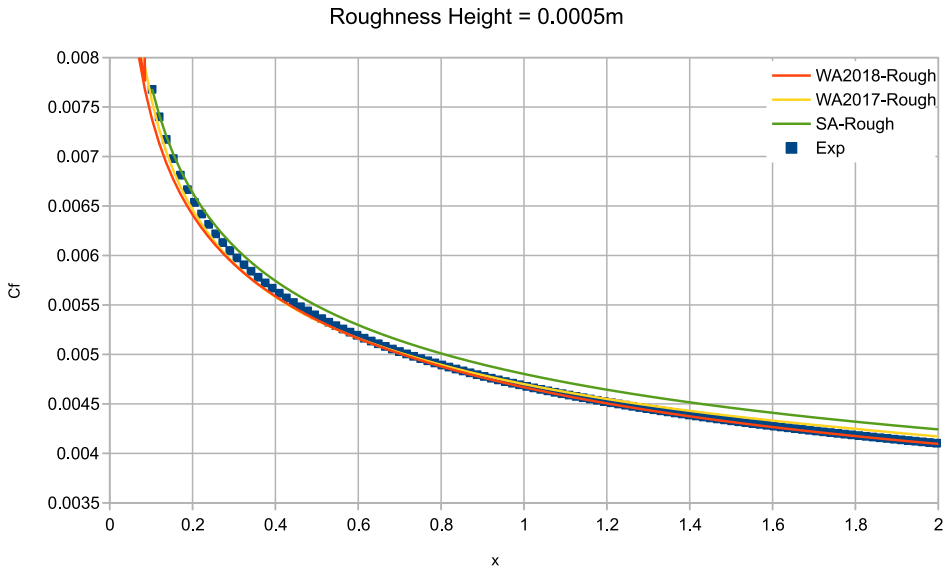

Figure 3.2 (b) Comparison of C_f for $k_s = 0.0005$ m.

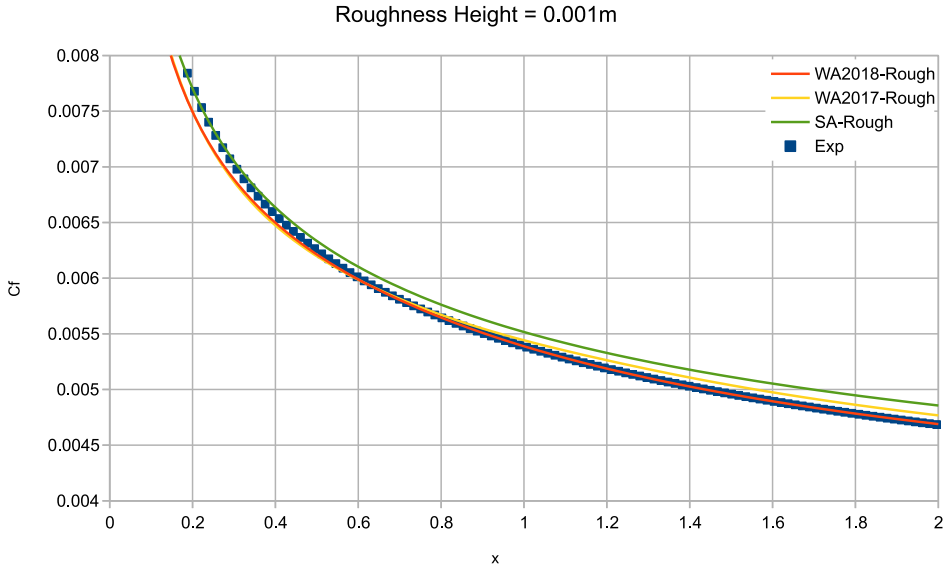

Figure 3.2 (c) Comparison of C_f for $k_s = 0.0010$ m.

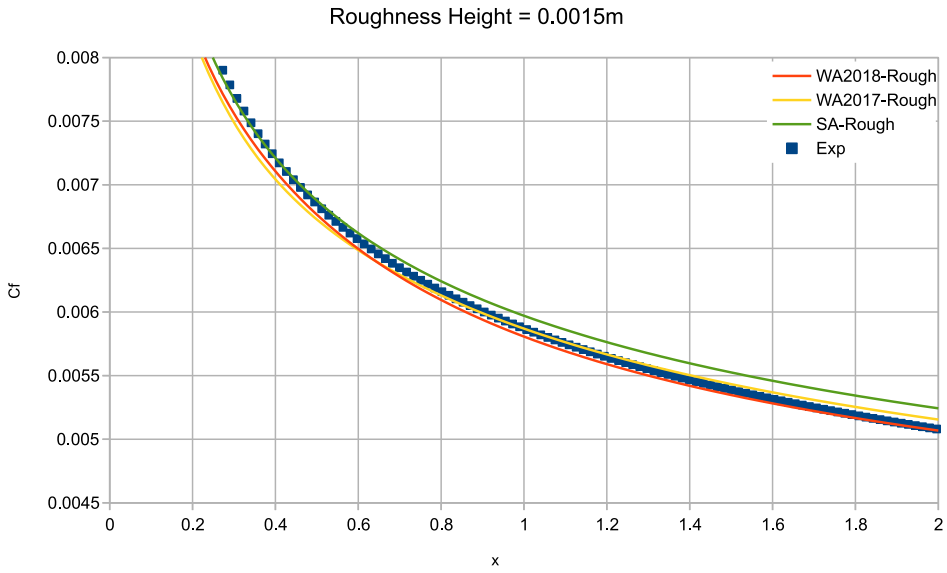

Figure 3.2 (d) Comparison C_f for $k_s = 0.0015$ m.

Figure 3.2 Comparison of C_f for three turbulence models with roughness.

3.2.3 Flow past a 2D Rough S809 Airfoil

The second validation case is that of flow over a rough S809 airfoil, which is commonly used on wind turbine blades. The working environment for a wind turbine may be harsh, and as a consequence the surface of the turbine blades may become rough due to erosion, sand grits and cavitation. The computation results are compared using the SA-Rough model, WA2018-Rough model, and WA2017-Rough model are compared with the experimental data collected by Ramsay of Ohio State University [11]. In this case, the chord length Reynolds number is 1 million. Based on Ramsay's work, the standard #40 lapidary grit is chosen to obtain a relationship between the roughness height and chord length of k_s/c=0.0019. Figure 3.3 shows the comparison of pressure coefficient between experimental and computational data. The results using the three turbulence models depict very similar behavior for pressure coefficient prediction,

showing a very small drop in Cp at the leading edge which may be improved by using a finer mesh or a better-defined geometry of S809 airfoil.

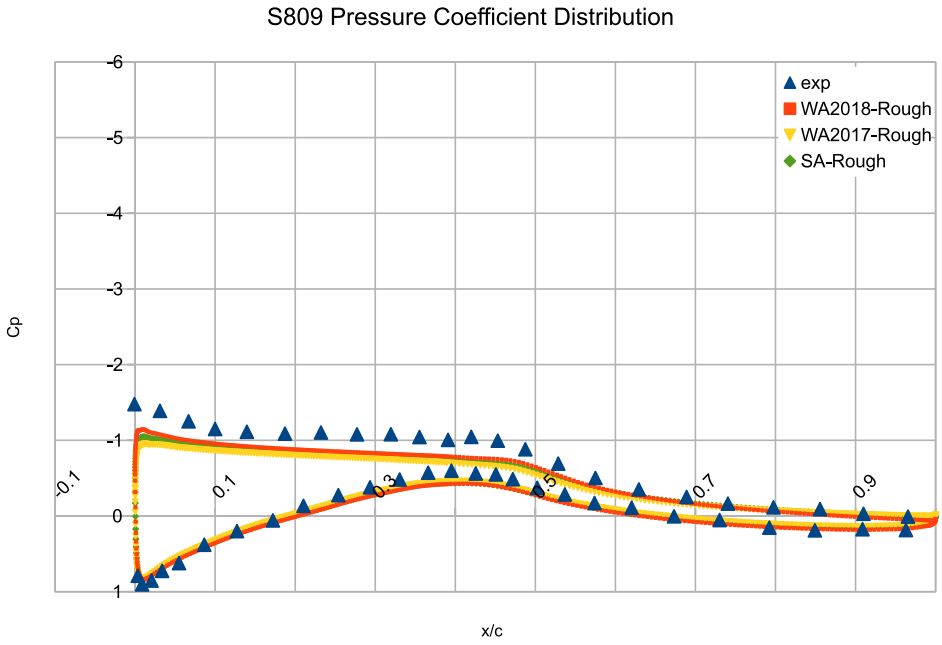

Figure 3.3 Comparison of computed and experimental Cp on rough S809 airfoil at α=6.1°.

Figure 3.4 shows the variation in computed lift coefficient with angle of attack for a smooth S809 airfoil and its comparison with experimental data. The results in Figure 3.4 are quite reasonable since all the three models show a quasi-linear relationship between the angle of attack (AOA) and lift coefficient when AOA is below 10°. White [12] has stated that an airfoil will have a stall when the AOA is about 10° when the flow separation may occur, and the theory fails to predict the lift coefficient. Figure 3.5 shows the variation in computed lift coefficient with angle of attack for a rough S809 airfoil and its comparison with experimental data. It can be seen that both WA2017-Rough and WA2018-Rough model fail to predict the experimental data while the SA-Rough model performs reasonably well. According to Wray and Agarwal [9], the WA model also require a laminar-turbulent transition model to accurately predict the C_l for AOA > 8°.

The overall performance of the three models for rough S809 case is not as good as for the smooth S809.

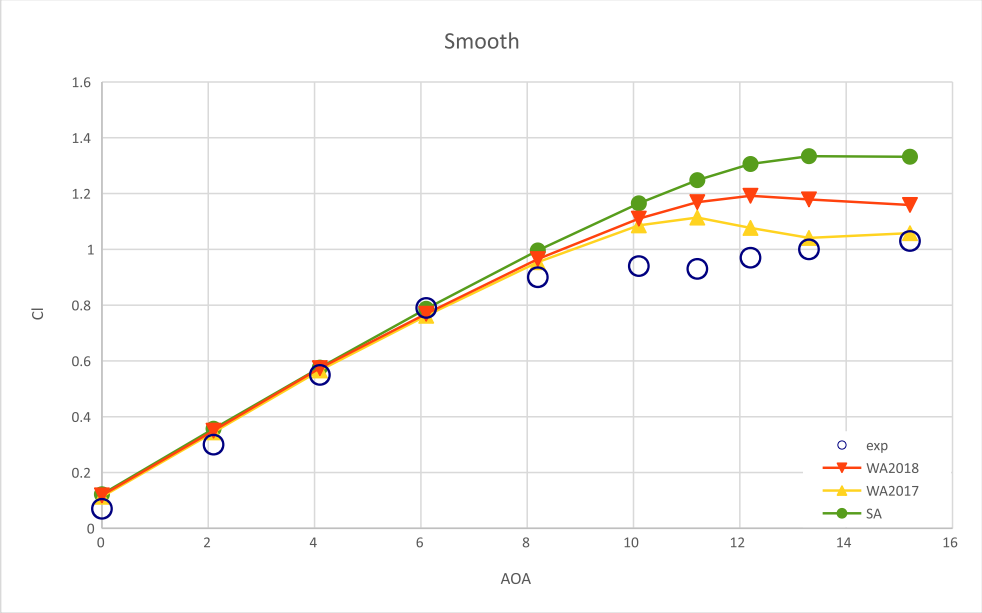

Figure 3.4 Comparison of computations with three turbulence models and experimental data for smooth S809 airfoil.

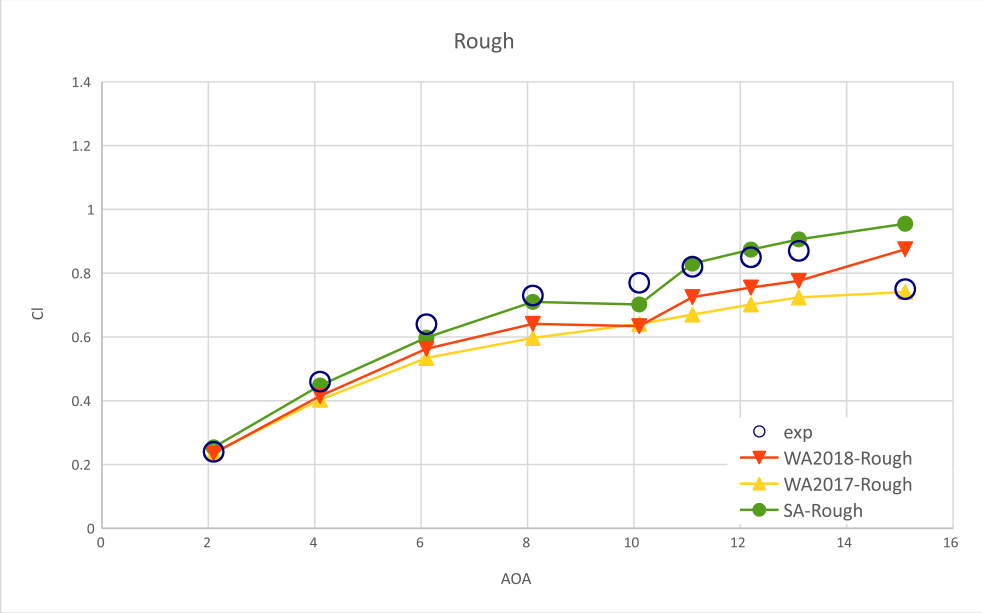

Figure 3.5 Comparison of computations with three turbulence models and experimental data for rough S809 airfoil.

3.2.4 Flow in a 2D Rough Wall Channel

The last validation case is the turbulent flow in a 2D rough-wall channel. The computational results from SA-Rough and WA2018-Rough models are compared to the experimental data collected by Saleh [13]. The channel has a cross section of 600*50 mm and a length of 2m. The channel cross section has an aspect ratio of 12.6:1 to ensure the two-dimensionality of the flow at the middle cross-section along the width of 600mm. Top and bottom surfaces of the channel are roughened by emery papers which have an average roughness height of 0.00125m. The computations are performed at two different inlet velocities which correspond to Reynolds numbers of 32,322 and 46,613 based on the channel height.

Figures 3.6 and 3.7 show the comparison of computed results and experimental data for fully developed turbulent flow in the channel at two different Reynolds numbers. WA2018-Rough model has overall better agreement with the experimental data than WA2017-Rough and SA-Rough model. The WA2018-Rough model matches the experimental data more closely in the region near the centerline of the channel. In the near wall region, WA2017-Rough and SA-Rough model under-predict the velocity compared to WA2018-Rough model; however all three models fail to match the experimental data.

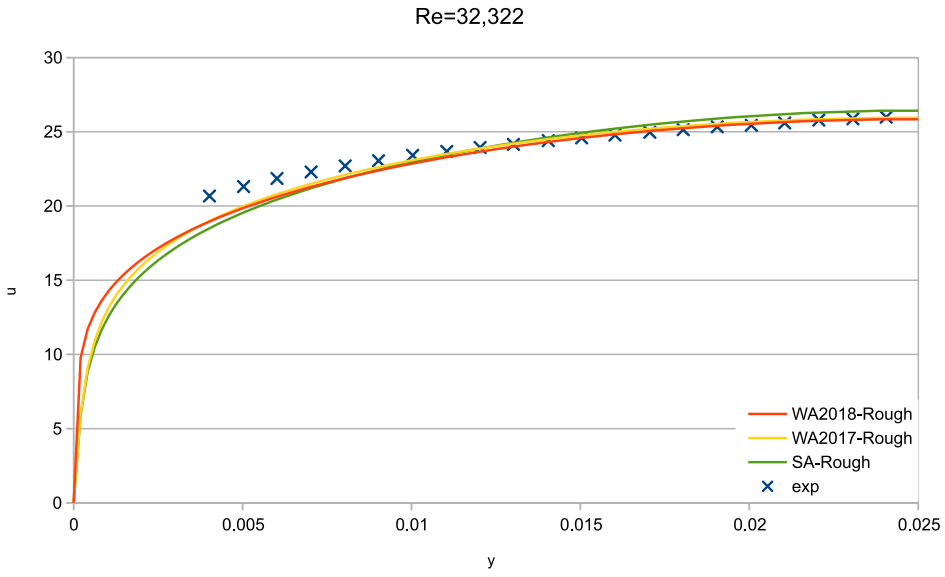

Figure 3.6 Comparison of computed and experimental velocity profiles for fully developed turbulent flow in a rough channel at Re = 32,322.

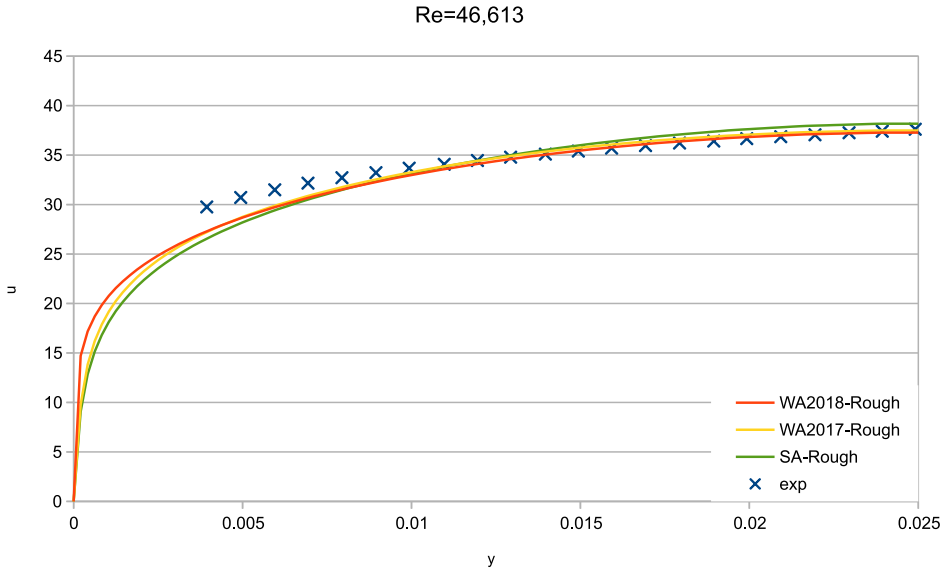

Figure 3.7 Comparison of computed and experimental velocity profiles for fully developed turbulent flow in a rough channel at Re = 46,613.

