9-1-2015

Computational Analysis of Turbulent Flow Around NACA 4412 Airfoil with Open Source CFD Software

Robert Habbit III

Robert Habbit

Candidate

Mechanical Engineering

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Dr. Svetlana V. Poroseva, Chairperson

Dr. C. Randall Truman

Dr. Peter Vorobieff

COMPUTATIONAL ANALYSIS OF TURBULENT FLOW AROUND NACA 4412 AIRFOIL WITH OPEN SOURCE CFD SOFTWARE

by

ROBERT HABBIT

B.S., MECHANICAL ENGINEERING, UNIVERSITY OF NEW MEXICO, 2013

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

Mechanical Engineering

The University of New Mexico
Albuquerque, New Mexico

July 2015

ACKNOWLEDGEMENTS

I would like to thank my academic advisor, Dr. Svetlana Poroseva, for sharing her passion for fluid dynamics. I would like to thank my committee, not only for helping with my thesis, but it is from your classes that I found the fluid dynamics field. I would like to thank all my friends and family for constantly supporting me through my educational career.

COMPUTATIONAL ANALYSIS OF TURBULENT FLOW AROUND NACA 4412 AIRFOIL WITH OPEN SOURCE CFD SOFTWARE

by

Robert Habbit

B.S., Mechanical Engineering, University of New Mexico, 2013

M.S., Mechanical Engineering, University of New Mexico, 2015

ABSTRACT

Computational Fluid Dynamics (CFD) is a tool utilized in industry and academia to help provide a better understanding of a flow field. It is advantageous to use because building experiments for all different flow scenarios can become extremely expensive. Also, there is no need to disturb or add particles to the flow field in order to make measurements like many experimental methods require. CFD does have its own flaws though. The simulations can be extremely time extensive and expensive whether it be the code itself or the processing unit chosen in which to run the simulations. An additional major problem with CFD is the difficulty in obtaining the same or even similar results when using a model across different codes. This has brought us to our first goal of validating OpenFOAM. OpenFOAM is the CFD tool chosen in this research because of its open-source nature allowing it to be free to the general public and allowing for the implementation of different models. In order to validate OpenFOAM two eddy viscosity models will be used to simulate the flow around a NACA 4412 airfoil. The NACA 4412 airfoil was chosen because it is a flow of interest that when ran at its critical angle of

attack a separation bubble forms at the trailing edge of the airfoil. This separation bubble is a common occurrence in everyday flows around any bluff body or streamlined body that is not place perfectly parallel to the flow direction. The NACA 4412 flow field is also a benchmark case provided by NASA for fluid dynamic researchers to validate their computational platform.

The second goal of this research is to implement a Reynolds Stress Transport Model (RSTM) in OpenFOAM. These types of models make less assumptions than the aforementioned eddy viscosity models, and therefore should be able to predict the flow field with a higher order of accuracy. If we can prove that RSTM models capture the physics surrounding the separation bubble more accurately, then potential arises to look into even higher order models that previously have been pushed aside due to the lack of computing power.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. Introduction

The four most common states of matter are solids, liquids, gases, and plasmas. Out of these four states, we are only able to describe the motion of solids with extreme accuracy. The physics surrounding the motion of the other three states: gases, liquids, and plasmas, is still not completely understood. These three states will be clumped together and called fluids while other categories will be made based on where certain physical phenomena or traits are constant within a set regime. For example, fluid flows can be described as laminar, turbulent or transitional; incompressible or compressible; subsonic, supersonic or hypersonic. In order to distinguish a flow as being laminar, transitional, or turbulent one must calculate the Reynolds number for the flow. To characterize a flow as being compressible, incompressible, subsonic, supersonic, or hypersonic one must analyze the Mach number at which the fluid is moving. Both the Reynold's number and the Mach number are extremely important characteristics of defining a flow. The Reynold's number was derived by Osbourne Reynolds an English physicist, who despite not being the first one to analyze turbulent flows, was the first one to mathematically characterize them. The Mach number is named after an Austrian physicist, who came with a dimensionless way to describe a fluids speed with respect to the speed of sound. For the current research only incompressible and turbulent flows will be analyzed.

Turbulent flows are one of the final frontiers that have yet to be solved in classical physics. The reason this physical phenomena is so difficult to characterize and solve is due to its chaotic nature. Despite this chaotic nature physcists Claude-Louis Navier and George Gabriel Stokes derived the Navier-Stokes equations by applying Newton's second law to fluid motion. Also an assumption that the stress in the fluid is the sum of a diffusing viscous term and a pressure term must be made. The Navier-Stokes equations in their incompressible form are presented below:

$$\rho\left[\frac{\partial u}{\partial t}+u_j\frac{\partial u_i}{\partial x_j}\right]=-\frac{dP}{dx_i}+\mu\left(\frac{\partial^2 u_i}{\partial x_j^2}\right). \qquad (1)$$

$$\frac{\partial u_i}{\partial x_i}=0 \qquad (2)$$

It should be noted that additional terms can be added to the above equation if there are gravitational effects, magnetic effects, or any other body forces acting on the fluid. In order to solve these equations, it is necessary to use high end supercomputers, however, they can still have trouble solving these equations. The problem arises in breaking down the fluid domain and time steps taken into small enough scales so that all of the turbulent quantities are being captured. These small scale quantities are the limiting factor of how large of a domain can be directly solved for. People have attempted to come up with models for the smaller scales of turbulence assuming they are very similar to each other and use the Navier-Stokes equations for the largest of scales, this method is called Large Eddy Simulations. Although this method runs much faster usually within one to two months, it is still not an applicable method for design optimization in industry in which cars and planes are being designed within a year. The method that will be utilized in this paper is the Reynolds averaging of the Navier-Stokes equations (RANS).

Despite all of the different methods that have been developed to solve and model fluid dynamic problems not every model is applicable for every type of flow. This requires the user to pick the right model for the application and implement it correctly in the program. Theoretically the same model in different programs should yield the same flow field of a given problem, however, this has not always been found to be the case

[1,2]. Because of this problem, validation of the models in different programs is necessary. Verification of models require a reference solution obtained with highly accurate numerical methods on benchmark problems. The goal of this work is to verify and validate a number of turbulence models using the Open Field Operation and Manipulation (OpenFOAM) computational toolbox [3]. Specific models implemented in OpenFOAM will be validated and verified by comparing them with reference results obtained by NASA's high-order codes, CFL3D [4] and FUN3D [5], Large Eddy Simulations (LES), and experimental measurements.

a. Turbulence Modeling

*******This section contains information from [6] and [7].

The previously mentioned Navier-Stokes equations define the motion of a fluid. The equations are comprised of the conservation of momentum in all three directions and the conservation mass. These equations will be repeated here for convenience:

$$\rho\left(\frac{\partial u_i}{\partial t} + u_k \frac{\partial u_i}{\partial x_k}\right) = -\frac{\partial P}{\partial x_i} + \mu \frac{\partial^2 u_i}{\partial x_k \partial x_k}$$

$$\frac{\partial u_i}{\partial x_i} = 0$$

In the above equations μ is the dynamic viscosity, defined as $\mu = \nu\rho$, and the subscript i represents each component of the corresponding variable. Summation across indices is assumed, unless otherwise stated, as is defined by Einstein notation. The Reynolds averaging is then performed on the Navier-Stokes equations to yield the RANS equation.