Chapter 4: Development of a One-Equation Algebraic Reynolds Stress Model based on k-kL Closure

4.1 Introduction

The current version of two-equation k-kL model in the literature utilizes Abdol-Hamid's closure based on Menter's modification to Rotta's two-equation model [14]. Rotta's formulation shows that the higher order velocity derivatives in the source terms in *kL* equation improve the accuracy of simulation of unsteady flows, which is an important feature of this model [15]. Beyond the Boussinesq approximation for eddy-viscosity, Abdol-Hamid's work shows that using an Algebraic Reynolds Stress Model (ARSM) to determine the Reynolds stress tensor would further improve the computational results for separated and corner flows [16].

In the present work, the one equation k-kL-ARSM is derived from the two-equation k-kL-ARSM model of Abdol-Hamid [16] by employing additional assumptions by Bradshaw et al. [17] and Townsend [18]. The derived model is validated by several benchmark incompressible flow cases listed on NASA Turbulence Modeling Recourse (TMR) website. Computational results are compared with experimental data and other one-equation models. The proposed one-equation k-kL-ARSM model is employed to simulate flow past a flat plate, flow in 2D channel, flow over a 2D hump, flow past a backward-facing step, and flow in an asymmetric plane diffuser. The proposed one-equation k-kL-ARSM model shows good agreements with the experimental data, and it is even better in some cases compared to the SA and WA models.

4.2 The k-kL-ARSM Model

The one equation k-kL-ARSM is derived from Abdol-Hamid's two equation k-kL-ARSM turbulence model [16]. To simplify the derivation process, the equations start with the boundary layer coordinates (*x*-streamwise coordinate, *y*- normal to the boundary layer) and the complete form of the model is given at the end of this section. The two-equation k-kL model in boundary layer coordinates can be written as:

$$\frac{Dk}{Dt} = G - C_{\mu}^{\frac{3}{4}} \frac{k^{\frac{5}{2}}}{kL} - 1.5\upsilon \frac{k}{d^2} + \frac{\partial}{\partial y}\left((\nu + \sigma_k \nu_t)\frac{\partial k}{\partial y}\right) \tag{4.1}$$

$$\frac{D(kL)}{Dt} = C_{\phi 1} G_{\phi} \frac{kL}{k} - C_{\phi 2} k^{\frac{3}{2}} - 6 f_{\phi} \upsilon \frac{kL}{d^2} + \frac{\partial}{\partial y}\left((\nu + \sigma_{\phi} \nu_t)\frac{\partial kL}{\partial y}\right) \tag{4.2}$$

where G and G_{ϕ} are the production terms for k and kL, respectively. Unlike the original k-kL model, the production terms P_k and P_{kL} here are different. The production term, P_{kL} is limited by strain rate, *S*, and linear turbulence viscosity μ_t^L as described below:

$$G = \frac{\tau_{ij}}{\rho}\frac{\partial u_i}{\partial y} \tag{4.3}$$

$$G_{\emptyset} = \frac{P_{kL}}{\rho} = \max(G, \nu_t^L S^2) = \max\left(G, -\frac{a_1^2 \nu_t S^2}{\alpha}\right) \tag{4.4}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial y} + \frac{\partial u_j}{\partial x}\right) \tag{4.5}$$

The turbulent viscosity is given by:

$$\nu_t = -\frac{\alpha}{C_{\mu}^{\frac{3}{4}}} \frac{kL}{k^{\frac{1}{2}}} \tag{4.6}$$

To derive the one-equation k-kL model, we can express the time derivative of the eddy viscosity in terms of the time derivative of *k* and *kL* as:

$$\frac{D\nu_t}{Dt} = -\frac{a_1}{S}\left(\frac{\alpha S^{\frac{1}{2}}}{a_1^2 \nu_t^{\frac{1}{2}}}\frac{D(kL)}{Dt} - \frac{1}{2}\frac{Dk}{Dt}\right) \tag{4.7}$$

To obtain a transport equation with only one independent scalar value, one more relation among ν_t, kL and k is required. The relationship is based on lot of experimental data and has been proposed by Bradshaw et al. [17] and Townsend [18] as:

$$\nu_t \left|\frac{\partial u}{\partial y}\right| = C_\mu^{\frac{1}{2}} k \tag{4.8}$$

In general, the absolute value of streamwise velocity gradient along the normal direction $\left|\frac{\partial u}{\partial y}\right|$ can be replaced by an invariant value *S* [19], and the value of $\frac{\partial v}{\partial x}$ can be neglected.

$$\left|\frac{\partial u}{\partial y}\right| \to S \tag{4.9}$$

Hence, *k* can be expressed as:

$$k = \frac{\nu_t S}{a_1} \tag{4.10}$$

and *kL* is given by:

$$kL = -\frac{a_1 \nu_t^{\frac{3}{2}} S^{\frac{1}{2}}}{\alpha} \tag{4.11}$$

By combining Eq. (4.7) with Eqs. (4.1), (4.2), (4.6), (4.8) and (4.9), the one-equation k-kL-ARSM model can be derived as:

$$\begin{aligned}\frac{D\nu_t}{Dt} = {} & \left(C_{\phi 1} G_\phi - \frac{G}{2}\right)\frac{a_1}{S} + \left(\frac{C_{\phi 2}}{a_1^{5/2}} - \frac{1}{2a_1}\right)\alpha\, \nu_t S + \frac{\nu \nu_t (\frac{3}{4} - 6f_\phi)}{d^2} + \left(2\sigma_\phi - \frac{3}{2}\sigma_k\right)\frac{\nu_t}{S}\frac{\partial \nu_t}{\partial x_i}\frac{\partial S}{\partial x_i} \\ & + \left(\frac{9}{4}\sigma_\phi - \frac{1}{2}\sigma_k\right)\frac{\partial \nu_t}{\partial x_i}\frac{\partial \nu_t}{\partial x_i} + \left(\frac{3}{2}\sigma_\phi - \frac{1}{2}\sigma_k\right)\nu_t \frac{\partial^2 \nu_t}{\partial y^2} + \left(\frac{1}{2}\sigma_\phi - \frac{1}{2}\sigma_k\right)\nu_t^2\, S^{-1}\frac{\partial^2 S}{\partial y^2} \\ & + \frac{2\sigma_\phi \nu_t^2}{\alpha^2}\frac{\partial \alpha}{\partial x_i}\frac{\partial \alpha}{\partial x_i} - \frac{\sigma_\phi \nu_t^2}{4S^2}\frac{\partial S}{\partial x_i}\frac{\partial S}{\partial x_i} - \frac{4\sigma_\phi \nu_t}{\alpha}\frac{\partial \alpha}{\partial x_i}\frac{\partial \nu_t}{\partial x_i} - \frac{\sigma_\phi \nu_t^2}{\alpha S}\frac{\partial \alpha}{\partial x_i}\frac{\partial S}{\partial x_i} - \frac{\sigma_\phi \nu_t^2}{\alpha}\frac{\partial^2 \alpha}{\partial y^2}\end{aligned} \tag{4.12}$$

The diffusive coefficients in *k* and *kL* equations are assumed to be equal, and this assumption helps us to simplify and calibrate the coefficient of the one-equation k-kL-ARSM model. The diffusive coefficients are calibrated as follows:

$$\sigma_\phi = \sigma_k = \sigma = 0.7 \tag{4.13}$$

In addition, to avoid singularity in the von Karman length-scale when $S \to 0$, especially for channel flow near the centerline region where *S* will be extremely small, and to ensure convergence, Eq. (4.12) can be rewritten as:

$$\frac{D\nu_t}{Dt} = \left(C_{\phi 1} G_{\emptyset 1} - \frac{1}{2} G\right)\frac{a_1}{S} + \left(\frac{C_{\phi 2}}{a_1^{5/2}} - \frac{1}{2} a_1^{-1}\right)\alpha \nu_t S + \frac{\nu \nu_t \left(\frac{3}{4} - 6 f_\phi\right)}{d^2} + \frac{\sigma \nu_t}{2S}\frac{\partial \nu_t}{\partial x_i}\frac{\partial S}{\partial x_i} + \frac{3\sigma}{4}\frac{\partial \nu_t}{\partial x_i}\frac{\partial \nu_t}{\partial x_i} + 2\sigma C_4 E_{BB} \tanh\left(\frac{E_{k\alpha}}{C_4 E_{BB}}\right) -$$

$$\frac{4\sigma \nu_t}{\alpha}\frac{\partial \alpha}{\partial x_i}\frac{\partial \nu_t}{\partial x_i} - \frac{\sigma \nu_t^2}{\alpha S}\frac{\partial \alpha}{\partial x_i}\frac{\partial S}{\partial x_i} - \frac{\sigma \nu_t^2}{\alpha}\frac{\partial^2 \alpha}{\partial x_i^2} - \frac{1}{4}\sigma C_3 E_{BB} \tanh\left(\frac{E_{kkl}}{C_3 E_{BB}}\right) + \frac{\partial}{\partial x_j}\left((\sigma \nu_t + \nu)\frac{\partial \nu_t}{\partial x_j}\right) \tag{4.14}$$

where

$$C_{\phi 1} = (\zeta_1 - \zeta_2 \left(\frac{-C_\mu \sqrt{\nu_t}}{\alpha \sqrt{S} L_{\nu k}}\right)^2)$$

$$C_{\phi 2} = \zeta_3$$

$$f_\phi = \frac{1 + C_{d1}\xi}{1 + \xi^4}$$

$$L_{\nu k} = \kappa \left|\frac{U'}{U''}\right|$$

$$U' = \sqrt{2 S_{ij} S_{ij}}$$

$$U'' = \sqrt{(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2})^2 + (\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2})^2 + (\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2})^2}$$

$$\xi = \frac{d\sqrt{0.3\frac{\nu_t S}{a_1}}}{20\nu}$$

$$E_{BB} = \frac{\partial \nu_t}{\partial x_i}\frac{\partial \nu_t}{\partial x_i}$$

$$E_{k\alpha} = \frac{\partial \alpha}{\partial x_i}\frac{\partial \alpha}{\partial x_i}\frac{{\nu_t}^2}{\alpha^2}$$

$$E_{kkl} = \frac{\partial S}{\partial x_i}\frac{\partial S}{\partial x_i}\frac{{\nu_t}^2}{S^2}$$

The following limiter on L_{vk} is employed:

$$L_{vk,min} \le L_{vk} \le L_{vk,max}$$

where

$$L_{vk,min} = -\frac{a_1^2 \nu_t^{1/2}}{C_{11}\alpha S^{1/2}}$$

$$L_{vk,max} = C_{12}\kappa d f_p$$

$$f_p = min[max\left(-\frac{C_{r1}G_{\emptyset}a_1^2}{\alpha \nu_t S^2}, 0.25\right), 1.0]$$

The constants in the model are:

$$\zeta_1 = 1.4 \qquad \zeta_2 = 0.97 \qquad \zeta_3 = 0.137 \qquad \kappa = 0.41 \qquad a_1 = \sqrt{C_{\mu}} = 0.3$$

$$C_{11} = 10.0 \qquad C_{12} = 1.3 \qquad C_{d1} = 4.7 \qquad C_3 = C_4 = 7.0 \qquad \sigma = 0.7 \qquad C_{r1} = 0.5$$

Instead of using the Boussinesq assumption, the turbulent stress term is given by:

$$\tau_{ij} = \tau_{ij}^{ARSM} = -\rho k\left[\sum a_{ij} + \frac{2}{3}\delta_{ij}\right] = -\rho k\left[\sum \beta_{\lambda} T^{(\lambda)} + \frac{2}{3}\delta_{ij}\right] \qquad (4.15)$$

where $T^{(\lambda)}$ is a group of second order tensors that are function of strain and vorticity rates, *S* and *W*, respectively. a_{ij} is a symmetric traceless second order tensor, which is dependent on $T^{(\lambda)}$:

$$T^{(1)} = \left[S^* - \tfrac{1}{3}tr\{S^*\}I\right], \qquad T^{(2)} = \left[S^{*2} - \tfrac{1}{3}tr\{S^{*2}\}I\right]$$

$$T^{(3)} = \left[W^{*2} - \tfrac{1}{3}tr\{W^{*2}\}I\right], \qquad T^{(4)} = [S^*W^* - W^*S^*]$$

$$T^{(5)} = [S^{*2}W^{*} - W^{*}S^{*2}], \qquad T^{(6)} = \left[S^{*}W^{*2} - W^{*2}S^{*} - \frac{2}{3}tr\{S^{*}W^{*2}\}I\right]$$

$$T^{(7)} = \left[S^{*2}W^{*2} + W^{*2}S^{*2} - \frac{2}{3}tr\{S^{*2}W^{*2}\}I\right], \; T^{(8)} = [W^{*}S^{*}W^{*2} - W^{*2}S^{*}W^{*}]$$

$$T^{(9)} = [W^{*}S^{*}W^{*2} - W^{*2}S^{*}W^{*}], \qquad T^{(10)} = [W^{*}S^{*2}W^{*2} - W^{*2}S^{*2}W^{*}]$$

The k-kL-ARSM model utilizes three tensors of $T^{(\lambda)}(\lambda = 1, 2, and\ 4)$ as follows:

$$\tau_{ij}^{ARSM} = -\rho k\left(\beta_1 T^{(1)} + \beta_2 T^{(2)} + \beta_4 T^{(4)} + \frac{2}{3}\delta_{ij}\right) \tag{4.16}$$

where $T^{(1)}$ is the linear part of the model, and $T^{(2)}$ and $T^{(4)}$ are the nonlinear anisotropic terms. *I* is the identity matrix. The β_λ coefficients are:

$$\beta_1 = -2C_\mu^* = 2\alpha, \qquad \beta_2 = -2a_4a_3\beta_1, \qquad \beta_4 = a_4a_2\beta_1$$

In the model, C_μ^* is limited to be no smaller than 0.0005. α is the root of the cubic equation:

$$\alpha^3 + p\alpha^2 + q\alpha + r = 0 \tag{4.17}$$

where coefficients in $T^{(\lambda)}$ are:

$$a_1 = \frac{1}{2}\left(\frac{4}{3} - C_2\right), \qquad a_2 = \frac{1}{2}(2 - C_4)$$

$$a_3 = \frac{1}{2}(2 - C_3), \qquad a_4 = [\gamma_1^* - 2\alpha\gamma_0^*\eta^2]^{-1}$$

The following definitions and constants are also used. Note that the coefficient a_1 above is the coefficient in $T^{(\lambda)}$ and is not computed from $\sqrt{C_\mu}$.

$$\tau = -\frac{1.51a_1}{\alpha S}$$

$$W_{ij}^* = \tau W_{ij}, \qquad S_{ij}^* = \tau S_{ij}$$

$$\eta^2 = \{S^{*2}\}, \qquad \gamma_0^* = \frac{C_1^1}{2}, \qquad \gamma_1^* = \frac{C_1^0}{2} + \frac{C_{\varepsilon 2} - C_{\varepsilon 1}}{C_{\varepsilon 1} - 1}$$

$$C_{\varepsilon 1} = 1.44, \qquad C_{\varepsilon 2} = 1.83$$

$$C_1^1 = 1.8, \qquad C_1^0 = 3.4$$

$$C_2 = 0.36, \qquad C_3 = 1.25, \qquad C_4 = 0.6$$

$$p = -\frac{\gamma_1^*}{\eta^2 \gamma_0^*}$$

$$q = \frac{1}{(2\eta^2\gamma_0^*)^2}(\gamma_1^{*2} - 2\eta^2\gamma_0^* a_1 - \frac{2}{3}\eta^2 a_3^2 - 2R^2\eta^2 a_2^2)$$

$$r = \frac{\gamma_1^* a_1}{(2\eta^2\gamma_0^*)^2}$$

$$\{W^{*2}\} = -W_{ij}^* W_{ij}^*, \qquad R^2 = -\frac{\{W^{*2}\}}{\{S^{*2}\}}$$

The root with the lowest real part was chosen from this cubic equation. If $\eta^2 < 10^{-6}$, then

$$\alpha = -\frac{\gamma_1^* a_1}{\gamma_1^{*2} - 2\{W^{*2}\}a_2^2}$$

Otherwise, define:

$$a = q - \frac{p^2}{3}, \qquad b = \frac{1}{27}(2p^3 - 9pq + 27r), \qquad d = \frac{b^2}{4} + \frac{a^3}{27}$$

If $d > 0$,

$$t_1 = \left(-\frac{b}{2} + \sqrt{d}\right)^{\frac{1}{3}}, \qquad t_1 = \left(-\frac{b}{2} - \sqrt{d}\right)^{\frac{1}{3}}$$

$$\alpha = \min(-\frac{p}{3} + t_1 + t_2, -\frac{p}{3} - \frac{t_1}{2} - \frac{t_2}{2})$$

else if $d \leq 0$,

$$\theta = \cos^{-1}\left(-\frac{b}{2\sqrt{-\frac{a^3}{27}}}\right)$$

$$\alpha = \min(t_1, t_2, t_3)$$

where

$$t_1 = -\frac{p}{3} + 2\sqrt{-\frac{a}{3}}\cos\left(\frac{\theta}{3}\right)$$

$$t_2 = -\frac{p}{3} + 2\sqrt{-\frac{a}{3}}\cos\left(\frac{2\pi}{3} + \frac{\theta}{3}\right)$$

$$t_3 = -\frac{p}{3} + 2\sqrt{-\frac{a}{3}}\cos\left(\frac{4\pi}{3} + \frac{\theta}{3}\right)$$

Note that all the terms having *k* and *kL* in the original two-equation k-kL-ARSM model are replaced by using Eq. (4.10) and (4.11).

The boundary conditions at the wall is

$$(\nu_t)_{wall} = 0$$

In the far field region, boundary condition is recommended to be:

$$(\nu_t)_{farfield} = 3\nu_\infty : \nu_t : 5\nu_\infty$$

4.3 Validation Cases

The one-equation k-kL-ARSM was validated by computing several benchmark test cases on NASA TMR's website which include flow past a flat plate, flow in a 2D channel, flow over a 2D hump, flow past a backward-facing step, flow past a curved backward-facing step, flow in an asymmetric plane diffuser and supersonic flow in a square duct. The computational results are compared with the model predictions from one-equation k-kL, WA2018 and SA models, available DNS results or experimental data for each case.

4.3.1 Zero Pressure Gradient Boundary-Layer Flow Past a Flat Plate

A low-speed zero pressure gradient turbulent flow past a flat plate is the first and foremost case need to validate a turbulence model. The experimental results are given by Wieghardt and

Tillman [20]. A uniform inlet velocity of U_{in}=68.6m/s (based on flow parameters in Figure 4.1) was used as fully turbulent inlet flow condition.