This operation involves breaking up the flow velocity into two components; a time-averaged component and an instantaneous fluctuating component:

$$u_i(x_i, t) = \overline{u_\iota}(x_i) + u_i'(x_i, t).$$

In the above decomposition the time-averaged velocity of a steady flow is defined as

$$\overline{u_\iota}(x_i) = \lim_{T\to\infty} \frac{1}{T} \int_t^{t+T} u_i(x_i, t) dt$$

where T is the averaging interval, which must be much larger than the time scale of the velocity fluctuations.

Now the Reynolds averaging is applied to the Navier-Stokes equations. Firstly, the incompressible continuity equation becomes

$$\frac{\partial \overline{u_J}}{\partial x_j} = 0. \tag{3}$$

Secondly, applying the Reynolds averaging to the conservation of momentum equation yields

$$\rho \left[\frac{\partial(\overline{u_\iota} + u_i')}{\partial t} + \frac{\partial((\bar{u}_i + u_i')(\bar{u}_j + u_j'))}{\partial x_j} \right] = -\frac{d(\bar{p} + p')}{dx_i} + \mu \left(\frac{\partial^2(\bar{u}_i + u_i'}{\partial x_j^2} \right). \tag{4}$$

After breaking up the velocity and pressure into their mean and fluctuating components, a time average is taken. The time average of fluctuations are zero, but not of products of fluctuations. Also the average of an average still yields the average. After expanding Eq.4 and eliminating the averaging of fluctuations the momentum equation becomes

$$\rho \left[\frac{\partial \overline{u_\iota}}{\partial t} + \bar{u}_j \frac{\partial(\bar{u}_i)}{\partial x_j} \right] = -\frac{d(\bar{p})}{dx_i} + \mu \left(\frac{\partial^2 \bar{u}_i}{\partial x_j^2} \right) - \rho \frac{\partial(\overline{u_\iota' u_J'})}{\partial x_j}. \tag{5}$$

In Newtonian fluids, the second to last term in Eq.5 is represented as a viscous stress tensor, defined as

$$\tau_{ij} = 2\mu S_{ij} \tag{6}$$

where S_{ij} is the strain-rate tensor

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

Substituting Eq.6 into Eq.5 yields the conservation of momentum for the Reynolds-Averaged Navier-Stokes equations

$$\rho\left[\frac{\partial \overline{u_i}}{\partial t} + \bar{u}_j \frac{\partial(\bar{u}_i)}{\partial x_j}\right] = -\frac{d(\bar{p})}{dx_i} + \frac{\partial}{\partial x_j}\left(2\mu S_{ij} - \rho\overline{u_i' u_j'}\right). \tag{7}$$

The quantity $-\rho\overline{u_i' u_j'}$ in Eq.7 is known as the Reynolds stress tensor. The tensor is symmetric, therefore only six components need to be solved for instead of all nine. With this we find a total of ten unknowns for three dimensional flow: three velocity components, pressure, and the six independent components of the Reynolds stress tensor. A problem arises in only having four equations, continuity and conservation of momentum in three directions, and ten unknowns. The need for additional equations is called the turbulence closure problem. In order to compute all mean-flow propertied of the turbulent flow, it is necessary to evaluate and compute the Reynolds stress tensor.

Types of Models

<u>Eddy Viscosity Models</u>

One very common method to describe the Reynolds stresses is to model it using the Boussinesq eddy viscosity approximation developed by Joseph Boussinesq in 1887. Boussinesq postulated that the momentum transfer caused by turbulent eddies can be modeled with an eddy viscosity [8]. The eddy viscosity, or turbulent viscosity is computed from a mixing length based on the flow. The use of an eddy viscosity, μ_t, assumes isotropic flow, which at times can lead to excessive diffusion [9]. The definition of the eddy viscosity varies depending on which model is implemented. A more descriptive evaluation of the eddy viscosity for our models will be shown in Section II. The Boussinesq approximation defines the Reynolds stress term from the conservation of momentum in the RANS equations as follows:

$$-\rho\overline{u_i'u_j'} = 2\mu_t S_{ij} - \frac{2}{3}k\rho\delta_{ij}\,. \tag{8}$$

The k in Eq.8 represents the turbulent kinetic energy, and δ_{ij} is the kronecker delta function. This model for the Reynolds stresses reduces the tensor of six unknowns down to one unknown, k. The kinetic energy can be directly related to the Reynolds stress by

$$k = \frac{1}{2}\rho\overline{u_i'u_i'}.$$

Although the Boussinesq approximation helps with the turbulent closure problem, we now have four equations and five unknowns. Two types of eddy viscosity models have been developed to accommodate this problem, one and two equation models. One equation models usually develop a transport equation for turbulent kinetic energy, where two equation models solve a transport equation of the turbulent kinetic energy as well as an additional turbulent variable. One example of an additional variable is ω, the specific turbulent dissipation rate. The specific turbulent dissipation rate represents how quickly the turbulent kinetic energy is dissipated or converted into thermal energy in the flow. Sometimes the specific turbulent dissipation is referred to as the mean frequency of the turbulence, because of its units s^{-1} [10].

The transport equation for the turbulent kinetic energy is derived from multiplying the conservation of momentum equation by u_i and time averaging the result. Due to the exact derivation of the transport equation for turbulent kinetic energy being covered in most fluid dynamic textbooks, only the final equation is shown[6]:

$$\frac{\partial k}{\partial t} + \overline{u}_j \frac{\partial k}{\partial x_j} = \underbrace{-\overline{u_i' u_j'} \frac{\partial \overline{u_i}}{\partial x_j}}_{P_k} - \underbrace{\nu \overline{\left(\frac{\partial u_i'}{\partial x_j}\right)^2}}_{\varepsilon} \underbrace{- \frac{\partial}{\partial x_j}\left[\overline{u_i' u_i' u_j'}\frac{1}{2} + \frac{1}{\rho}\overline{p' u_j'}\right] + \overline{\frac{\partial}{\partial x_j} \nu \left(\frac{\partial k}{\partial x_j}\right)}}_{D_k}. \qquad (9)$$

The left hand side of the equation is described by the convective terms and the unsteady terms. On the right hand side the first term P_k, is known as production, representing the rate at which the kinetic energy is transferred from the mean flow to turbulence, or fluctuations. The second term ε, is the dissipation rate for which the kinetic energy is converted to thermal energy. Lastly, D_k is the diffusive transport, made up of three components: molecular diffusion, pressure diffusion and the triple velocity correlation. Molecular diffusion represents the diffusion by the fluid's natural molecular transport. Pressure diffusion represents diffusion through the pressure-velocity fluctuations. The triple velocity correlation represents the transport of turbulence through the fluctuations. To be able to close Eq.9, the Reynolds stresses, dissipation, turbulent transport, and pressure diffusion have to be modeled.

The Reynolds stresses are modeled through the Boussinesq approximation as defined in Eq.8. It should be noted that a common way to write the Reynolds stress in a shorter notation is τ_{ij}. The dissipation model is commonly defined as

$$\varepsilon = C_D \frac{k^{3/2}}{\ell}, \tag{10}$$

although it can vary depending on the model [11], where C_D is a closure coefficient that commonly takes on the value of 0.09 [6] and ℓ is the length scale of the turbulent structures. Both diffusive terms are traditionally modeled as a single term due to the complexity of both terms and there being no good physical idea of how they these terms interact with each other. They are modeled in the following way:

$$\overline{u_i' u_i' u_j'}\frac{1}{2} + \frac{1}{\rho}\overline{p' u_j'} = -\frac{\nu_t}{\sigma_k}\frac{\partial k}{\partial x_j}. \tag{11}$$

In the above equation, ν_t is the turbulent eddy viscosity and σ_k is another closure coefficient known as the turbulent Prandtl number. The combination of these models Eq.8, 10 and 11 into Eq.9 yields the modeled turbulent kinetic energy equation:

$$\frac{\partial k}{\partial t} + \overline{u_j}\frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right) + \tau_{ij}\frac{\partial \overline{u_i}}{\partial x_j} - C_D\frac{k^{3/2}}{\ell}. \tag{12}$$

The modeling that was required to close the system leads to a significant loss in physical meaning and detail, however it is necessary to make the system solvable.

Reynolds Stress Transport Models

Reynolds Stress Transport Models (RSTMs) are more complicated and computationally more intensive that EVMs. This is because they solve for all six components of the Reynolds stress tensor instead of modeling it through a scalar. The turbulence closure method for RSTMs is called second-order closure. This comes from closing the statistical second moment of the velocity fluctuations, or product of velocity fluctuations. Some of the many benefits to calculating the components of the Reynolds stress tensor are accounting for directional effects of the Reynolds stress fields, allowing for a more accurate physical representation of the flow field, and capturing sudden changes in strain rate. Despite all of the benefits RSTMs are not commonly used in industry because they are very expensive and they are numerically much more unstable. It is important to use second-moment closures for turbulent shear flows because they are not describable by a model based on a linear eddy viscosity model [12].

The transport equation for the Reynolds stresses is derived from multiplying the conservation of momentum equations by velocity fluctuations, and then adding them together. The process will not be shown here, only the final result [6]

$$\frac{\partial \overline{u_i'u_j'}}{\partial t} + \bar{u}_k \frac{\partial \overline{u_i'u_j'}}{\partial x_k} = \underbrace{-\overline{u_j'u_k'}\frac{\partial \bar{u}_i}{\partial x_k} - \overline{u_i'u_k'}\frac{\partial \bar{u}_j}{\partial x_k}}_{P_{ij}} + \underbrace{\frac{\partial}{\partial x_k}\left[\nu \frac{\partial \overline{u_i'u_j'}}{\partial x_k} - \overline{u_i'u_j'u_k'}\right]}_{D_{ij}}$$

$$-\underbrace{2\nu \overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_j'}{\partial x_k}}}_{\varepsilon_{ij}} - \underbrace{\frac{1}{\rho}\left(\overline{u_i'\frac{\partial p'}{\partial x_k}} + \overline{u_j'\frac{\partial p'}{\partial x_i}}\right)}_{\Pi_{ij}}. \tag{13}$$

The terms on the left hand side of the equation account for the unsteady and convective changes of the Reynolds stresses. On the right hand side there are production P_{ij}, diffusion D_{ij}, dissipation ε_{ij}, and fluctuating pressure Π_{ij}. All of the terms beside production on the right hand side need to be modeled. The molecular viscous part of the diffusion term can be solved for directly, where the triple product or turbulent transport is modeled through the use of the generalized gradient diffusion hypothesis (GGDH) developed by Daly and Harlow [13]. The GGDH approximation takes the following form:

$$\overline{u_i'u_j'u_k'} = -c_s \frac{k}{\varepsilon} \overline{u_k'u_l'} \frac{\partial \overline{u_i'u_j'}}{\partial x_l} , \tag{14}$$

where c_s is a model constant with a value of 0.2. The dissipation term is commonly treated very similar to how it is in two equation models. It will be treated as if it is isotropic, and the non-isotropic parts will be modeled in the turbulent parts of the fluctuating pressure [14]. The typical isotropic approximation of the dissipation tensor is defined as

$$\varepsilon_{ij} = \frac{2}{3}\, \varepsilon \delta_{ij} , \tag{15}$$

where ε is the dissipation rate and is determined from its own transport equation.

The fluctuating pressure term is traditionally decomposed into two parts:

$$-\frac{1}{\rho}\left(\overline{u_i' \frac{\partial p'}{\partial x_k}} + \overline{u_j' \frac{\partial p'}{\partial x_i}}\right) = \underbrace{-\frac{\partial}{\partial x_k}\left[\overline{p'u_i'}\delta_{jk} + \overline{p'u_j'}\delta_{ik}\right]}_{D_{ij}^P} + \underbrace{\frac{1}{\rho}\overline{p'\left[\frac{\partial u_i'}{dx_j} + \frac{\partial u_j'}{dx_i}\right]}}_{\phi_{ij}} . \tag{16}$$

The first component D_{ij}^P is the pressure diffusion, which describes the diffusion of the Reynolds stress due to the pressure fluctuations. There is no explicit model for this term and is frequently clumped with the diffusive component. The second component is the pressure-rate-of-strain tensor ϕ_{ij}. This term is modeled differently depending on which RSTM is chosen. In this paper the pressure-rate-of-strain tensor will be decomposed into four different parts:

$$\phi_{ij} = \phi_{ij_1} + \phi_{ij_2} + \left(\phi_{ij}^{w1} + \phi_{ij}^{w2}\right) f(x_n).$$

The first term ϕ_{ij_1} is called the "slow" or Rotta term, which represents the return to isotropy of non-isotropic turbulence [15], which is represented as follows:

$$\phi_{ij_1} = -\frac{c_1 \varepsilon}{k}\left(\overline{u_i' u_j'} - \frac{2}{3} k \delta_{ij}\right). \tag{17}$$

The second term ϕ_{ij_2} is called the "rapid" term, which represents the isotropization of strain production [16], which is represented as follows:

$$\phi_{ij_2} = -c_2 \left(P_{ij} - \frac{1}{3} P_{kk} \delta_{ij}\right). \tag{18}$$

The third and fourth terms are designed to help with near wall effects with ϕ_{ij}^{w1} being developed in [17] and ϕ_{ij}^{w2} being developed in [18]. Their definitions are

$$\Phi_{ij}^{w1} = C_1^w \left[R_{km} n_k n_m \delta_{ij} - \frac{3}{2}(R_{ki} n_k n_j + R_{kj} n_k n_i)\right], \tag{19}$$

and

$$\Phi_{ij}^{w2} = C_2^w \left[R_{2km} n_k n_m \delta_{ij} - \frac{3}{2}(R_{2ki} n_k n_j + R_{2kj} n_k n_i)\right], \tag{20}$$

where R_{ij} is the Reynolds stress tensor and n represents the direction normal to the wall, also summation over repeated k indices is not implied in Eqs. 19 and 20. The damping function $f(x_n)$, is defined as:

$$f(x_n) = \frac{1}{5} \frac{k^{\frac{3}{2}}}{\varepsilon x_n}$$

where x_n is the normal distance away from the wall [19]. In all of these models the C's represent model coefficients that can be altered or tailored for different flows. This concludes a general description of how various approaches have been used to solve the RANS equations.