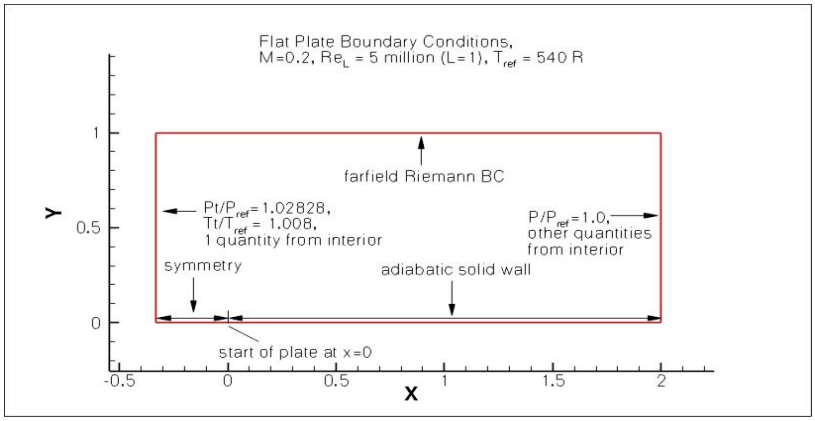

Figure 4.1 Boundary conditions for the flat plate case [1].

Figure. 4.2 shows the skin friction coefficient C_f computed using the one-equation k-kL-ARSM model, SA model and experimental data. The Reynolds number in x direction Re_x, is computed form Eq. (4.18):

$$\mathrm{Re_x} = \frac{U_\infty \mathrm{x}}{\nu_\infty} \tag{4.18}$$

In the region when Re_x is less than approximate 5 million, it is obvious that the one-equation k-kL-ARSM has the best agreement with the experimental data. The overall performance of four turbulence models are reasonably good and close as expected.

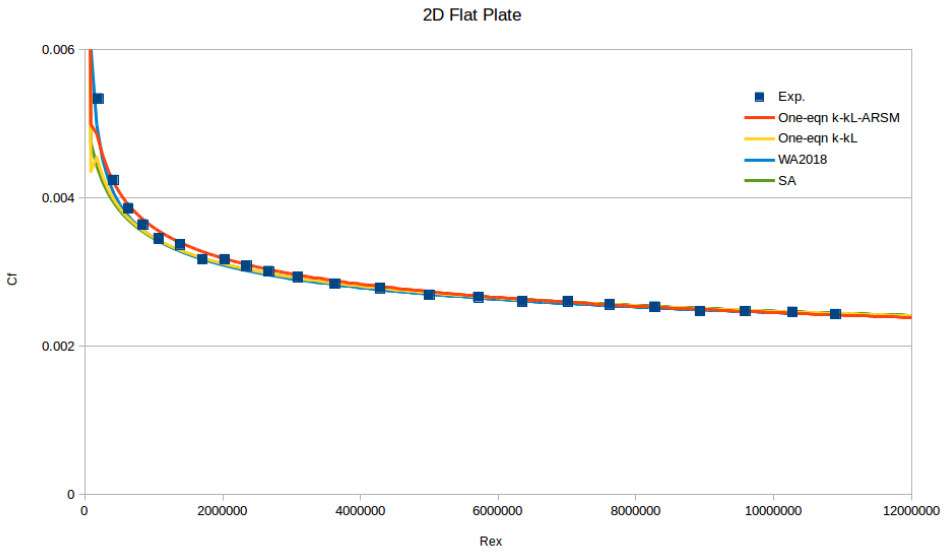

Figure 4.2 Skin friction vs. Re_x for turbulent boundary layer flow past a flat plate.

4.3.2 Flow in a 2D Channel at Different Reynolds Numbers

Another basic validation case is the fully developed turbulent channel flow. For the channel flow, three friction Reynolds numbers of 182, 1000 and 5200 were selected to validate the one-equation k-kL-ARSM model. Fully developed velocity profiles were compared with SA, WA2018 and one-equation k-kL models and the DNS data by Lee and Moser [21]. Figures. 4.3-4.8 show the fully developed velocity profiles and near wall velocity profiles for friction Reynolds numbers of 182, 1000, and 5200, respectively. Overall, the four models have good agreement with the DNS data. It is obvious that the SA model has the best results for the near wall velocity profiles, especially in the log layer. The one-equation k-kL-ARSM has reasonably good results for fully developed velocity profile in large y+ region. It is better than the one-equation k-kL model when Re_τ is small. However, with increasing friction Reynolds number, the one-equation k-kL model has better results than WA2018 and the k-kL-ARSM model. The new k-kL-ARSM model may need to include an extra elliptic partial differential equation, which is also known as elliptic relaxation, to improve the near wall velocity profile in log layer [22].

Excellent improvement in the near wall region velocity profile using elliptic blending has been shown by several researchers.

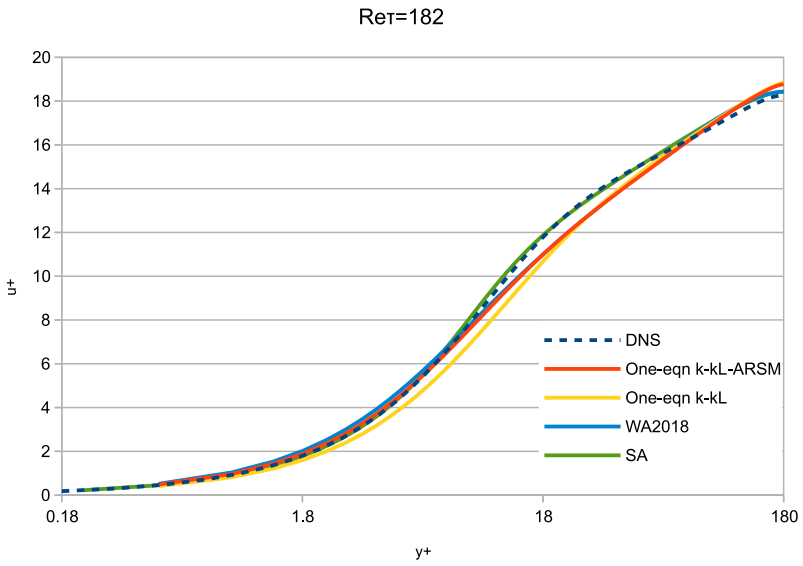

Figure 4.3 Near wall velocity profile in the channel at Reτ = 182.

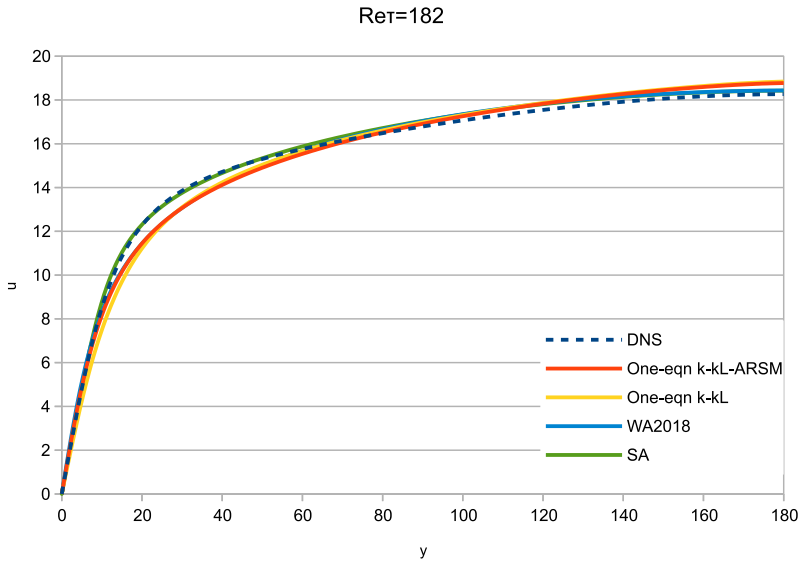

Figure 4.4 Velocity profile in the channel at Reτ = 182.

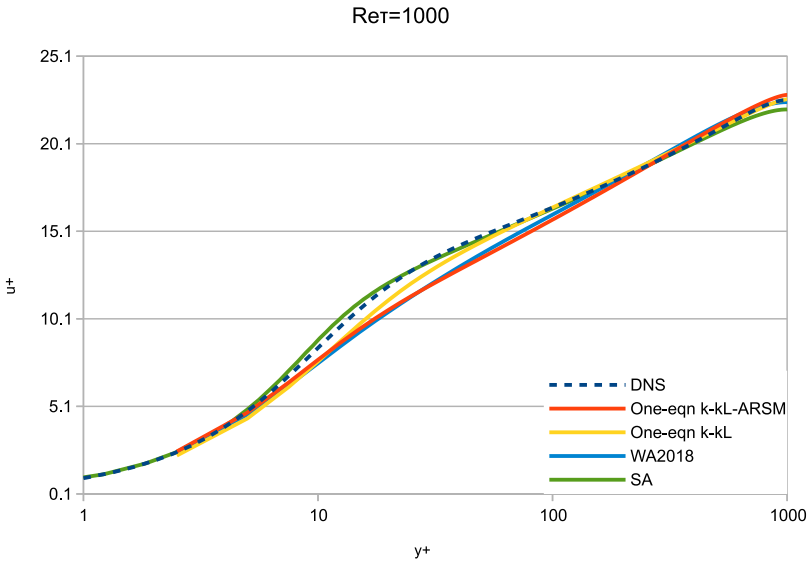

Figure 4.5 Near Wall velocity profile in the channel at Reτ = 1000.

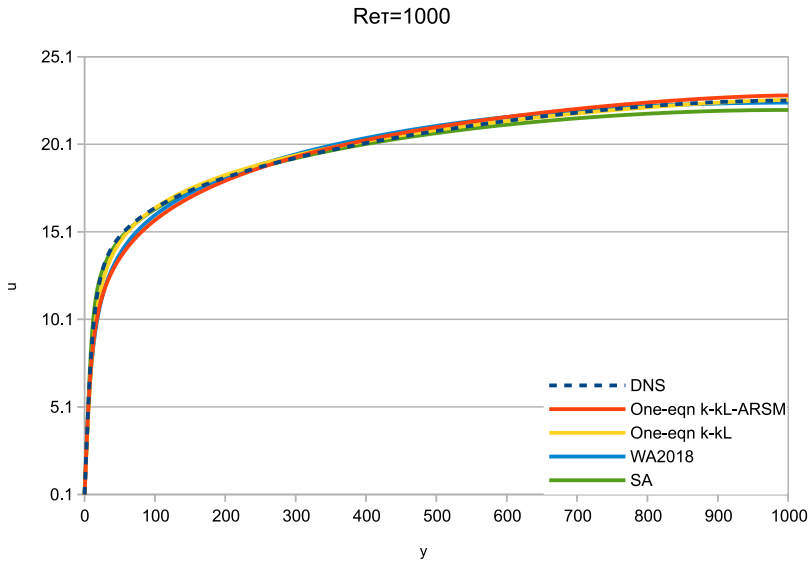

Figure 4.6 Velocity profile in the channel at Reτ = 1000.

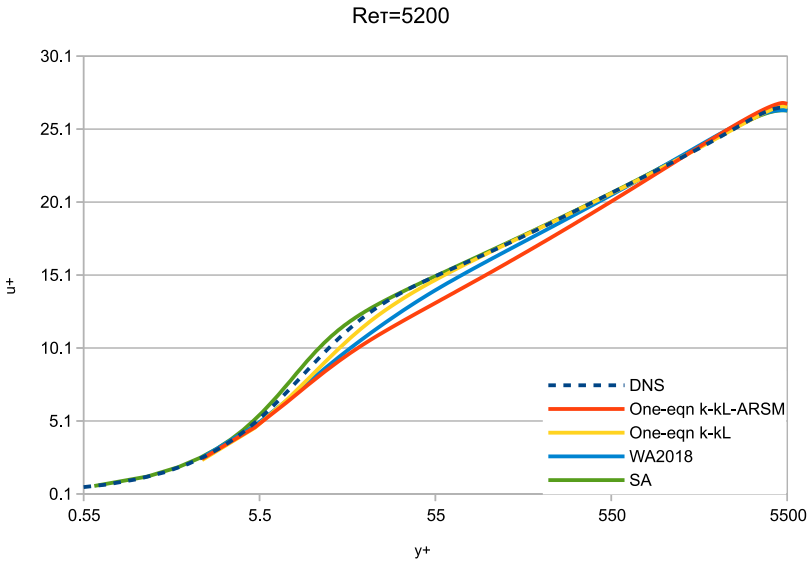

Figure 4.7 Near Wall velocity profile in the channel at Reτ = 5200.

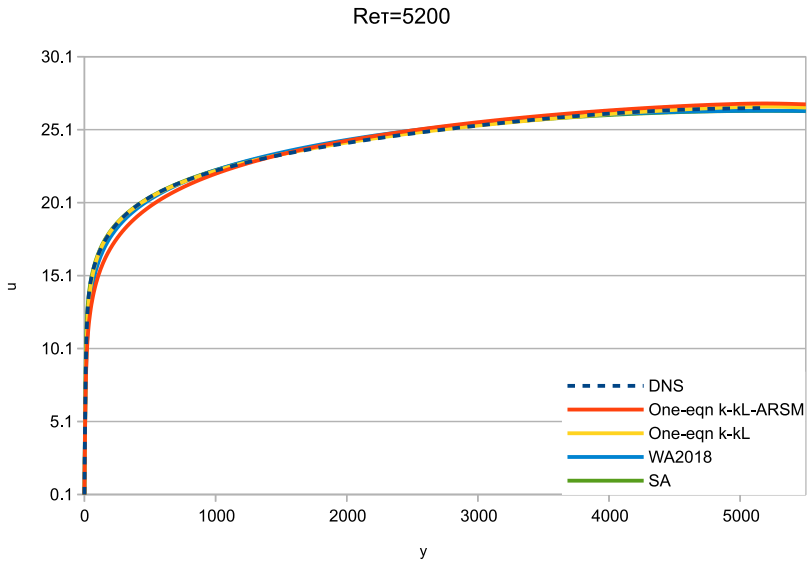

Figure 4.8 Velocity profile in the channel at Reτ = 5200.

An additional channel flow at high Reynolds number was also computed (as suggested on NASA TMR) to test the new model with inflow Mach number of 0.2 and Reynolds number of 80 million based on channel height. The velocity profile from the new k-kL-ARSM model is

compared to that obtained from SA model at x=500. Figure 4.9 shows that the new one-equation k-kL-ARSM model agrees well with SA model for the velocity profile.

Figure 4.10 shows the comparison of the turbulent viscosity ratio (μ_t/μ). The WA model has an obvious the best profile among the four models; the other models have "kinks" near the centerline region. It can be seen from this figure that the two k-kL models have very similar "kinks", which indicate that the models cannot properly handle the sudden drop in the shear strain rate S near centerline. Hence, the conversion of the term $\frac{\sigma_\phi \nu_t^2}{4S^2}\frac{\partial S}{\partial x_i}\frac{\partial S}{\partial x_i}$ in Eq. (4.12) may not be enough, and further correction may be needed to fix the "kink."

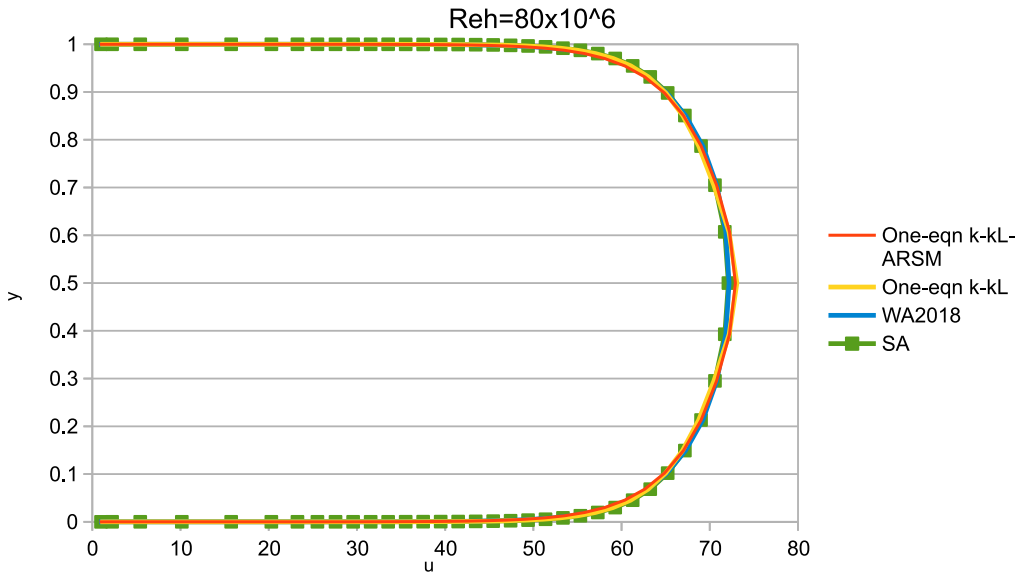

Figure 4.9 Comparison of velocity profiles in fully developed turbulent channel flow, $Re_h = 80 \times 10^6$.

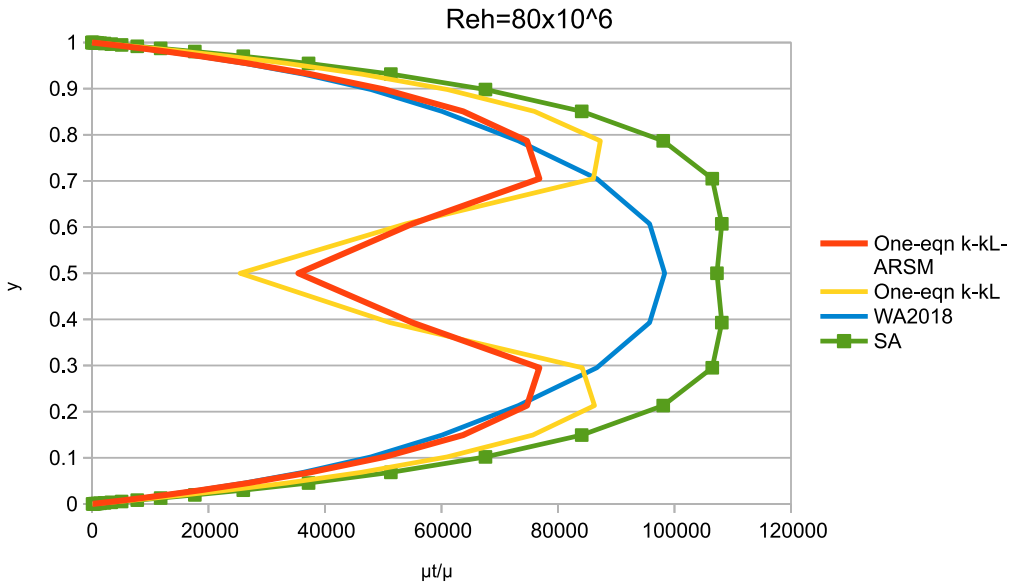

Figure. 4.10 Comparison of turbulent viscosity ratio in fully developed turbulent channel flow, $Re_h = 80 \times 10^6$.