b. Literature Review

Because of extreme variation in airfoil design and shape only the NACA 4412 airfoil will be analyzed in the following document. The flow around the NACA 4412 airfoil is a popular flow that many researchers have studied. Both experimental and computational analyses have been performed on the NACA 4412 airfoil. The earliest experimental analysis is that of Pinkerton [20]. He performed two different studies with the first analyzing the pressure distribution across the airfoil for multiple angles of attack ranging from -20° to 30° at a set Reynold's number of three million. The second was a similar study that also analyzed the effect of Reynolds number ranging from 1×10^5 to 8.2×10^6 [21]. More recently a Master's Thesis by Tsung-Ju Wu from the University of Texas at Arlington also analyzed the NACA 4412 airfoil using a smoke-wire technique [22]. He only analyzed the lift coefficient for a few different angles of attack though. There is a journal paper by Wadcock and Coles [23] in which the primary results of the flow around the NACA 4412 airfoil are presented. There is a more complete presentation of the experiment in a NASA contractor report by Wadcock [24] in which the separation bubble is examined in great detail using hot wires. These first two reports claim a maximum lift angle of 13.87°. In following years, Hastings and Williams [25] and Wadcock [26] once again repeated the experiment using slightly different flow parameters and using laser anemometry to analyze the flow field and found a maximum lift angle closer to 12°. There are also a couple of publications at a lower Reynolds number of 3.6×10^5 by Bradran [27] and Bradran and Bruun [28]. Although this concludes the experimental analyses of the NACA 4412 airfoil that are directly relevant to the research presented in this paper, there are other experiments on the NACA 4412 airfoil. For example, the research by Chang involves an oscillating airfoil [29]. Also there are experimental studies of the effects of a gurney flap on the NACA 4412 airfoil like that by Storms and Jang [30]. This concludes the research done to experimentally analyze the NACA 4412 airfoil.

Many more researchers have analyzed the flow field around the NACA 4412 airfoil utilizing computational methods. Similar to the experimental analyses of the NACA 4412 airfoil, not all the computational research regarding the NACA 4412 airfoil is strictly relevant to the research that will be presented in this document. There is a similar numerical approach to analyze the effect of the gurney flap [31]. As with the gurney flap investigations, there are computational analyses of an oscillating airfoil [32]. Many people from around the world have analyzed the NACA 4412 airfoil in a similar flow regime as what will be presented in this document, however they all seem to differ slightly in the Reynolds number or Mach number chosen. There are many similar papers with the main differences coming in which CFD tool the researcher chose. Fluent is a popular CFD tool of choice as it is used in [33], [34], and [35]. Another CFD tool of choice was OVERFLOW, which Rosen chose for his Master's thesis [36]. Although these prior studies will not be repeated or used for comparison in this study it is important to know all of the relevant research. The computational research that will be used for comparison in this study consists of Large Eddy Simulations (LES) and different RANS models. There are two separate research analyses of the flow field around the NACA 4412 airfoil near maximum lift utilizing LES. The first by Jansen where early studies are presented in [37] and [38], then the results are compiled into a final report [39]. The second LES study is comprised of two different papers by the same researchers in which comparison to the two separate experimental analyses are performed. The first paper by Park and Moin [40] matches the experimental data by Hastings [25] and Wadcock's second paper [26], along with the LES data by Jansen [39]. The second paper by Park and Moin [41] matches the first experimental data [23] and [24] in which the peak coefficient of lift was found to be 13.87°.

Despite all this research surrounding the NACA 4412 airfoil, it is still not guaranteed that if a turbulence model is implemented in different CFD codes the same result will be obtained. In an effort to improve consistency, verification, and validation of turbulence models, NASA has established a website to provide a central location for the documentation of RANS turbulence models [42]. The website, Turbulence Modeling Resource (TMR), is a collaboration between NASA's Langley Research Center and the Turbulence Model Benchmarking Working Group (TMBWG) [43]. TMBWG is a working group of the Fluid Dynamics Technical Committee [44] of the American Institute of Aeronautics and Astronautics (AIAA) [45]. One of the test cases on the TMR website is the flow field around the NACA 4412 airfoil. The experimental data they chose for comparison is that of Wadcock [23] and [24]. The objective of the website is to provide a resource for CFD developers to:

- Obtain accurate and up-to-date information of widely used RANS turbulence models.

- Verify that models are implemented correctly.

II. Simulation Parameters

a. OpenFOAM

The computational simulations that will be discussed in Section III were conducted on NASA's Pleiades Supercomputer [46] using OpenFOAM. Details about Pleiades can be found in Appendix A.

OpenFOAM is a free, open source CFD software, licensed and distributed by the OpenFOAM Foundation [47] and developed by OpenCFD Ltd [48]. OpenFOAM is a powerful tool utilized by those in industry and in academia for solving complex fluid dynamic, chemical reaction, and heat transfer problems, which are all boundary value problems that require solving a set of differential equations. The major appeal of the software is it being based in C++, which coupled with its open source nature, allows users to have complete freedom in changing anything they desire. Also the operations defined in OpenFOAM are built so that one can run the simulations in parallel, thereby greatly increasing the efficiency of the software. The software is built so that everything is defined by separate directories and subdirectories. The folder system for a single simulation is comprised of three directories. First, you have the system directory, where the numerical solution procedure is chosen. Second, there is the constant directory, where the mesh and physical properties are defined. Lastly, the zero directory contains the initial and boundary conditions for the simulation. It is from the zero directory that the solution will be built from at the desired output times.

b. Turbulence Models

This section describes the turbulence models that were used in this study. The model equations will be described in full along with the model coefficients used. Two different Eddy Viscosity Models (EVMs) were used along with a Reynolds Stress Transport Model (RSTM). The two different EVMs are Wilcox's 2006 version of the k-ω model [49][50] and the Menter Shear Stress Transport (SST) model [51]. The RSTM chosen is a version of the Launder-Reece-Rodi Isotropization of production model [52]. The equations implemented in OpenFOAM did not perfectly match the definitions found on NASA's Turbulence Modeling Resource website [53]. Therefore, the source code has been altered so that the model equations perfectly match their definitions so that an accurate comparison can be made. The source code for all models can be found in Appendix B.

k-ω Model

The Wilcox 2006 version of the k-ω model was used in this study [49][50]. The model is a two equation model which solves transport equations for the turbulent kinetic energy, and the transport of the specific dissipation rate. The incompressible transport equations for k and ω are

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

and

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

where

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

$$\tau_{ij} = 2\nu_t S_{ij} - \frac{2}{3}k\delta_{ij}$$

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

$$\nu_t = \frac{k}{\widetilde{\omega}}$$

$$\widetilde{\omega} = max\left[\omega, C_{lim}\sqrt{\frac{2S_{ij}S_{ij}}{\beta^*}}\right]$$

$$\sigma_d = \begin{cases} 0, \frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} \leq 0 \\ \frac{1}{8}, \frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} > 0 \end{cases}$$

$$f_\beta = \frac{1 + 85\chi_\omega}{1 + 100\chi_\omega}$$

$$\chi_\omega = \left|\frac{\Omega_{ij}\Omega_{jk}S_{ki}}{(\beta^*\omega)^3}\right|$$

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial \overline{u_i}}{\partial x_j} - \frac{\partial \overline{u_j}}{\partial x_i}\right).$$

It should be noted that in two-dimensional flows, as is in our case, χ_ω needs to bet set to zero. The following table shows the model coefficients for the k-ω model.