4.3.3 Flow past 2D NASA Wall-Mounted Humps

Due to the insensitivity to Reynolds number and response to active flow control, flow over a wall-mounted hump is a classical flow separation case which has been extensively computed for validating various turbulence models. The geometry and boundary conditions of experimental flow are shown in Figure. 4.11, which are given by Seifert and Pack [23] and Greenblatt et al. [24]. Due to freestream Ma=0.1, and Re_c=936,000 based on hump chord length, a fully turbulent flow condition is used with a uniform inlet velocity of U_{inlet}=34.6m/s. The calculation of skin friction coefficient and pressure coefficient are of interest in this case.

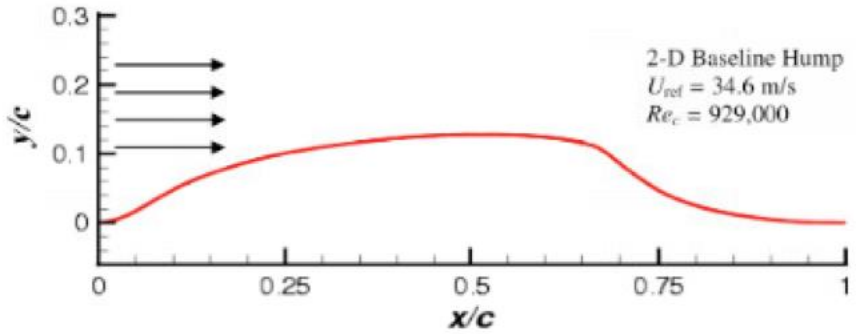

Figure 4.11 Geometry and flow parameters for the 2D hump [23, 24].

The comparison of skin friction coefficient C_f is shown in Figure. 4.12. Before the flow separation, the one-equation k-kL model obviously fails to predict the magnitude of C_f along the hump wall. On the other hand, the new k-kL-ARSM model has the best result in the region near the separation point, and the WA2018 underpredicts the magnitude of C_f in this region. The kinks in the result from the four models may be caused by mesh near the wall, and it should be eliminated by a set of finer grids. The flow separation occurs at location x/c=0.6~0.7; all four models have overall good agreement with the experimental data, however the SA model underpredicts the value of C_f at the location $x/c \approx 0.6$. The one-equation k-kL-ARSM gives a good prediction of the reattachment point near x/c=1.0 and the C_f between separation point and reattachment point is identical to the experimental data and SA model. On the other hand, WA2018 and one-equation k-kL fail to predict the C_f in this region accurately. For the region x/c >1, it is obvious that the newly proposed k-kL-ARSM model has C_f, which is much closer to the experimental data.

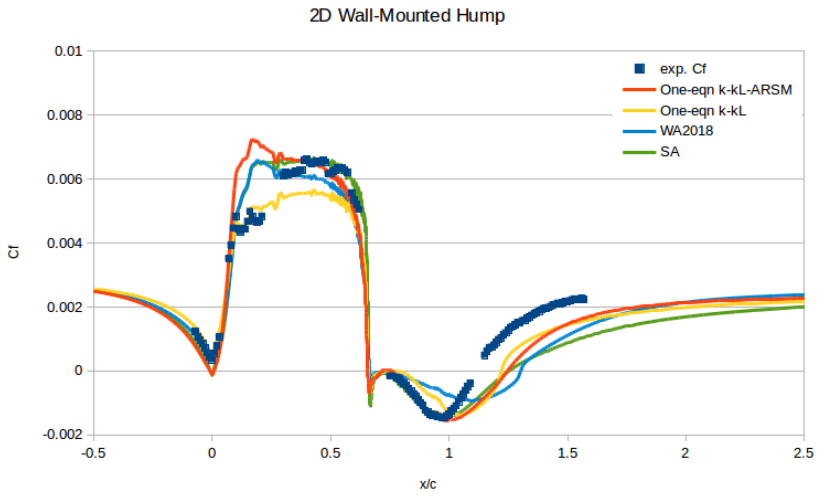

Figure 4.12 Comparison of skin-friction coefficient for flow over a 2D hump.

Figure 4.13 shows the comparison of pressure coefficient C_p. The new k-kL-ARSM model and SA model have identical results which overlap each other. The one-equation k-kL model is most accurate while WA2018 is relatively not that accurate.

In this case, the new one-equation k-kL-ARSM has successfully predicted the flow separation and reattachment. The overall performance of the new model in prediction of C_f and C_p is the best among the four models.

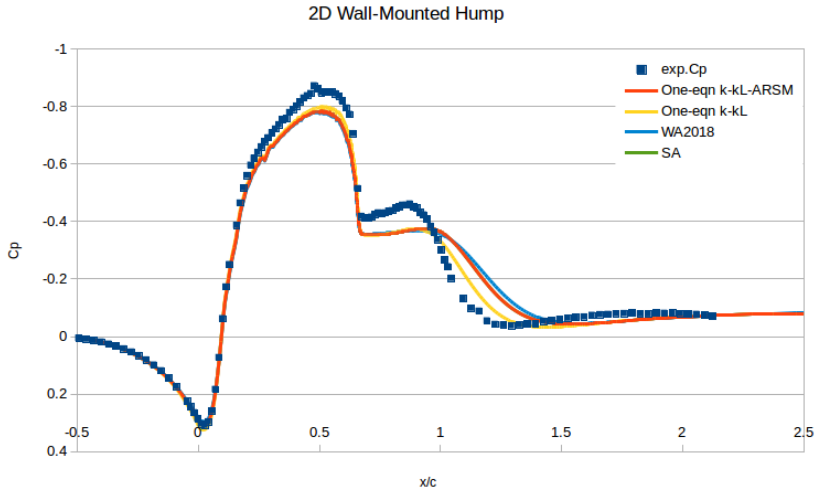

Figure 4.13 Comparison of pressure coefficient for flow over a 2D hump.

4.3.4 Flow over a 2D Backward-Facing Step

The 2D backward-facing step is a typical benchmark case for computation of separated flow. In this case, flow separation is induced by a turbulent boundary layer encountering a sudden back step. The experimental data has been obtained by Driver and Seegmillar [25]. The step has a height of H=0.0127m. Hence, the Reynolds number Re based on the step height is 3.6×10^4, and the Mach number Ma is 0.128 at the reference point (x/H=-4). The fully turbulent boundary conditions are used in this case; the details of the boundary conditions and geometry are given in Figure 4.14.

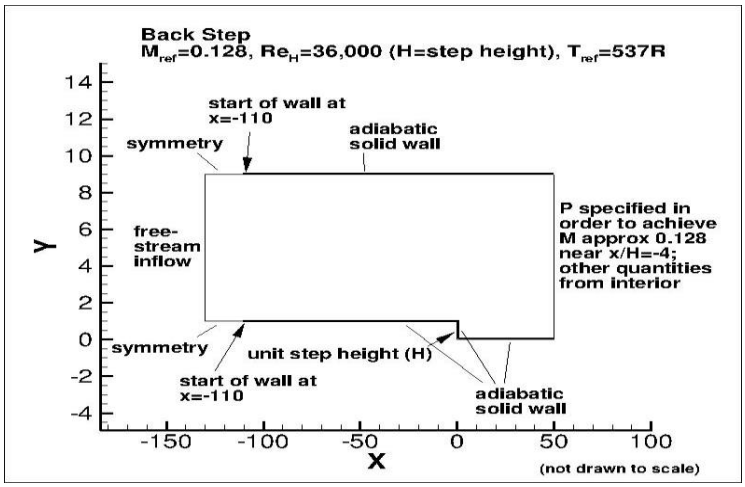

Figure 4.14 Backward facing step geometry and flow conditions [1].

The comparison of skin friction coefficient at the bottom wall is shown in Figure 4.15. One of the key measures of success is the capability to predict the reattachment point. The experimental data shows that the reattachment point is at about x/H=6.3, which is accurately predicted by all the models. Even though SA model is more accurate at upstream of the step, the one-equation k-kL-ARSM performs better all the way down to the outlet of the channel. Moreover, the one-equation k-kL model is slightly better than the k-kL-ARSM model after the reattachment point. The WA2018 has the best agreement with the experimental data.

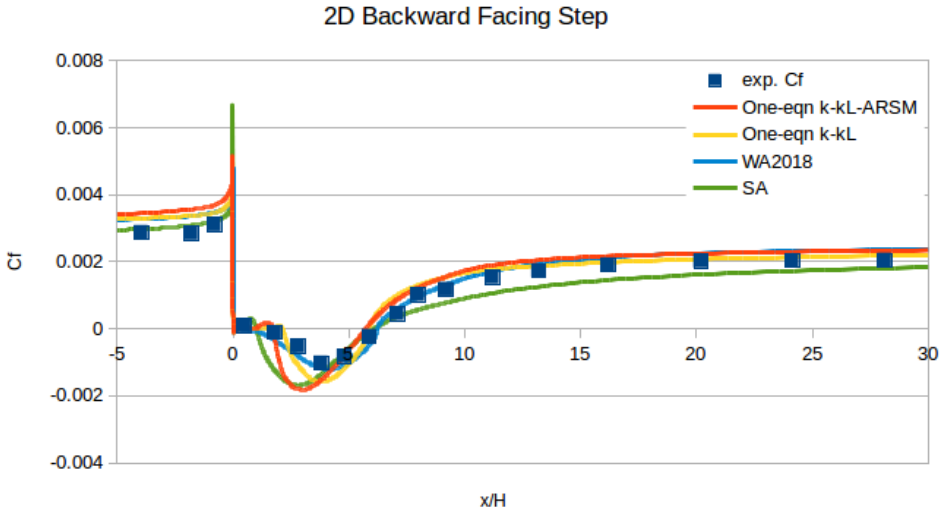

Figure 4.15 Skin friction coefficients along the lower wall of the backward facing step.

Figure 4.16 shows the comparison of pressure coefficients at the step wall. All the models fail to predict accurately the magnitude of C_p at the beginning of the separation region, however the one-equation k-kL model has the almost identical results as the experimental data near the reattachment point in the separation region. WA2018 model predicts a delayed reattachment point compared to the experimental data. Except for WA2018, the pressure coefficients shown in Figure 4.16 are consistent with the results in Figure 4.15 for the prediction of the reattachment point. For C_p in Figure 4.16, even though the one-equation k-kL is the most accurate model, the new k-kL-ARSM is reasonably good with acceptable error.

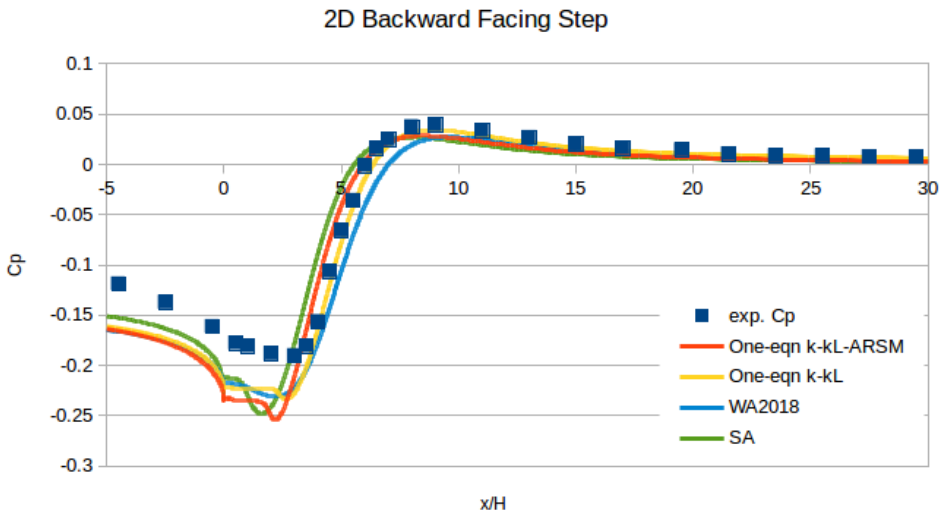

Figure 4.16 Pressure coefficients along the step wall of the backward facing step.

4.3.5 Flow over a 2D Curved Backward-Facing Step

Another flow separation case is the flow over a curved backward-facing step, which is more complex than the previous backward-facing step case. The Reynolds number based on step height and inlet velocity is 13,700. The validation data is from LES by Bentaleb et al [26].

As shown in Figure 4.17, for C_f, the flow separates at x/H=0.83 and reattaches at x/H = 4.36. In the region where the flow suddenly encounters the curved step to the separation point (x/H=0-

0.83), the one-equation k-kL-ARSM and one-equation k-kL models give the best agreement with the LES data. The new k-kL-ARSM model has identical result as the LES data after the separation point (x/H≈0.83-2.8) but has slightly delayed prediction of the separation location. All four models fail to accurately predict the reattachment point. Right after the flow reattachment, WA2018 and SA models have similar and better results than the k-kL, k-kL-ARSM models. However, both the k-kL and k-kL-ARSM models have very good agreement with the LES data downstream (x/H>6).

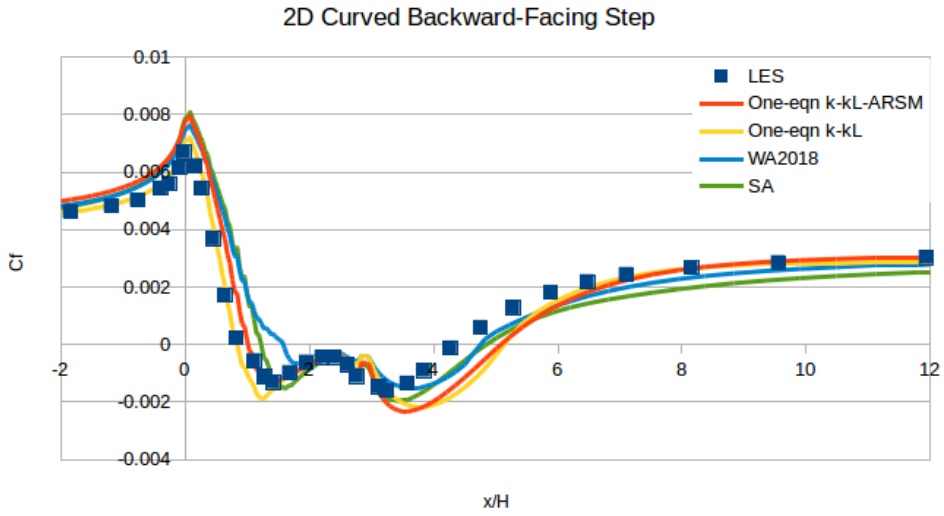

Figure 4.17 Skin friction coefficients along the lower wall of the curved backward facing step.

The comparison of pressure coefficient is shown in Figure 4.18. The one-equation k-kL-ARSM model shows the best result among the four models. The significant difference occurs in the separation and reattachment regions. This case shows that the algebraic Reynolds-Stress model (ARSM) can improve the performance when computing a separated flow.

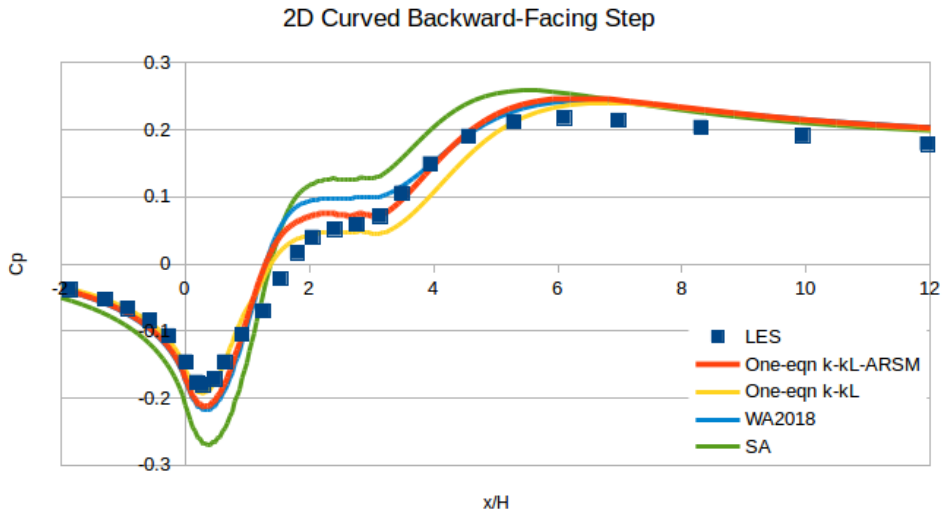

Figure 4.18 Pressure coefficients along the step wall of the curved backward facing step.

4.3.6 Flow in a 2D Asymmetric Plane Diffuser

Figure 4.19 shows the computational geometry of the asymmetric plane diffuser. It is the Buice and Eaton asymmetric diffuser study #1 (baseline) from NPARC Alliance CFD Verification and Validation Archive [27]. Based on Buice and Eaton's work [28], the Reynolds number Re based on the width of the inflow is 20,000, and inlet Mach number Ma is 0.06. The flow separation occurs at the inclined wall due to the adverse pressure gradient. It is a challenging case to predict accurately by the turbulence models. The quantities of interests in this case are skin friction coefficient and pressure coefficient along the lower wall and the upper wall of the diffuser.

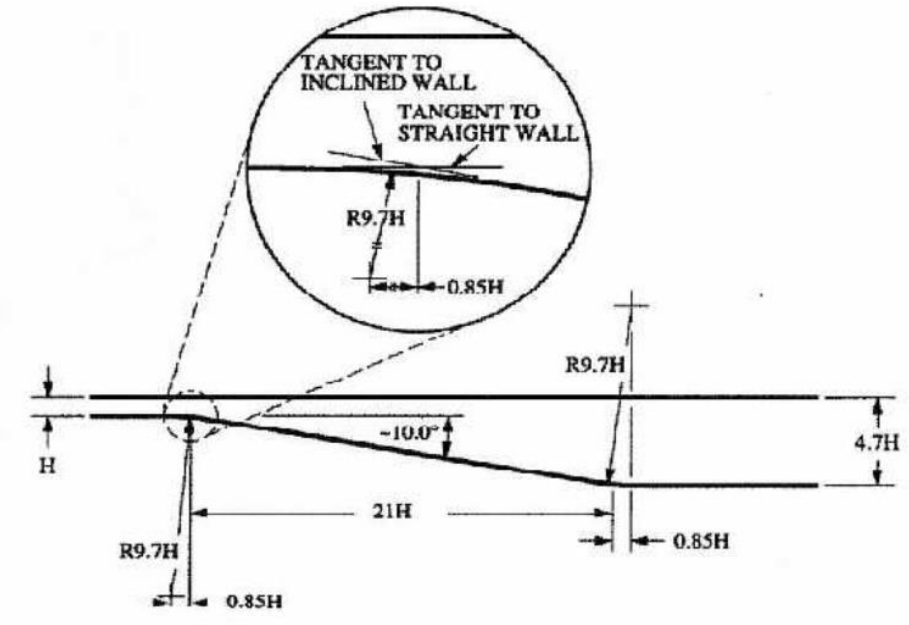

Figure 4.19 Geometry of the 2D asymmetric plane diffuser [27].

Figures 4.20 and 4.21 show the comparison of skin friction coefficient on the bottom wall and top wall, respectively. It can be seen from Figure 4.20 that the experimental data indicates the flow separates at x/h≈7.6 and reattaches at x/h≈28.0 along the bottom wall. In this case, the one-equation k-kL model fails to predict either the separation point or the reattachment point, as well as the behavior in the separation bubble as shown in Figure 4.24. The SA and new k-kL-ARSM model give an early prediction of the separation point; however WA2018 model shows the ability to accurately predict the separation. Considering the downstream region, the one-equation k-kL-ARSM is the only model that has good agreement with the experimental data near the reattachment region and after-reattachment region. Figure 4.25 shows that the new k-kL-ARSM model successfully captures the separation bubble on the inclined wall. The WA2018 model predicts the reattachment point at x/h≈24.6, which is much earlier than the experimental data. Meanwhile, even though the SA model predicts the reattachment point correctly, it underpredicts the magnitude of C_f after the flow reattaches. Figure 4.21 shows C_f along the upper

wall; it is obvious that the WA2018 model has the best agreement with the experimental data. The one-equation k-kL-ARSM slightly underpredicts the magnitude of C_f in the region right ahead of the separation bubble while SA model significantly underpredicts C_f in that region. As stated previously, since the one-equation k-kL model cannot predict the flow separation in this case, it gives a monotonically decreasing result for C_f.

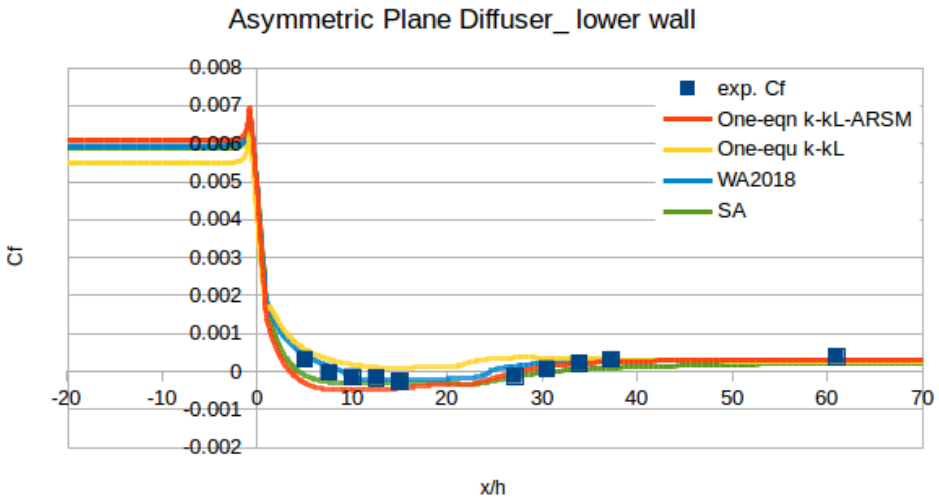

Figure 4.20 Skin friction coefficients along the lower wall of the asymmetric plane diffuser.

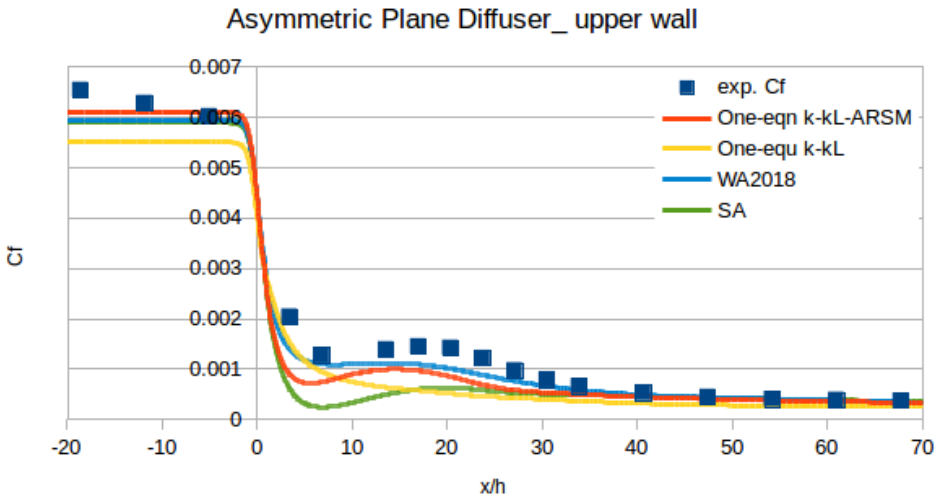

Figure 4.21 Skin friction coefficients along the upper wall of the asymmetric plane diffuser.

Figures 4.22 and 4.23 show the comparisons of pressure coefficient along the bottom and top walls, respectively. All four models fail to predict the C_p correctly; they overpredict the magnitude of C_p. Relatively, the one-equation k-kL-ARSM has result closest to the experimental data.

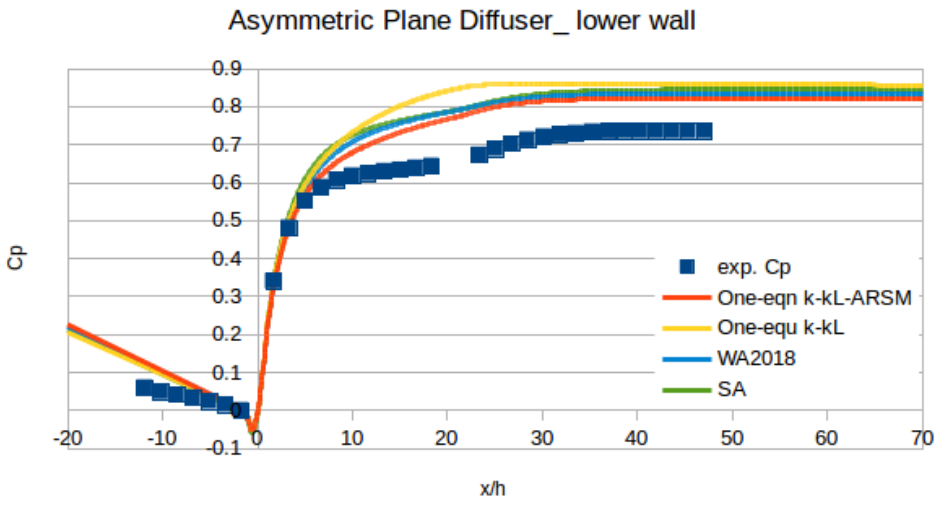

Figure 4.22 Pressure coefficients along the lower wall of the asymmetric plane diffuser.

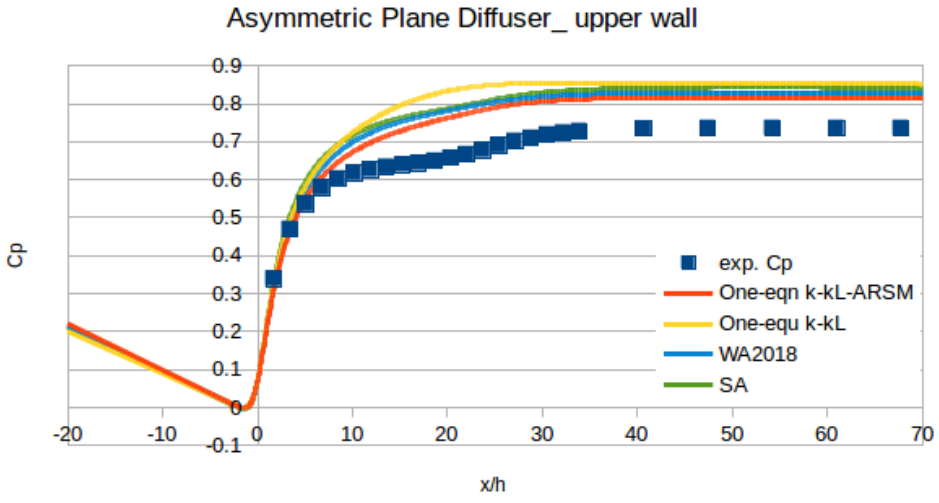

Figure 4.23 Pressure coefficients along the upper wall of the asymmetric plane diffuser.

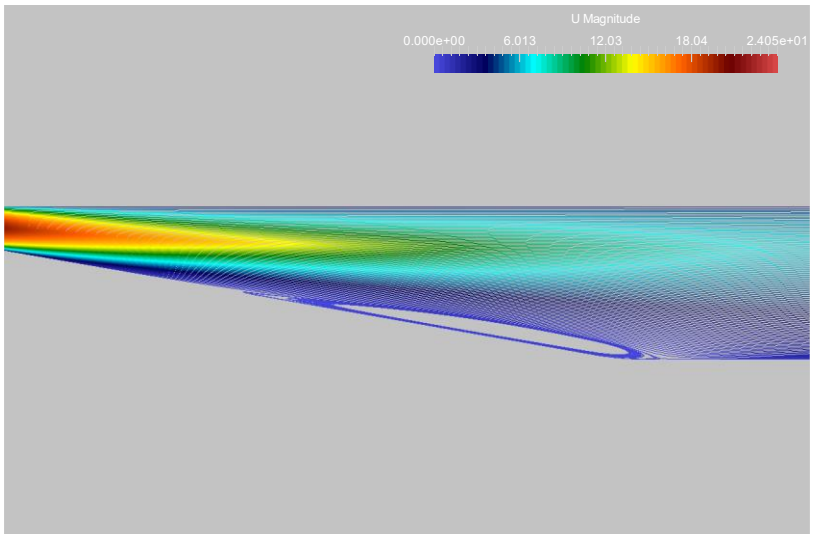

Figure 4.24 Prediction of the separation bubble in the diffuser by one-equation k-kL model.

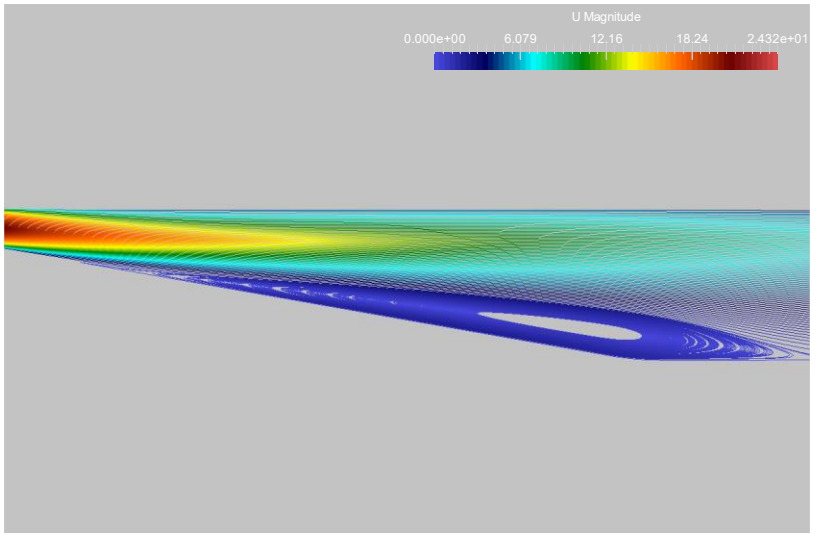

Figure 4.25 Prediction of the separation bubble in the diffuser by the k-kL-ARSM model.

4.3.7 Flow in a 3D Supersonic Square Duct

The computation of supersonic flow in a 3D square duct is another test case often used for assessing the accuracy of turbulence models for internal flows with corners. Due to the turbulence anisotropies, the models that use the Boussinesq assumption generally do not work in this case. Figure 4.26 shows the geometry and boundary conditions as well as the two cuts

inside the geometry along which the data is extracted. The experiment was conducted by Davis and Gessner's [29]. The square duct has a width and height of D=25.4 mm, and a length of x/D=50. The Reynolds number Re_D based on channel height and width is 508,000, and the Mach number is 3.9 at a reference temperature of T=520 R. In this case, the dimensionless velocity profile is compared at two different cross-sections (diagonal and vertical).

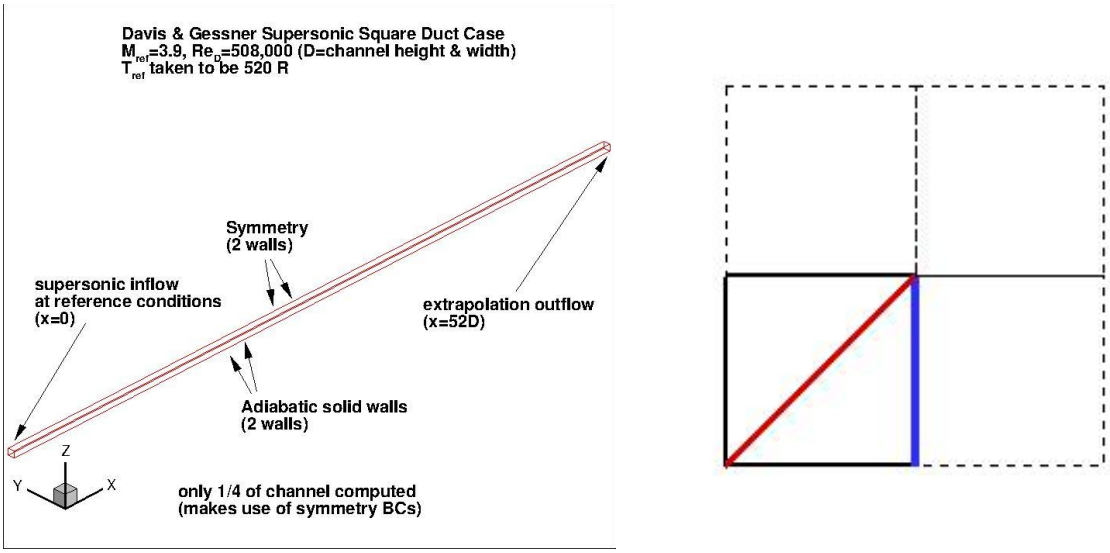

Figure 4.26 Geometry and boundary conditions of the square duct (left) and diagonal/vertical cut (right) [1]

Figure 4.27 shows the comparison of the dimensionless velocity profile at cross-section x/D=40 in two different cuts, diagonal and vertical as shown in Figure 4.26, respectively. It is obvious that all four models fail to get accurate results in the region beyond the center; the flow in the corner regions is very complex due to the presence of cross-flow vortices and none of the models capture the vortices in the corner region. Based on Abdol-Hamid's recent work [16], even the two-equation k-kL-ARSM cannot give accurate prediction in this region. Hence, the k-kL-ARSM does not work in this case. It has been shown by several researchers that the turbulence models with Quadratic Constitutive Relation (QCR) can significantly improve the prediction in

this case. The most recent work by Abdol-Hamid [16] indicates that k-kL-QCR has a very good agreement with the experimental data in this case.

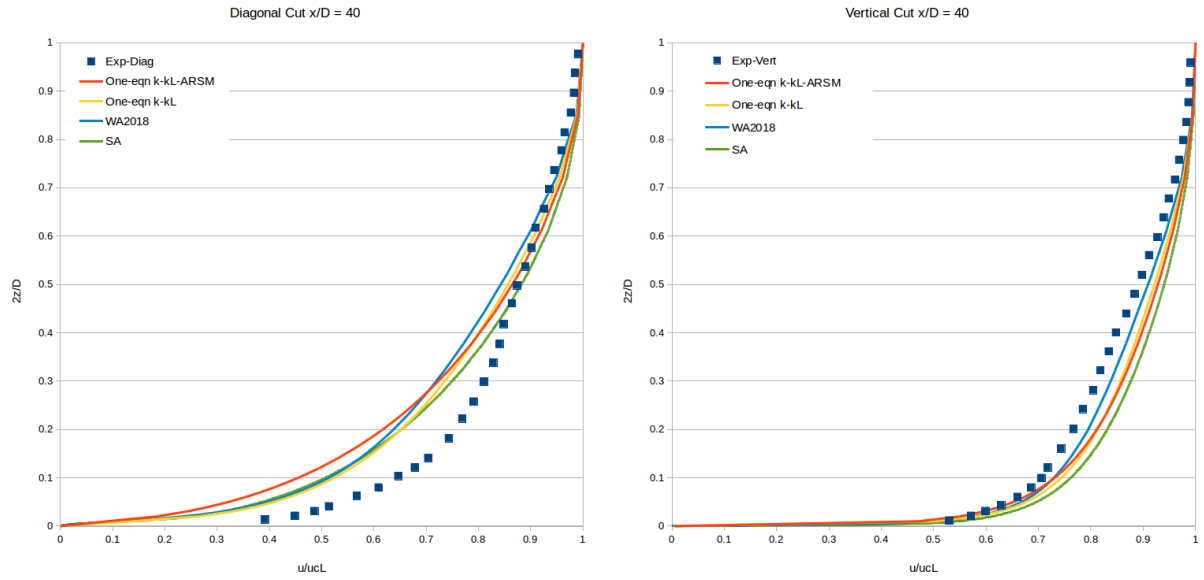

Figure 4.27 Comparison of dimensionless velocity profile along diagonal cut (left) and vertical cut (right) in Figure 4.26 at x/D =40.