$\boldsymbol{\sigma_k}$	$\boldsymbol{\sigma_\omega}$	$\boldsymbol{\beta^*}$	$\boldsymbol{\gamma}$	$\boldsymbol{C_{lim}}$	$\boldsymbol{\beta}$	$\boldsymbol{\beta_0}$
0.6	0.5	0.09	13/25	7/8	$\beta_0 f_\beta$	0.0708

Shear Stress Transport Model

The Menter 2003 Shear Stress Transport [51] version of the shear stress transport model was used in this study. This model will be referred as the SST model throughout the rest of the paper. This model is similar to the k-ω model in that the two turbulent transport equations the model attempts to resolve are the turbulent kinetic energy and specific dissipation rate equations.

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

$$\frac{\partial \omega}{\partial t} + \bar{u}_j \frac{\partial \omega}{\partial x_j} = \frac{\gamma_\omega}{\nu_t} P - \beta \omega^2 + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_\omega \nu_t)\frac{\partial \omega}{\partial x_j}\right] + 2(1 - F_1)\frac{\sigma_{\omega 2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}$$

$$P^* = \min(P, 20\beta^* \omega k)$$

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

$$\tau_{ij} = 2\nu_t S_{ij} - \frac{2}{3}\rho k \delta_{ij}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)$$

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

$$\Omega = \sqrt{2W_{ij}W_{ij}}$$

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

$$F_2 = \tanh({arg_2}^2)$$

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

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

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

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

The variable y in the above equations stands for the distance to the first grid point away from the wall. In order for the SST model to blend the k-ω and the k-ϵ models together a blending function is required, which takes the following form:

$$\varphi = F_1\varphi_1 + (1 - F_1)\varphi_2.$$

In the above equation the φ_1 and the φ_2 constants will be described by coefficients with matching subscripts. For example the φ_1 and φ_2 terms would be replaced by the β_1 and β_2 terms respectively when trying to calculate the β coefficient to input in the specific dissipation equation. The remaining model coefficients can be found below:

$$\gamma 1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1} k^2}{\sqrt{\beta^*}} \quad and \quad \gamma 2 = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2} k^2}{\sqrt{\beta^*}}$$

σ_{k1}	σ_{k2}	$\sigma_{\omega 1}$	$\sigma_{\omega 2}$	β_1	β_2	β^*	κ	a_1
0.85	1	0.5	0.856	0.075	0.0828	009	0.41	0.31

Launder-Reece-Rodi Isotropization-of-Production Model

The LRR-IP [52] model attempts to resolve the Reynolds stresses fluctuations. The LRR-IP model also requires an additional transport equation for the scalar dissipation rate. The transport equation for the Reynolds stresses will be written in its tensor form as follows.

$$\frac{\partial R_{ij}}{\partial t} + U_k \frac{\partial R_{ij}}{\partial x_k} = P_{ij} + D_{ij} - \varepsilon_{ij} + \Phi_{ij1} + \Phi_{ij2} + \left(\Phi_{ij}^{w1} + \Phi_{ij}^{w2}\right) f(x_n)$$

$$\frac{\partial \varepsilon}{\partial t} + U_i \frac{\partial \varepsilon}{\partial x_i} = \frac{\partial}{\partial x_k}\left[\left(\nu \delta_{kl} + C_\varepsilon \frac{k}{\varepsilon} R_{kl}\right)\frac{\partial \varepsilon}{\partial x_l}\right] + \frac{\varepsilon}{k}\left(C_{\varepsilon 1} P - C_{\varepsilon 2}^{*} \varepsilon\right) - \frac{2\nu\varepsilon}{x_n^2} f_1$$

$$P = \frac{1}{2} P_{kk}$$

$$C_{\varepsilon 2}^{*} = C_{\varepsilon 2} f_2$$

$$f_1 = e^{\left[-\frac{x_n u_\tau}{2\nu}\right]}$$

$$f_2 = 1 - \frac{2}{9} e^{\left[-\left(\frac{k^2}{6\nu\varepsilon}\right)^2\right]}$$

$$P_{ij} = -\left(R_{ik} \frac{\partial U_j}{\partial x_k} + R_{jk} \frac{\partial U_i}{\partial x_k}\right)$$

$$D_{ij} = \frac{\partial}{\partial x_k}\left[\left(\nu \delta_{kj} + C_s \frac{k}{\varepsilon} R_{kl}\right)\frac{\partial R_{ij}}{\partial x_l}\right]$$

$$\varepsilon_{ij} = \frac{2}{3} \varepsilon \delta_{ij} + 2\nu \frac{R_{ij}}{x_n^2}$$

$$\Phi_{ij1} = -C_1 \left(\frac{\varepsilon}{k} Rij - \frac{2}{3} \delta_{ij} \varepsilon\right)$$

$$\Phi_{ij2} = -C_2 \left(P_{ij} - \frac{1}{3} P_{kk} \delta_{ij} \right)$$

$$\Phi_{ij}^{w1} = C_1^w \left[R_{km} n_k n_m \delta_{ij} - \frac{2}{3} (R_{ki} n_k n_j + R_{kj} n_k n_i) \right]$$

$$\Phi_{ij}^{w2} = C_2^w \left[R_{2km} n_k n_m \delta_{ij} - \frac{2}{3} (R_{2ki} n_k n_j + R_{2kj} n_k n_i) \right]$$

$$f(x_n) = \frac{1}{5} \frac{k^{\frac{3}{2}}}{\varepsilon x_n}$$

$$k = \frac{1}{2} R_{ij} \delta_{ij}$$

C_ε	C_s	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	C_1	C_2	C_1^w	C_2^w
0.15	0.2	1.44	1.92	1.8	0.6	0.3	0.3

c. Numerical Methods

OpenFOAM utilizes the finite volume method in order to solve the set of partial differential equations. The finite volume method requires the domain to be divided up into smaller volume pieces. This allows for the terms to be evaluated as fluxes through each volume piece, and due to the law of conservation the flux entering must be equal to that leaving.

The OpenFOAM solver chosen for the problem is the semi-implicit method for pressure-linked equations (SIMPLE) algorithm. The SIMPLE algorithm solves iteratively for the velocity and pressure fields from predefined initial conditions in steady flow simulations. The exact order of operations taken by the SIMPLE algorithm are as follows[55]:

1. Set the boundary conditions.
2. Compute the gradients of velocity and pressure.

3. Solve the discretized momentum equation to compute the intermediate velocity field.
4. Compute the uncorrected mass fluxes at cell faces.
5. Solve the pressure correction equation to produce new/corrected pressure values.
6. Update the pressure field using an under-relaxation factor.
7. Update the boundary values using the pressure corrections.
8. Correct the face mass fluxes.
9. Calculate corrected cell velocities using the pressure gradient of the pressure corrections.