Chapter 5: Development of a New Algebraic Transitional Flow Model

5.1 Introduction

Accurate prediction of laminar to turbulent transition has remained a challenging problem in computational fluid dynamics for decades. Transitional flows frequently occur in many industrial applications such as flow past airplanes, automobiles, wind turbines and turbomachines to name a few. The relatively moderate Reynolds number can result in flow past a body in all three flow regimes --- laminar, transitional and turbulent which makes its prediction very difficult and challenging. Due to large uncertainty in the prediction of transition location, the current transition prediction approaches based on empirical methods need to be improved. Accurate prediction of transition will greatly help the design of airfoils that are widely used in airplanes, turbomachinery and wind turbines.

The most widely used model for computing transitional flow in industry is the Langtry-Menter four-equation transitional model, which is also known as γ-Re_θ-SST model developed by Menter et al. [30]. The model couples Menter's SST k-ω two-equation turbulence model with an additional intermittency equation "γ" and a "Re_θ" transport equation. To remove the lack of Galilean invariance and to reduce the computational cost, a three-equation γ-SST model was developed by Menter et al., which is independent of the Re_θ equation [31].

This work is based on the recent work by Cakmakcioglu et al. [32]. The original work was development of an algebraic transition model for the Spalart-Allmaras (SA) model. Instead of using a transport equation for solving the intermittency γ, the γ is solved by an algebraic equation

which reduces the computational cost and still shows very good accuracy in several benchmark test cases.

The starting point of this research is to try to develop a new one-equation transition model, WA-T, based on algebraic intermittency gamma equation of Cakmakcioglu et al. [32] and Wray-Agarwal (WA2018) [6] model. Since it has only one transport equation, it will have theoretically lower computational cost than the existing two-equation WA-γ transition model [33]. The newly developed model is validated by computing the benchmark transitional flow cases of flow over the zero-pressure gradient T3 series of flat plates. Additionally, the model is also validated by computing the transitional flow past S809 airfoil. The computational results are compared to available experimental data and results from other transition models.

5.2 Integration of Algebraic Transition Model with WA2018 Model

The one equation Wray-Agarwal (WA2018) model is a newly proposed turbulence model developed by Han et al. [6]. The one-equation WA model inherits the advantages of both the two-equation k-ω model and k-ε model with lower computational cost, and it sometimes performs even better than both the k-ω model and the k-ε model in some of the benchmark validation cases from NASA Turbulence Modeling Resource (TMR). Without any modification, this model alone cannot predict the transition from laminar flow to turbulent flow. Based on Cakmakcioglu et al.'s work [32], coupling the turbulence model with the turbulence intermittency γ obtained from the algebraic equation can provide the capability to predict the transition. The integration of one-equation Wray-Agarwal turbulence model with γ term can provide an accurate and efficient one-equation transition prediction model with capability similar

to the four equation Shear-Stress Transport (SST) transition model proposed by Menter et al [30] but at least two to three times more efficient.

The baseline one-equation WA2018 model by Han et al. [6] is modified to include γ term as shown in Eq. (5.1). The WA model is coupled with intermittency term γ by multiplying γ with the kinetic energy production term C_1RS as shown below:

$$\frac{\partial R}{\partial t}+\frac{\partial u_j R}{\partial x_j}=\frac{\partial}{\partial x_j}\left[(\sigma_R R+\nu)\frac{\partial R}{\partial x_j}\right]+C_1\gamma RS+f_1 C_{2kw}\frac{R}{S}\frac{\partial R}{\partial x_j}\frac{\partial S}{\partial x_j}$$
$$-(1-f_1)min\left[C_{2k\omega}R^2\left(\frac{\frac{\partial S}{\partial x_j}\frac{\partial S}{\partial x_j}}{S^2}\right),C_m\frac{\partial R}{\partial x_j}\frac{\partial R}{\partial x_j}\right] \quad (5.1)$$

The value of γ will be 0 in laminar flow, and it turns into 1 in fully turbulent flow. In Wray-Agarwal (WA2018) model, the eddy viscosity is given by:

$$\nu_t=f_\mu R \quad (5.2)$$

In Eq. (5.1), the intermittency term γ is formulated as:

$$\gamma=1-\exp\left(-\sqrt{Term_1}-\sqrt{Term_2}\right) \quad (5.3)$$

Here $Term_1$ is designated to trigger the transition location, and $Term_2$ helps the intermittency to penetrate into the boundary layer [32]. $Term_1$ is given by:

$$Term_1=\frac{\max(Re_\theta-Re_{\theta c},0.0)}{\chi_1 Re_{\theta c}} \quad (5.4)$$

where

$$Re_\theta=\frac{Re_v}{2.193}\text{ and }Re_v=\frac{\rho d^2}{\mu}\Omega \quad (5.5)$$

and d is the wall distance. In Eq. (5.6), the local turbulence intensity is different from the work by Menter et al [31] which is calculated by using k and ω. Instead, it is set to a constant value in Eq. (5.6)

$$Re_{\theta c} = 803.73(Tu_{\infty} + 0.6067)^{-1.027} \tag{5.6}$$

$Term_2$ is given by:

$$Term_2 = \frac{\max(\nu_W - \chi_2, 0.0)}{\chi_2} \tag{5.7}$$

In the original work [32], the term ν_W is formulated as $\nu_{BC} = \frac{\nu_t}{Ud}$. The use of local velocity magnitude is not Galilean invariant. Menter et al. provided an approximation to the freestream velocity [31], and here the local velocity magnitude can be replaced by an approximation as well, which is given by:

$$U \sim Sd \tag{5.8}$$

Hence, the term ν_W is formulated as:

$$\nu_W = \frac{R}{Sd^2} \tag{5.9}$$

χ_1 and χ_2 are calibrated constants. Instead of using $\chi_2 = \frac{5.0}{\mathrm{Re}}$ (the use of Re may cause problem), the new χ_1 and χ_2 are given by:

$$\chi_1 = 0.002 \text{ and } \chi_2 = \frac{0.6U_{\infty}}{a_{\infty}} \tag{5.10}$$

where U_{∞} is the freestream velocity and a reference speed of sound $a_{\infty} = 343\ m/s$ is used to nondimensionalize χ_2.

In WA model, the damping function f_{μ} is designed to account for wall blocking effect, which is given by:

$$f_{\mu} = \frac{\chi^3}{\chi^3 + C_w^3}, \qquad \chi = \frac{R}{\nu} \tag{5.11}$$

where ν is the kinematic viscosity and $R = k/\omega$.

S and *W* are the mean strain rate and mean vorticity; they are given by:

$$S = \sqrt{2S_{ij}S_{ij}}, \qquad S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{5.12}$$

$$W = \sqrt{2S_{ij}S_{ij}}, \qquad W_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) \tag{5.13}$$

The WA model combines the features of standard *k*-ω and *k*-ε models. The switching function f_1 triggers the behavior of a one-equation *k*-ω or a one-equation *k*-ε model. Recall Eqs. (2.25) - (2.28), the switching function f_1 is given by:

$$f_1 = \tanh(arg_1^4), \quad arg_1 = \frac{\nu + R}{2}\frac{\eta^2}{C_\mu k\omega} \tag{2.25}$$

where

$$k = \frac{\nu_T S}{\sqrt{C_\mu}} \tag{2.26}$$

$$\omega = \frac{S}{\sqrt{C_\mu}} \tag{2.27}$$

$$\eta = S\max\left(1, \left|\frac{W}{S}\right|\right) \tag{2.28}$$

The model constants are:

$$C_{1k\omega} = 0.0829, \quad C_{1k\varepsilon} = 0.1284$$

$$C_1 = f_1(C_{1k\omega} - C_{1k\varepsilon}) + C_{1k\varepsilon}$$

$$\sigma_{kw} = 0.72, \quad \sigma_{k\varepsilon} = 1.0$$

$$\sigma_R = f_1(\sigma_{k\omega} - \sigma_{k\varepsilon}) + \sigma_{k\varepsilon}$$

$$C_{2k\omega} = \frac{C_{1k\omega}}{\kappa^2} + \sigma_{kw}, \quad C_{2k\varepsilon} = \frac{C_{1k\varepsilon}}{\kappa^2} + \sigma_{k\varepsilon}$$

$$\kappa = 0.41, \quad C_\omega = 8.54$$

$$C_\mu = 0.09, \quad C_m = 8.0$$

The boundary conditions for *R* are set to be:

$$R_\infty = 0.2\nu_\infty$$

and

$$R_{wall} = 0$$

5.3 Validation Cases

5.3.1 Zero Pressure Gradient Boundary-Layer Flow Past a Flat Plate

The first set of validation cases is flow past zero pressure gradient T3 series flat plates (T3A, T3B and T3A-), which employ different inlet velocities and turbulence intensities as shown in Table 5.1. A typical grid for the computational domain is shown in Figure 5.1. The computational results from one-equation WA-T model are compared with the results from the four-equation SST-Transition model and the experimental data [34].

Table 5.1 Inlet flow conditions for T3 series of flat plates.

	U_∞ (m/s)	Tu_∞(%)	μ_T/μ	ρ (kg/m^3)	μ (kg/m.s)
T3A	5.4	3.5	13.3	1.2	1.8e-5
T3B	9.4	6.5	100	1.2	1.8e-5
T3A-	19.8	0.874	8.72	1.2	1.8e-5

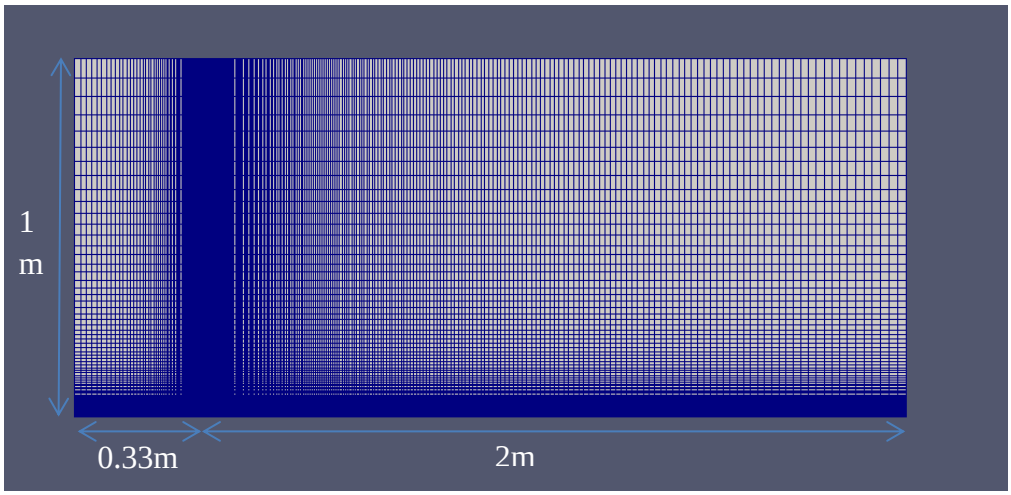

Figure 5.1 Mesh (291x191) for T3 series flat plates.

Figures 5.2-5.4 show the comparison of skin friction coefficient C_f along the flat plate. In Figure 5.2, for flow past T3A flat plate, the SST transition model identically matches the experimental data in the laminar flow region and in the transition regime. However, it cannot reach the peak value in the fully turbulent region. The WA-T model successfully predicts the peak value; however it has an obvious delayed prediction of the transition point. In Figure 5.3, the SST transition model fails to predict the transition flow over T3B flat plate. The WA-T model is much better than SST transition model in predicting this flow. In Figure 5.4, the WA-T model is unable to predict the transition point in T3C flat plate flow, showing a fully laminar prediction. This may be caused by the very low free stream turbulence intensity, which could be fixed by an additional source term.

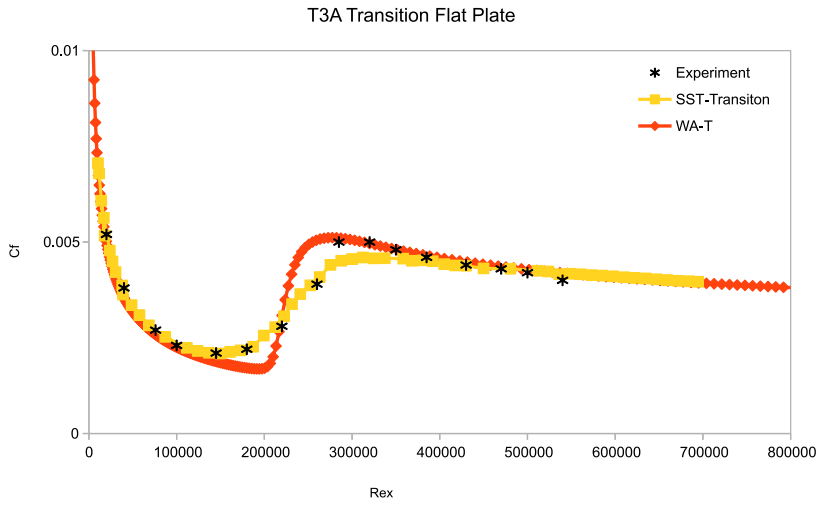

Figure 5.2 Transitional flow past T3A flat plate.

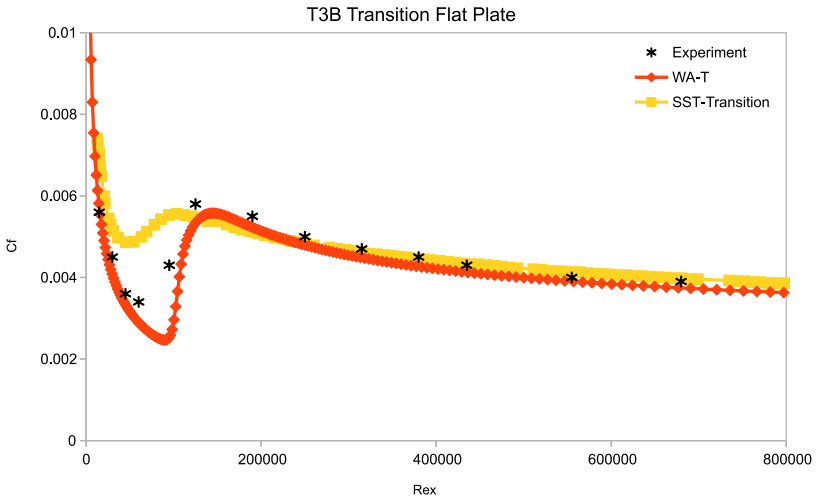

Figure 5.3 Transitional flow past T3B flat plate.

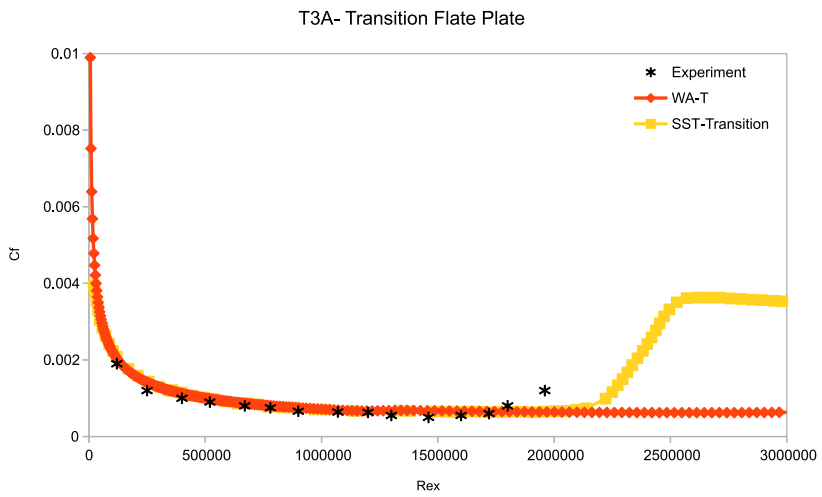

Figure 5.4 Transitional flow past T3A- flat plate.

5.3.2 Flow Past a 2D S809 Airfoil

The S809 airfoil was designed for wind turbines; it is thick airfoil with 21% thick chord [35]. It is a laminar-flow airfoil for horizontal-axis wind-turbine applications. Accurate computational results may greatly improve the efficiency of the airfoil in energy-generating applications. In this case, the Reynolds number based on the chord length is 2 million and three angles of attack are considered --- 0°, 5°, and 10°. For every AOA, the inlet turbulence intensity is Tu_∞ = 0.2%, and the viscosity ratio $\mu_t/\mu = 10$.

Figures 5.5-5.7 show the comparison of pressure coefficients on the airfoil. The WA-T transition model is compared with experimental data and SST transition model. For each of the selected angle of attack, the WA-T transition model has better result than the four-equation SST transition model. Especially when AOA = 0 and 5 degrees, the WA-T model accurately predicts C_p at the trailing edge while the SST transition model fails to have good results. The overall performance of WA-T model in this case is much better than SST transition model.

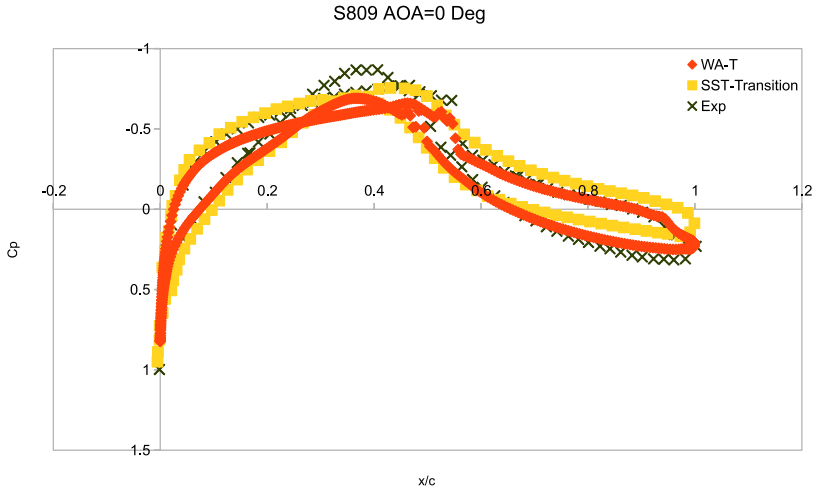

Figure 5.5 Pressure coefficient distribution on S809 airfoil at AOA = 0°.

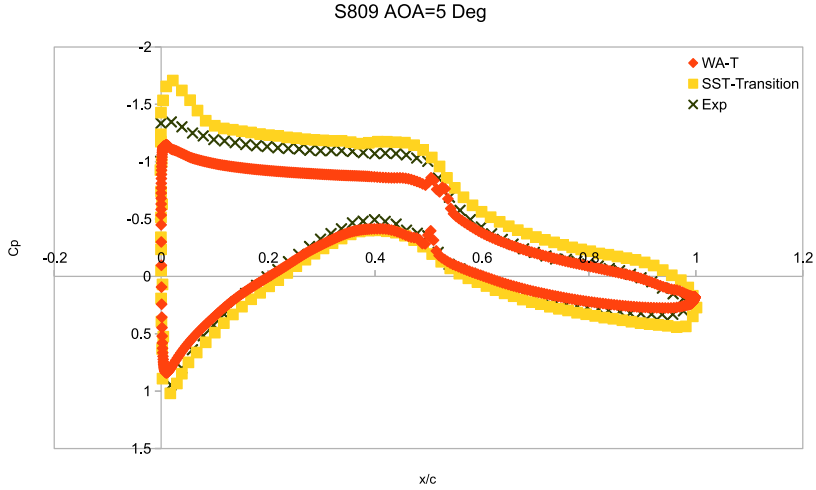

Figure 5.6 Pressure coefficient distribution on S809 airfoil at AOA = 5°.

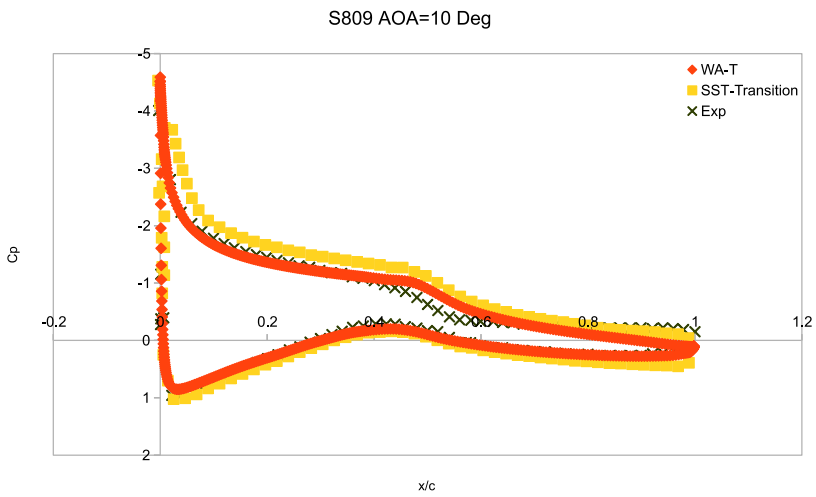

Figure 5.7 Pressure coefficient distribution on S809 airfoil at AOA =10°.

Chapter 6: Summary

6.1 The WA-Rough Model

The WA2017 and WA2018 turbulence models are extended to compute flows with surface roughness and are successfully applied for computing the flow past a rough flat plate with varying roughness heights, flow past a S809 smooth and rough airfoil, and fully developed flow in a rough channel. The roughness extension of wall distance free version of Wray-Agarwal turbulence model (WA2018) gives better results compared to the original WA2017 model. Overall, the new WA2018-Rough performs better than the SA-Rough model. The WA2018-Rough model can accurately predict the skin friction coefficient for any k_s in the range $0.00025m \leq ks \leq 0.0015m$. For the S809 airfoil, SA-Rough model gives the best results compared to both WA2017-Rough and WA2018-Rough models. Since stall occurs when AOA is greater than 10°, WA2018-Rough model is unable to accurately predict the flow for AOA greater than 8°. A laminar to turbulent transitional flow model modification to WA2018 model is needed to predict the flow separation on rough S809 airfoil. However, for small AOA below 8°, both WA2018-Rough and SA-Rough models give good results. For fully developed turbulent flow in a rough-wall channel, the WA2018-Rough model performs better than SA-Rough model, giving reasonably good prediction of velocity profile compared to the experimental data. It is demonstrated that WA2018-Rough model can be used to compute attached flows over objects with surface roughness quite accurately.

6.2 The One-Equation k-kL-ARSM Model

The newly developed k-kL-ARSM model shows improved accuracy compared to the SA, WA2018 and one-equation k-kL models for the benchmark test cases considered. With successful validation of the flat plate, channel, NASA hump, backward facing step, curved backward-facing step and asymmetric plane diffuser benchmark test cases, it has demonstrated good capability to predict wall-bounded flows and flow with small regions of separation. It should be mentioned that the model is still in early stage of development and needs further improvements and validation. However, it provides a new model with good potential for improved accuracy and efficiency and should be further exercised and improved by researchers in turbulence modeling community. The accurate simulation of supersonic square duct requires further modification of all the turbulence models by using Quadratic Constitutive Relation (QCR) instead of the linear Boussinessq approximation. Additionally, including an elliptic relaxation (also known as elliptic blending) may improve the near wall velocity profile for the channel flow. Despite the lack of accuracy in the near wall region for channel flow case, the one-equation k-kL-ARSM model is competitive in efficiency and is more accurate than the other three models (WA2018, k-kL, SA) in computing the flat plate, hump, backward facing step, curved backward-facing step and asymmetric plane diffuser cases. It has great potential to be an improvement to the existing one-equation models.

6.3 The One-Equation WA-T Transition Model

This work employs the recent research idea by Cakmakcioglu et al [32] which couples an algebraic transition model with the Spalart-Allmaras (SA) one-equation turbulence model. A new algebraic transition model is developed which is coupled with the one-equation Wall-

Distance Free Wray-Agarwal model. The newly proposed one-equation WA-T model is successfully validated by computing the transitional flow past zero-pressure gradient T3A and T3B flat plates, and flow past a S809 airfoil. It shows improved accuracy compared to the four-equation SST transition model. As a one-equation model, WA-T is very efficient which has lower computational cost and reasonably good results. However, in T3A- flat plate case, it cannot predict the transition, which means that the intermittency term γ cannot properly trigger the correct point where the flow turns into fully turbulent. The model still needs further validation e.g. transition flow past non-zero-pressure gradient flat plates (T3C series), Aerospatiale-A airfoil and NRL-7301 two-element airfoil among other benchmark cases for transitional flow. Based on the current work, it can be concluded that the WA-T model has great potential and it is worth investigating it more in-depth. This model will be further developed in the future research.

References

[1] NASA Langley Turbulence Modeling Resource website, http://turbmodels.larc.nasa.gov., (retrieved November 2017).

[2] Spalart, P. R. and Allmaras, S. R, "A One Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 1992-0439, 1992.

[3] Menter, F. R., "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," AIAA Journal, Vol. 32, No. 8, August 1994, pp. 1598-1605.

[4] Wilcox, D. C., "Formulation of the k-ω Turbulence Model Revisited," AIAA Journal, Vol. 46, No. 11, 2008, pp. 2823-2838.

[5] Wray. T., "Development of a One-Equation Eddy Viscosity Turbulence Model for Application to Complex Turbulent Flows," PhD thesis, Washington University in St. Louis, December 2016.

[6] Han, X., Rahman, M. M., and Agarwal, R. K., "Development and Application of a Wall Distance Free Wray-Agarwal Turbulence Model," AIAA Paper 2018-0593, AIAA SciTech Forum, Kissimmee, FL, 8-12 January 2018.

[7] Han, X., Wray, T. J., Agarwal, R. K., "Application of a New DES Model Based on Wray-Agarwal Turbulence Model for Simulation of Wall-Bounded Flows with Separation," AIAA Paper 2017-3966, 47th AIAA Fluid Dynamics Conference, June 2017.

[8] Nikuradse, J., "Law of Flow in Rough Pipes," Technical Report 1292, VDI-Forchungsheft 361, Series B, Vol. 4, 1933; NACA TM-1292, 1950.

[9] Wray, T. J., and Agarwal, R. K., "Extension of Wray-Agarwal Turbulence Model for Flow Over Rough Surfaces," AIAA Paper 2015-2785, 45th AIAA Fluid Dynamics Conference, AIAA AVIATION Forum, 2015.

[10] Mills, A., and Hang, X., "On the Skin Friction Coefficient for a Fully Rough Flat Plate," ASME Journal of Fluids Engineering, September 1983, pp. 364-365.

[11] Ramsay, R., Hoffmann, M. J., and Gregorek, G. M., "Effects of Grit Roughness and Pitch Oscillations on the S809 Airfoil," NREL/TP-442-7817, December 1999.

[12] White F. M., "Some Examples of Viscous-flow Phenomena," Viscous Fluid Flow, 3rd ed., McGraw-Hill, New York, 2006, pp. 6.

[13] Saleh. O.A.B., “Fully Developed Turbulent Smooth and Rough Channel and Pipe Flows,” Doctoral Thesis, University of Erlangen, Germany, 2005.

[14] 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,"AIAA 2016-3941, 46th AIAA Fluid Dynamics Conference. Washington D.C., June 2016.

[15] Rotta, J., “Statistische Theorie Nichthomogener Turbulenz,” Zeitschrift fur Physik, Vol.129, No.6, 1951, pp. 547-572.

[16] Abdol-Hamid, K., “Development and Documentation of kL-Based Linear, Nonlinear, and Full Reynolds Stress Turbulence Models,” NASA Technical Memorandum. 2018-219820. 2018.

[17] Bradshaw, P., Ferriss, D. H., and Atwell, N. P., "Calculation of Boundary Layer Development Using the Turbulent Energy Equation," Journal of Fluid Mechanics, Vol. 23, 1967, pp. 31-64.

[18] Townsend, A. A., "Equilibrium Layers and Wall Turbulence," Journal of Fluid Mechanics, Vol. 11. 1962.

[19] Menter, F.R., “Eddy viscosity Transport Equations and their Relation to the k–ε Model,” NASA-TM-108854, 1994.

[20] Weighardt, K., and Tillman, W., “On the Turbulent Friction Layer for Rising Pressure,” NACA TM-1314, 1951.

[21] Lee, M., and Moser, R. D., "Direct Numerical Simulation of Turbulent Channel Flow Up to Re_tau = 5200,"Journal of Fluid Mechanics, Vol. 774, 2015, pp. 395-415.

[22] Rahman, M. M., Wallin, S., and Siikonen, T., “Exploring k and ω with R–Equation Model Using Elliptic Relaxation Function,” Flow, Turbulence and Combustion, Vol. 89, 2012, pp. 121–148.

[23] Seifert, A., and Pack, L. G., “Active Flow Separation Control on Wall- Mounted Hump at High Reynolds Numbers,” AIAA Journal, Vol. 40, No. 7, July 2002, pp. 1363–1372.

[24] Greenblatt, D., Paschal, K. B., Yao, C.-S., Harris, J., Schaeffer, N. W., and Washburn, A. E., “A Separation Control CFD Validation Test Case, Part 1: Baseline and Steady Suction,” AIAA Journal, Vol. 44, No. 12, 2006, pp. 2820–2830.

[25] Driver, D. M. and Seegmiller, H. L., "Features of Reattaching Turbulent Shear Layer in Divergent Channel Flow," AIAA Journal, Vol. 23, No. 2, Feb 1985, pp. 163-171.

[26] Bentaleb, Y., Lardeau, S., and Leschziner, M. A., "Large-Eddy Simulation of Turbulent Boundary-Layer Separation from a Rounded Step," Journal of Turbulence, Vol. 13, No. 4, 2011, pp. 1-28.

[27] NPARC Alliance Verification and Validation Archive, https://www.grc.nasa.gov/WWW/wind/valid/whatsnew.html. [retrieved February 2019]

[28] Bruice, C., and Eaton, J. K., "Experimental Investigation of Flow through an Asymmetric Plane Diffuser," Stanford Univ. Dept. of Mechanical Engineering, Thermoscience Division Rept. TSD-107, Stanford, CA, 1997.

[29] Davis, D. O. and Gessner, F. B., "Further Experiments on Supersonic Turbulent Flow Development in a Square Duct," AIAA Journal, Vol. 27, No. 8, August 1989, pp. 1023-1030.

[30] Menter, F. R., Langtry, R. B., Likki, S. R., Suzen, Y. B., Huang, P. G., and Völker, S., "A Correlation-Based Transition Model Using Local Variables: Part I — Model Formulation," Volume 4: Turbo Expo 2004, 2004

[31] Menter, F. R., Smirnov, P. E., Liu, T., and Avancha, R., "A One-Equation Local Correlation-Based Transition Model," Flow, Turbulence and Combustion, Vol. 95, 2015, pp. 583–619.

[32] Cakmakcioglu, S. C., Bas, O., and Kaynak, U., "A correlation-based algebraic transition model," Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, Vol. 232, 2018, pp. 3915–3929.

[33] Nagapetyan, H., and Agarwal, R. K., "Development of a New Transitional Flow Model Integrating the Wray-Agarwal Turbulence Model with an Intermittency Transport Equation," 2018 Fluid Dynamics Conference, Reston, Virginia: American Institute of Aeronautics and Astronautics, 2018, pp. 1–12.

[34] ERCOFTAC (European Research Community on Flow, Turbulence and Combustion) [online database], URL: http://ercoftac.mech.surrey.ac.uk/[retrieved 15 Feb 2008]

[35] Somers, D. M., "Design and Experimental Results for the S809 Airfoil," Airfoils Inc., State College, PA, Jan 1997.