The discretization scheme applied is the second-order Gaussian integration scheme. The interpolation schemes are first order linear approximations, except for the divergence scheme which uses upwind approximations. For the Laplacian scheme, the surface normal gradient scheme is chosen, which is a corrected unbounded, second order, conservative scheme.

d. Computational Domain

The computational domain used for the NACA 4412 simulations is a structured C-type mesh (Fig. 1), which was generated by NASA and is available at Turbulence Model Benchmarking Working Group's website[61]. The far field boundaries of the computational domain used in flow simulations are a hundred times the chord length away. Five grids with different refinement are used in these simulations: 113×33 with 65 points placed along the airfoil surface, 225×65 with 129 points along the airfoil surface, 449×129 with 257 points along the airfoil surface, and 897×257 with 513 points along the airfoil surface. The finest grid has the first computational node at y+ between 0.2 and 0.4. The grids are of a nested type, meaning each coarser grid corresponds to every other point from the previous finer grid. The grids obtained from NASA are two-dimensional, therefore, the grid points must be duplicated one unit away to create volume sections. These front and back faces have the fluxes set to zero in order to keep the simulations two-dimensional.

e. Boundary & Initial Conditions

Two separate sets of flow parameters were used. The first set contains the same parameters as in the simulations conducted at NASA Langley [61] and in the experiments conducted by Wadcock and Coles [23][24]. For this set, the inlet velocity magnitude corresponds to M = 0.09 and viscosity defined from the simulation at Re = 1.52×10^6. The velocity direction is at $\alpha = 13.87^\circ$ to reproduce the angle of attack from these experiments. The second set of flow parameters is based on more recent experiments conducted by Wadcock [26]: M = 0.085, Re = 1.64×10^6 and $\alpha = 12^\circ$. The initial internal velocity field is the same as at the inlet. The outlet boundary conditions are set to be a zero gauge pressure outlet. This is also the initial conditions for the internal pressure field. On the airfoil surface, the no-slip boundary condition is applied. The initial condition for the turbulent kinetic energy is:

$$k_{farfield} = \frac{3}{20}(U_\infty I)^2$$

The initial conditions for the specific dissipation rate are:

$$\omega_{wall} = 10\frac{6\nu}{\beta_1 y^2}, \qquad \omega_{farfield} = 1\times10^{-6}\frac{\rho_\infty a_\infty^2}{\mu_\infty}$$

III. RESULTS & DISCUSSION

a. Eddy Viscosity Models

Results of the OpenFOAM simulations will now be presented. The results obtained with the two-equation eddy viscosity models will be compared with those obtained by NASA with their high fidelity codes CFL3D and FUN3D, also they will be compared with data obtained through LES simulations. Firstly, a sensitivity analysis of the simulation results to the grid refinement was performed for the NACA 4412 airfoil. Four different grids and two turbulence models were used in this study. Convergence of simulation results were achieved with the Wilcox 2006 k-ω model and the Menter SST model. Figures 2 and 3 show the convergence of the streamwise velocity profile at all six locations, while figures 4 and 5 show the transverse velocity components for the k-ω model and the Menter SST model respectively. As can be seen by the second finest grid the velocity profiles appear to be sufficiently converged.

Figures 6 and 7 show the results of a sensitivity analysis on the grid density for the coefficient of pressure and the skin friction coefficient for the k-ω model and the Menter SST model respectively. Notice how for the coefficient of pressure and the skin friction coefficient there is a much smaller variation between grid refinements. The only differences occur near the trailing edge where the flow physics of the separation bubble still needs to converge.

Despite the velocity profiles and both the friction coefficient and pressure coefficient profiles seeming converged, if the convergence of these values at specific points is analyzed one can see the flow field is not yet completely converged by the second finest grid. Figures 8 and 9 show how the characteristic mesh length for the constant domain size, *h*, is used to analyze the grid resolution effects on the solutions for each the k-ω model and the Menter SST model respectively. The skin friction coefficient and the coefficient of pressure were analyzed at two points, one towards the leading edge of the airfoil and another in the recirculation zone at the trailing edge of the airfoil. The second finest grid would typically suffice for convergence criteria for obtaining a flow field of sufficient accuracy for most engineering applications. However, because we are trying to validate the implementation of turbulence models in OpenFOAM, it is necessary to make sure no discrepancies are caused by the difference in grid refinements.

For the NACA 4412 airfoil, data obtained with the two equation turbulence models implemented in OpenFOAM were compared with the data from Ref. 61 obtained with the CFL3D and FUN3D codes. Results are shown in Figs. 10-12. Figures 10 and 11 show streamwise and normal velocities, and Fig. 12 demonstrate variations of the pressure coefficient along the airfoil surface. In Figures 10-12 data with OpenFOAM are obtained on the fine grid 897×257 which is the same as that on which the CFL3D and FUN3D data were computed. The color schemes used in Figs. 10 and 11 correspond to the position at which a line probe was placed extending perpendicular from the surface. Line probes were located at the positions where experimental and CFL3D and FUN3D data are available. In the figures, symbols show experimental data at different locations, solids lines are results of simulations with OpenFOAM, and dashed lines are data from Ref. 61.

The results obtained with the Menter SST model converge closer to the velocity profiles of the experimental data, rather than the results obtained with the k-ω model. Both models converged essentially to the results obtained by NASA with their high fidelity codes. The k-ω model is known to struggle in accurately predicting the flow fields near walls, which a part of the separation zone is in. The Menter SST model on the other hand uses a blending function to allow for the k-ε model to be applied in near wall reqions and use the k-ω model outside these regions where the k-ε model has difficulty. By analyzing the streamwise velocity profiles for the k-ω model, you will notice the values never go negative, meaning that this model does not predict a seperation zone along the back of the airfoil. Caution should therefore be used when attempting to model flows in which separation will occur, and if available the Menter SST model should be used.

In addition to analyzing the velocity profiles within the separation region, an analysis of the lift and drag coefficients was performed. Although, accurately knowing the separation region can lead to tremendous design considerations, like the optimal rate for diffusers and the correct angle of attack for airfoils. In the real world at the end of the day the only characteristics of an airfoil you care about are the stall angles, lift coefficients, and drag coefficients. Figure 13 shows variations of the lift coefficient against the angle of attack obtained with the two turbulence models. Menter's SST model underpredicts the lift coefficient, with the tendency increasing with the growth of the angle of attack. Simulations at the angle of attack higher than the stall angle failed. The k-ω model overpredicts both the lift coefficient and the stall angle. The overprediction of the stall angle with the k-ω model, follows from underpredicting the size of the separation bubble.

A second set of flow parameters for the NACA 4412 airfoil based on Wadcock's second experiments [26] are presented in Fig. 14. The simulations were only performed on the finest grid 897×257 for these flow parameters based on the sensitivity study performed previously. Notice how once again the Menter SST turbulence model predicts a flow field that corresponds to both the LES and the experimental data more accurately than the *k*-*ω* turbulence model. Also the *k*-*ω* turbulence model once again predicts a smaller separation bubble than that of the Menter SST turbulence model and the LES data. However, there is still room for improvement for both the *k*-*ω* turbulence model and the Menter SST turbulence model. Overall, the previous [54] and current analyses of the OpenFOAM capabilities in wall-bounded flows show that when a turbulence model is implemented correctly in the software, simulation results are close to those obtained with the NASA codes. Therefore, we will proceed with implementing the Reynolds-stress transport models in OpenFOAM.

b. Reynolds Stress Transport Model

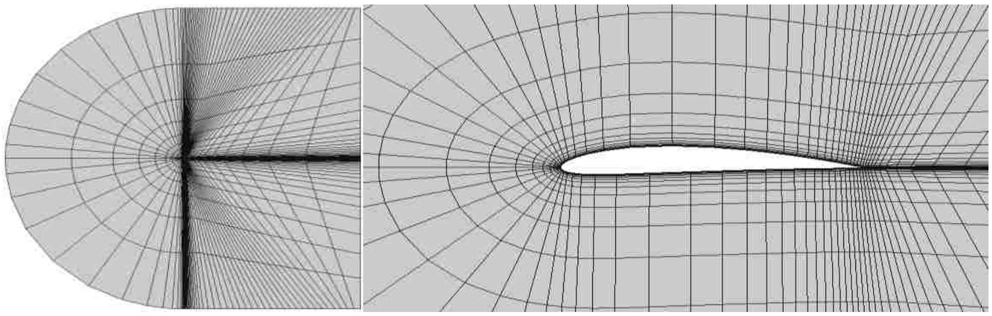
The results from the Launder-Reece-Rodi Isotropization-of-Production model (LRR-IP) will now be presented. All of the results shown were performed on the finest 897×257 grid. The streamwise and transverse velocity profiles are shown in Fig. 15 and the variation of the pressure coefficient is shown in Fig. 16. Because the velocity profiles go negative the model does predict a separation bubble. However, the separation bubble extends higher into the far field, which can be determined by the velocity profiles not getting back to free stream velocity in the scope of the probe lines. Also close to the wall the separation bubble has a smaller magnitude especially when analyzing the transverse velocity profiles. These results indicate that the model has too much turbulent diffusion allowing the turbulent motions to grow farther into free stream conditions. The model might also have too much dissipation seeing as the magnitude of the velocity profiles in the separated region are smaller than what was found with experiments. The coefficient of pressure profile along the airfoil surface seems more accurate however it show the pressure along the top surface of the airfoil is underpredicted. The LRR-IP model did not provide a more accurate representation of the velocity profile around the NACA 4412 airfoil. Although the LRR-IP model did not predict a more accurate flow field it does not entirely prove wrong the initial idea that RSTM's should provide a more accurate flow filed. There are many other RSTM's that still need to be analyzed including some models that use a specific turbulent dissipation rate equation, like the k-ω and SST models, instead of a turbulent dissipation rate equation, like the LRR-IP.

IV. CONCLUSION

Two different types of turbulence models were analyzed the eddy viscosity model and the Reynolds Stress Transport Model. With the correct implementation of the eddy viscosity models we were able to prove the validity of OpenFOAM. Both models got essentially the exact same results as what NASA got when they implemented the turbulence models in their high fidelity CFL3D and FUN3D codes. Analyzing the two eddy viscosity models are their own as their capability in accurately capturing the separation bubble behind an airfoil at almost stall conditions. The Menter SST model captures the physics very well yielding streamwise velocity profiles that almost match the experimental profiles exactly. The transverse velocity profiles were not as accurately captured especially near the trailing edge. This difference could be caused from the simulations being executed in a two-dimensional form. Also, it could be the model under predicting the rotational intensity within the separation bubble, or having too much turbulence dissipation. The k-ω turbulence model does not predict any part of the flow very well. It does not predict a separation bubble because none of the streamwise velocity profiles go negative, however, if you look closely at the trailing edge it is trending in that direction. The same thing goes for the transverse velocity profiles. In comparison the Menter SST turbulence model out preforms the k-ω turbulence model with this flow geometry and conditions.

The LRR-IP model does predict a separation bubble; however it seems to over predict the size indicating that the model has too much diffusion. The Reynolds stress transport model did not provide any more accuracy in capturing the physical phenomena of the separation bubble. One idea is that the Reynolds stress transport model chosen is still based on the turbulent dissipation rate unlike the two eddy viscosity models chosen which were based on the specific turbulent dissipation rate. Although the turbulent dissipation rate allows for a more physical grasp of its meaning it is known to have trouble in predicting complicated flow phenomena. One idea moving forward would to test a Reynolds stress transport model that utilizes the specific turbulent dissipation rate like the Menter SST turbulence model and the k-ω turbulence model do. In Conclusion, OpenFOAM provided an excellent platform on which to run and analyze turbulence models. However, in order to capture the physical phenomena of the separation bubble with extreme accuracy a different turbulence model other than the ones chosen here need to be implemented.

Figures



a) b)

Figure 1. Grid 113 x 33 (65 Points on Airfoil Surface) : a) complete computational domain, b) zoomed view of the grid near the airfoil surface.

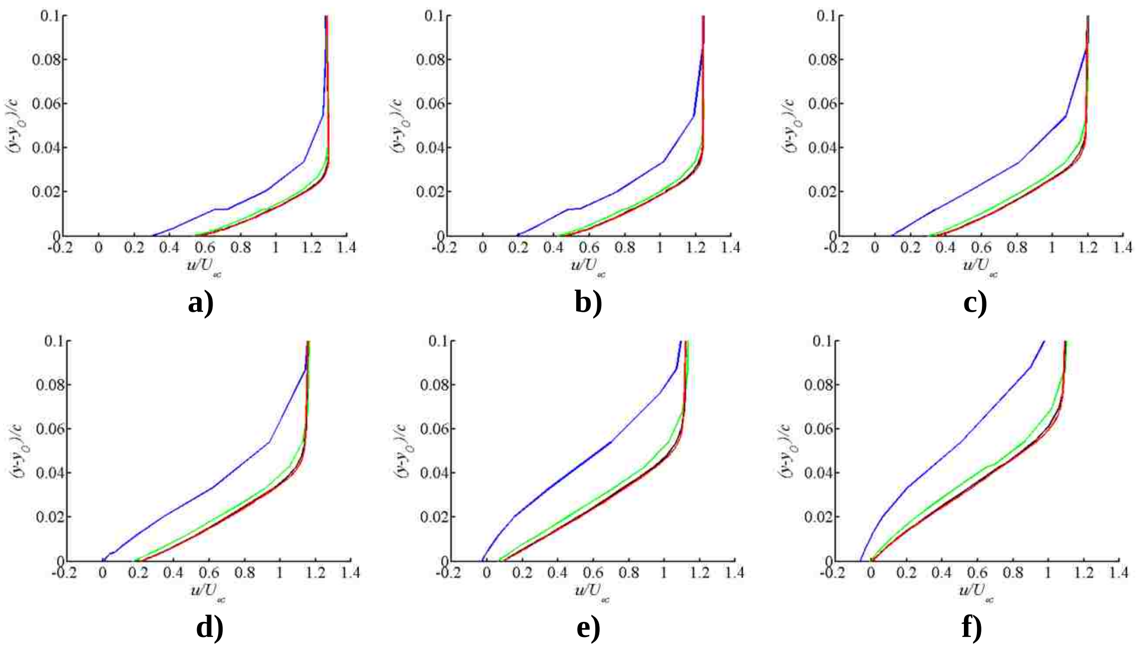


Figure 2. Sensitivity analysis for k-ω streamwise velocity a) x/c =0.6753, b) x/c=0.7308, c) x/c=0.7863, d) x/c=0.8418, e) x/c=0.8973 and, d) x/c=0.9528. Notations: —— grid 113×33, —— grid 225x65, —— grid 449×129, —— grid 897×257.