Appendix A: Source Code of One-Equation k-kL-ARSM Model

A1. kklARSM.C

```
/*---------------------------------------------------------------------------*\
 =========                 |
 \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
  \\    /   O peration     |
   \\  /    A nd           | Copyright (C) 2011-2015 OpenFOAM Foundation
    \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
```

```
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

\*---------------------------------------------------------------------------*/

#include "kklARSM.H"
#include "bound.H"
#include "wallDist.H"
#include "wallFvPatch.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

namespace Foam
{
namespace RASModels
{

// * * * * * * * * * * * * * Protected Member Functions  * * * * * * * * * * * * * //
template<class BasicTurbulenceModel>
tmp<volScalarField> kklARSM<BasicTurbulenceModel>::chi() const
{
    return nuTilda_/(this->nu());
}

template<class BasicTurbulenceModel>
tmp<volScalarField> kklARSM<BasicTurbulenceModel>::ka
(
    const volScalarField& S
```

```
) const
{
    return nuTilda_*S/a1_;
}

template<class BasicTurbulenceModel>
tmp<volScalarField> kklARSM<BasicTurbulenceModel>::len
(
    const volScalarField& S, volScalarField& al_
) const
{
    return -sqr(a1_)*pow(nuTilda_,0.5)*pow(S,-0.5)/al_;
}

template<class BasicTurbulenceModel>
tmp<volScalarField> kklARSM<BasicTurbulenceModel>::lvkmin
(
    const volScalarField& len
) const
{
    return len/C11_;
}

template<class BasicTurbulenceModel>
tmp<volScalarField> kklARSM<BasicTurbulenceModel>::ep
(
    const volScalarField& ka
```

```
) const
{
    return y_*sqrt(scalar(0.3)*ka)/(scalar(20.0)*(this->nu()));
}


template<class BasicTurbulenceModel>
tmp<volScalarField> kklARSM<BasicTurbulenceModel>::fphi
(
    const volScalarField& ep
) const
{
    return (1.0+Cd1_*ep)/(1.0+pow(ep,4.0));
}

//////////////////////////////////////////////////////////////////ARSM


template<class BasicTurbulenceModel>
tmp<volScalarField> kklARSM<BasicTurbulenceModel>::tau
(
    const volScalarField& S,
     const volScalarField& al_
) const
{
     return -a1_*Cr1_/(al_*S);
}
```

```
template<class BasicTurbulenceModel>
tmp<volScalarField> kklARSM<BasicTurbulenceModel>::trW2
(
    const volScalarField& tau
) const
{
        return tmp<volScalarField>
        (
            tr((tau * skew(fvc::grad(this->U_))) & (tau * skew(fvc::grad(this->U_))))
        );
}

template<class BasicTurbulenceModel>
tmp<volScalarField> kklARSM<BasicTurbulenceModel>::trS
(
    const volScalarField& tau
) const
{
        return tmp<volScalarField>
        (
            tr((tau*symm(fvc::grad(this->U_))))
        );
}

template<class BasicTurbulenceModel>
tmp<volScalarField> kklARSM<BasicTurbulenceModel>::trW
```

```
(
    const volScalarField& tau
) const
{
        return tmp<volScalarField>
        (
            tr((tau*skew(fvc::grad(this->U_))))
        );
}

template<class BasicTurbulenceModel>
tmp<volScalarField> kklARSM<BasicTurbulenceModel>::trS2
(
    const volScalarField& tau
) const
{
        return tmp<volScalarField>
        (
            tr((tau*symm(fvc::grad(this->U_))) & (tau*symm(fvc::grad(this->U_))))
        );
}




template<class BasicTurbulenceModel>
void kklARSM<BasicTurbulenceModel>::correctNut()
```

```
{
    this->nut_ = nuTilda_;
    this->nut_.correctBoundaryConditions();

    BasicTurbulenceModel::correctNut();
}

// * * * * * * * * * * * * * * * * Constructors  * * * * * * * * * * * * * * * //

template<class BasicTurbulenceModel>
kklARSM<BasicTurbulenceModel>::kklARSM
(
    const alphaField& alpha,
    const rhoField& rho,
    const volVectorField& U,
    const surfaceScalarField& alphaRhoPhi,
    const surfaceScalarField& phi,
    const transportModel& transport,
    const word& propertiesName,
    const word& type
)
:
    eddyViscosity<RASModel<BasicTurbulenceModel> >
    (
        type,
        alpha,
        rho,
```

```
    U,
    alphaRhoPhi,
    phi,
    transport,
    propertiesName
),

kappa_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "kappa",
        this->coeffDict_,
        0.41
    )
),

eta1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "eta1",
        this->coeffDict_,
        1.4
    )
),
```

```
eta2_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "eta2",
        this->coeffDict_,
        0.97
    )
),

eta3_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "eta3",
        this->coeffDict_,
        0.137
    )
),

Cphi2_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Cphi2",
        this->coeffDict_,
        eta3_.value()
```

```
    )
),

f1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "f1",
        this->coeffDict_,
        6.0
    )
),

C11_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "C11",
        this->coeffDict_,
        10.0
    )
),

C12_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
```

```
        "C12",
        this->coeffDict_,
        1.3
    )
),

Cd1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Cd1",
        this->coeffDict_,
        4.7
    )
),

C3_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "C3",
        this->coeffDict_,
        7.0
    )
),

C4_
```

```
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "C4",
        this->coeffDict_,
        7.0
    )
),

Cmu_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Cmu",
        this->coeffDict_,
        0.09
    )
),

a1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "a1",
        this->coeffDict_,
        0.3
    )
```

```
),

pl_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "pl",
        this->coeffDict_,
        0.7
    )
),

pk_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "pk",
        this->coeffDict_,
        0.7
    )
),

ph_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "ph",
```

```
        this->coeffDict_,
        0.7
    )
),

pf_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "pf",
        this->coeffDict_,
        0.7
    )
),

//////////////////////////////////////////////////////////////////////////ARSM

 C1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "C1",
        this->coeffDict_,
        1.8
    )
),
```

```
C01_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "C01",
        this->coeffDict_,
        3.4
    )
),

C22_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "C22",
        this->coeffDict_,
        0.36
    )
),

C33_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "C33",
        this->coeffDict_,
        1.25
```

```
    )
),

C44_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "C44",
        this->coeffDict_,
        0.6
    )
),

Cep1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Cep1",
        this->coeffDict_,
        1.44
    )
),

Cep2_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
```

```
        "Cep2",
        this->coeffDict_,
        1.83
    )
),

gamma0Star_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "gamma0Star",
        this->coeffDict_,
        C1_.value()/2.0
    )
),

gamma1Star_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "gamma1Star",
        this->coeffDict_,
        (1.0/2.0)*C01_.value()+(Cep2_.value()-Cep1_.value())/(Cep1_.value()-1.0)
    )
),
```

```
a11_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "a11",
        this->coeffDict_,
        0.5*(4.0/3.0-C22_.value())
    )
),


a22_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "a22",
        this->coeffDict_,
        0.5*(2.0-C44_.value())
    )
),


a33_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "a33",
```

```
        this->coeffDict_,
        0.5*(2.0-C33_.value())
    )
),

nuTilda_
(
    IOobject
    (
        "nuTilda",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    this->mesh_
),

 al_
(
    IOobject
    (
        "al",
        this->runTime_.timeName(),
        this->mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
```

```
        ),
        this->mesh_,
        dimensionedScalar("Cmustar", dimless, scalar(0.18))
    ),
    Cr1_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "Cr1",
            this->coeffDict_,
            1.51
        )
    ),

    Cr2_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "Cr2",
            this->coeffDict_,
            0.5
        )
    ),

    y_(wallDist::New(this->mesh_).y())

{
```

```
    if (type == typeName)
    {
        this->printCoeffs(type);
    }

}

// * * * * * * * * * * * * * * * * Member Functions  * * * * * * * * * * * * * * * * //


template<class BasicTurbulenceModel>
bool kklARSM<BasicTurbulenceModel>::read()
{
    if (eddyViscosity<RASModel<BasicTurbulenceModel> >::read())
    {
        return true;
    }
    else
    {
        return false;
    }
}


template<class BasicTurbulenceModel>
tmp<volScalarField> kklARSM<BasicTurbulenceModel>::DnuTildaEff() const
{
```

```
    return tmp<volScalarField>
    (
        new volScalarField("DnuTildaEff", nuTilda_*pk_ + this->nu())
    );
}

template<class BasicTurbulenceModel>
void kklARSM<BasicTurbulenceModel>::correct()
{
    if (!this->turbulence_)
    {
        return;
    }

     //Calculate Strain rate magnitude S
    volScalarField S2(2.0*magSqr(symm(fvc::grad(this->U_))));
    volScalarField S = sqrt(S2);
    bound(S, dimensionedScalar("0", S.dimensions(), SMALL));
    bound(S2, dimensionedScalar("0", S2.dimensions(), SMALL));

     // Calculate vorticity magnitude W
    volScalarField W2(2.0*magSqr(skew(fvc::grad(this->U_))));
    volScalarField W = sqrt(W2);
     bound(W, dimensionedScalar("1e-15", W.dimensions(), SMALL));
     bound(W2, dimensionedScalar("1e-15", W2.dimensions(), SMALL));

    // Local references
```

```
const alphaField& alpha = this->alpha_;
const rhoField& rho = this->rho_;
const surfaceScalarField& alphaRhoPhi = this->alphaRhoPhi_;
const volVectorField& U = this->U_;
const volScalarField ka(this->ka(S));


const volScalarField ep(this->ep(ka));
const volScalarField fphi(this->fphi(ep));

 const volScalarField tau(this->tau(S,al_));
 const volScalarField trS(this->trS(tau));
 const volScalarField trW(this->trW(tau));
 const volScalarField trW2(this->trW2(tau));
const volScalarField trS2(this->trS2(tau));

volSymmTensorField SS(symm(fvc::grad(this->U_)));   //Strain Rate Tensor
volTensorField WW(skew(fvc::grad(this->U_)));    //vorticity Tensor


volSymmTensorField S_star = tau * SS;
 volTensorField W_star = tau * WW;

 volTensorField SW_star = S_star & W_star;
 volTensorField WS_star = W_star & S_star;
```

```
volScalarField WW2 = -WW && WW;

volScalarField SS2 = S_star && S_star;

volScalarField R2 = -(-W_star&&W_star)/(S_star&&S_star);

volScalarField et2 = S_star&&S_star;




eddyViscosity<RASModel<BasicTurbulenceModel> >::correct();

volScalarField divU(fvc::div(fvc::absolute(this->phi(), U)));

tmp<volTensorField> tgradU = fvc::grad(U);

tgradU.clear();


volScalarField p = - gamma1Star_
                   /
                   (et2*gamma0Star_);

volScalarField q = (1.0
                  /
```

```
                    sqr(2.0*et2*gamma0Star_) )

                                                *

                                                (sqr(gamma1Star_)-
2.0*et2*gamma0Star_*a11_-(2.0/3.0)*et2*sqr(a33_)-2.0*R2*et2*sqr(a22_));
    volScalarField r = gamma1Star_*a11_

                                    /

                                    sqr(2.0*et2*gamma0Star_);


    forAll(al_, cellI)
{

    if(et2[cellI] < 1.0e-6)
        {

            al_[cellI] =  -(gamma1Star_.value()*a11_.value())/(sqr(gamma1Star_.value())-
2.0*sqr(a22_.value()))*(-W_star[cellI] &&W_star[cellI]));


        }

    else
        {
           scalar a = q[cellI]-sqr(p[cellI])/3.0;
          scalar b = (1.0/27.0)*(2.0*pow(p[cellI],3.0)-(9.0*p[cellI]*q[cellI])+27.0*r[cellI]);
          scalar d = (sqr(b)/4.0) + (pow3(a)/27.0);

              if(d > 0.0)
              {
```

```
                scalar t11  = pow(max((-b/2.0)+sqrt(d),0.0), 1.0/3.0);
                scalar t12  = pow(max((-b/2.0)-sqrt(d),0.0), 1.0/3.0);
                scalar alm1 = -p[cellI]/3.0+t11+t12;
               scalar alm2 = -p[cellI]/3.0-t11/2.0-t12/2.0;
                al_[cellI]  = min(alm1,alm2);
              }

              else if (d <= 0.0)
              {
                 scalar theta = acos((-b)/(2.0*sqrt(-pow3(a)/27.0)));
               scalar t21 = -p[cellI]/3.0+2.0*sqrt(-a/3.0)*cos(theta/3.0);
               scalar t22 = -p[cellI]/3.0+2.0*sqrt(-
a/3.0)*cos(2.0*constant::mathematical::pi/3.0+(theta/3.0));
               scalar t23 = -p[cellI]/3.0+2.0*sqrt(-
a/3.0)*cos(4.0*constant::mathematical::pi/3.0+(theta/3.0));
                al_[cellI] = min(t21,min(t22,t23));
              }

        }

    al_ = min(al_,-0.0005);
}

    const volScalarField len(this->len(S,al_));
    const volScalarField lvkmin(this->lvkmin(len));

    volSymmTensorField T1 = S_star-(1.0/3.0)*(trS*I);
```

```
    volTensorField T2 = (S_star&S_star)-(1.0/3.0)*(trS2*I);
    volTensorField T4 = SW_star-WS_star;

   volScalarField a44_ = 1.0/(gamma1Star_-2.0*gamma0Star_*al_*et2);

    volScalarField beta1 = 2.0*al_;
    volScalarField beta2 = -2.0*a44_*a33_*beta1;
    volScalarField beta4 = a44_*a22_*beta1;

    volSymmTensorField T_ = symm(-ka*( beta1*T1+beta2*T2+beta4*T4+(2.0/3.0)*I));

   volScalarField G(this->GName(),T_ && fvc::grad(U));



   volScalarField Gphi(max(G,-sqr(a1_)*nuTilda_*magSqr(S)/al_));
   volScalarField Glim(min(Gphi,scalar(20.0)*nuTilda_*S2));

   volScalarField Cmu_star = -al_;

   volScalarField fp =
min(max(Cr2_*G*sqr(a1_)/max(Cmu_star*nuTilda_*magSqr(S),dimensionedScalar("small",
dimensionSet(0, 2, -3, 0, 0), 1e-20)), 0.25),1.0);

   volScalarField lvkmax = C12_*kappa_*y_*fp;

   volScalarField Ux=U.component(0);
   volScalarField Uy=U.component(1);
   volScalarField Uz=U.component(2);
```

```
volScalarField U2=sqrt(
                    sqr(fvc::laplacian(Ux))
                   +sqr(fvc::laplacian(Uy))
                   +sqr(fvc::laplacian(Uz))
                  );

volScalarField lvkl = kappa_*mag(S/U2);

//limiter on lvk
volScalarField lvk = max(lvkmin,min(lvkmax,lvkl));
volScalarField Cphi1_= eta1_-eta2_*sqr(len/lvk);

//E1e
volScalarField Eke = sqr(nuTilda_) * magSqr(fvc::grad(S)) / S2;
volScalarField Ebb = max(magSqr(fvc::grad(nuTilda_)),
                dimensionedScalar("EbbMin", dimensionSet(0, 2, -2, 0, 0), 1e-15));
volScalarField E1e = C3_ * Ebb * tanh(Eke/(C3_*Ebb));


//E1a
volScalarField Eka = sqr(nuTilda_) * magSqr(fvc::grad(al_)) / sqr(al_);
volScalarField Eba = max(magSqr(fvc::grad(nuTilda_)),
                dimensionedScalar("EbaMin", dimensionSet(0, 2, -2, 0, 0), 1e-15));
volScalarField E1a = C4_ * Ebb * tanh(Eka/(C4_*Eba));

volScalarField sqnu = sqr(nuTilda_);
```

```
    tmp<fvScalarMatrix> nuTildaEqn
    (
        fvm::ddt(alpha, rho, nuTilda_)
      + fvm::div(alphaRhoPhi, nuTilda_)
      - fvm::laplacian(alpha*rho*DnuTildaEff(), nuTilda_)
     ==
        alpha*rho*a1_*(Cphi1_*Gphi-0.5*G)/S
      + alpha*rho*(pow(a1_,-5.0/2.0)*Cphi2_-0.5*pow(a1_,-1.0))*fvm::Sp(S*al_, nuTilda_)
      + alpha*rho*this->nu()*nuTilda_*(0.75-f1_*fphi)/sqr(y_)
      + alpha*rho*pl_*0.5*fvm::Sp((fvc::grad(nuTilda_) & fvc::grad(S))/S, nuTilda_)
      + alpha*rho*pl_*0.75*magSqr(fvc::grad(nuTilda_))
      + alpha*rho*pl_*2.0*E1a
       - alpha*rho*pl_*4.0*fvm::Sp(fvc::grad(al_)/al_ & fvc::grad(nuTilda_), nuTilda_)
      - alpha*rho*pl_*fvm::Sp(fvc::grad(al_)/al_ & nuTilda_*fvc::grad(S)/S, nuTilda_)
      - alpha*rho*pl_*(1.0/al_)*fvm::Sp(nuTilda_*fvc::laplacian(al_),nuTilda_)
      - alpha*rho*pl_*0.25*E1e

     nuTildaEqn().relax();
    solve(nuTildaEqn);
    bound(nuTilda_, dimensionedScalar("0", nuTilda_.dimensions(), 0.0));
    nuTilda_.correctBoundaryConditions();

    correctNut();

}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

} // End namespace RASModels

} // End namespace Foam

// *** //

A2. kklARSM.H

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2015 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
```

for more details.

You should have received a copy of the GNU General Public License

along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Class

Foam::RASModels::kklARSM

Group

grpRASTurbulence

SourceFiles

kklARSM.C

---/

#ifndef kklARSM_H

#define kklARSM_H

#include "RASModel.H"

#include "eddyViscosity.H"

// * //

namespace Foam

```
{
namespace RASModels
{

/*---------------------------------------------------------------------------*\
                    Class kklARSM Declaration
\*---------------------------------------------------------------------------*/

template<class BasicTurbulenceModel>
class kklARSM
:
    public eddyViscosity<RASModel<BasicTurbulenceModel> >
{
    // Private Member Functions

        // Disallow default bitwise copy construct and assignment
        kklARSM(const kklARSM&);
        kklARSM& operator=(const kklARSM&);


protected:

    // Protected data

        // Model coefficients
```

```
        dimensionedScalar kappa_;
        dimensionedScalar Aplus_;
        dimensionedScalar eta1_;
        dimensionedScalar eta2_;
        dimensionedScalar eta3_;
        dimensionedScalar Cphi2_;
        dimensionedScalar f1_;
        dimensionedScalar C11_;
        dimensionedScalar C12_;
        dimensionedScalar Cd1_;
        dimensionedScalar C3_;
        dimensionedScalar C4_;
        dimensionedScalar Cmu_;
        dimensionedScalar a1_;
        dimensionedScalar pl_;
        dimensionedScalar pk_;
        dimensionedScalar ph_;
        dimensionedScalar pf_;
//////////////////////////////////////////////ARSM
        dimensionedScalar C1_;
        dimensionedScalar C01_;
        dimensionedScalar C22_;
        dimensionedScalar C33_;
        dimensionedScalar C44_;
        dimensionedScalar Cep1_;
```

```
        dimensionedScalar Cep2_;
        dimensionedScalar gamma0Star_;
        dimensionedScalar gamma1Star_;
        dimensionedScalar a11_;
        dimensionedScalar a22_;
        dimensionedScalar a33_;


    // Fields

        volScalarField nuTilda_;
        volScalarField al_;
        dimensionedScalar Cr1_;
        dimensionedScalar Cr2_;


        //- Wall distance
        //  Note: different to wall distance in parent RASModel
        //  which is for near-wall cells only
        const volScalarField& y_;


// Protected Member Functions

    tmp<volScalarField> chi() const;
```

```
    tmp<volScalarField> ka(const volScalarField& S) const;

    tmp<volScalarField> len(const volScalarField& S,volScalarField& al_) const;

    tmp<volScalarField> lvkmin(const volScalarField& len) const;

    tmp<volScalarField> ep(const volScalarField& ka) const;

    tmp<volScalarField> fphi(const volScalarField& ep) const;


tmp<volScalarField> tau(const volScalarField& S,const volScalarField& al_) const;

tmp<volScalarField> trS(const volScalarField& tau) const;

tmp<volScalarField> trW(const volScalarField& tau) const;

tmp<volScalarField> trW2(const volScalarField& tau) const;

tmp<volScalarField> trS2(const volScalarField& tau) const;


    virtual void correctNut();
```

```
public:

    typedef typename BasicTurbulenceModel::alphaField alphaField;
    typedef typename BasicTurbulenceModel::rhoField rhoField;
    typedef typename BasicTurbulenceModel::transportModel transportModel;


    //- Runtime type information
    TypeName("kklARSM");


    // Constructors

        //- Construct from components
        kklARSM
        (
            const alphaField& alpha,
            const rhoField& rho,
            const volVectorField& U,
            const surfaceScalarField& alphaRhoPhi,
            const surfaceScalarField& phi,
            const transportModel& transport,
            const word& propertiesName = turbulenceModel::propertiesName,
            const word& type = typeName
        );
```

```
//- Destructor
virtual ~kklARSM()
{}


// Member Functions

    //- Return the turbulence kinetic energy
    virtual tmp<volScalarField> k() const
    {
        return tmp<volScalarField>
        (
            new volScalarField
            (
                IOobject
                (
                    "k",
                    this->runTime_.timeName(),
                    this->mesh_
                ),
                this->mesh_,
                dimensionedScalar("0", dimensionSet(0, 2, -2, 0, 0), 0)
            )
```

```
        );
}

//- Return the turbulence kinetic energy dissipation rate
virtual tmp<volScalarField> epsilon() const
{
    return tmp<volScalarField>
    (
        new volScalarField
        (
            IOobject
            (
                "epsilon",
                this->runTime_.timeName(),
                this->mesh_
            ),
            this->mesh_,
            dimensionedScalar("0", dimensionSet(0, 2, -3, 0, 0), 0)
        )
    );
}

//- Return the effective diffusivity for nuTilda
virtual tmp<volScalarField> DnuTildaEff() const;
```

```
        //- Re-read model coefficients if they have changed
        virtual bool read();

        //- Solve the turbulence equations and correct the turbulence viscosity
        virtual void correct();
};


// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

} // End namespace RASModels
} // End namespace Foam

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

#ifdef NoRepository
#   include "kklARSM.C"
#endif

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

#endif
// ************************************************************************* //
```

Curriculum Vita

Tianshu Wen

Degrees **M.S. in Mechanical Engineering**, Washington University in St. Louis, St. Louis, Missouri May 2019

B.S. in Mechanical Engineering, Central Michigan University, Mount Pleasant, Michigan Dec 2016

Publications:

1. **Wen, T**., and Agarwal, R. K. “Development of a One-Equation Algebraic Reynolds Stress Model based on k-kL Closure,” AIAA Aviation and Aeronautics Forum and Exposition, Dallas, TX, 2019.
2. **Wen, T**., and Agarwal, R. K. "A New Extension of Wray-Agarwal Wall Distance free Furbulence Model to Rough Wall Flows,” AIAA SciTech Forum and Exposition, San Diego, CA, 2019.
3. Xi, J., Si, X. A., Peters, S., Nevorski, D., **Wen, T**., and Lehman, M. "Understanding the Mechanisms Underlying Pulsating Aerosol Delivery to the Maxillary Sinus: In Vitro Tests and Computational Simulations," International Journal of Pharmaceutics, 2017.
4. Xi, J., Yang, T., Talaat, K., **Wen, T**., Zhang, Y., Klozik, S., and Peters, S. "Visualization of Local Deposition of Nebulized Aerosols in A Human Upper Respiratory Tract Model,” Journal of Visualization, 2017.