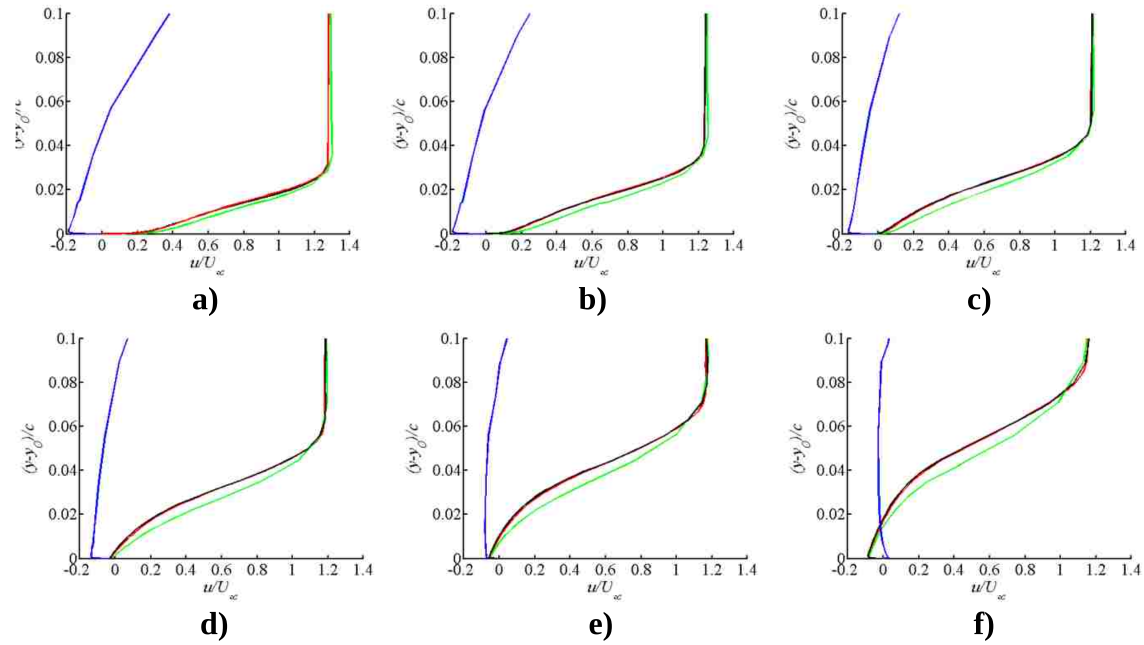


Figure 3. Sensitivity analysis for Menter SST streamwise velocity a) x/c =0.6753, b) x/c =0.7308, c) x/c=0.7863, d) x/c=0.8418, e) x/c=0.8973 and, d) x/c=0.9528. Notations: ——— grid 113×33, ——— grid 225x65, ——— grid 449×129, ——— grid 897×257.

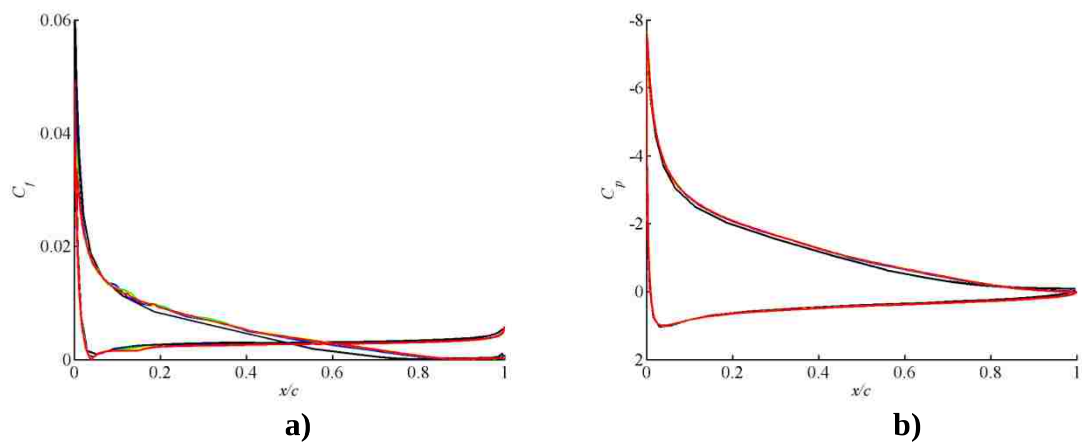
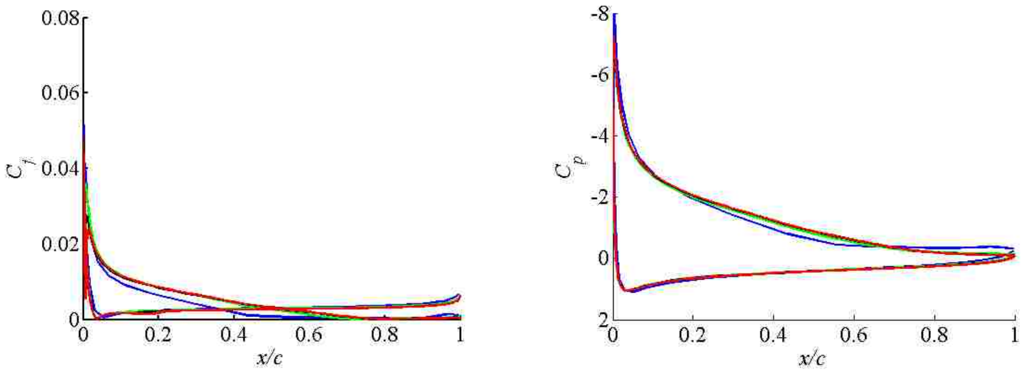


Figure 6. Sensitivity analysis for *k-ω* a) Skin Friction Coefficient, b) Coefficient of Pressure. Notations: ——— grid 113×33, ——— grid 225x65, ——— grid 449×129, ——— grid 897×257.



b) b)

Figure 7. Sensitivity analysis for Menter SST a) Skin Friction Coefficient, b) Coefficient of Pressure. Notations: —— grid 113×33, —— grid 225x65, —— grid 449×129, —— grid 897×257.

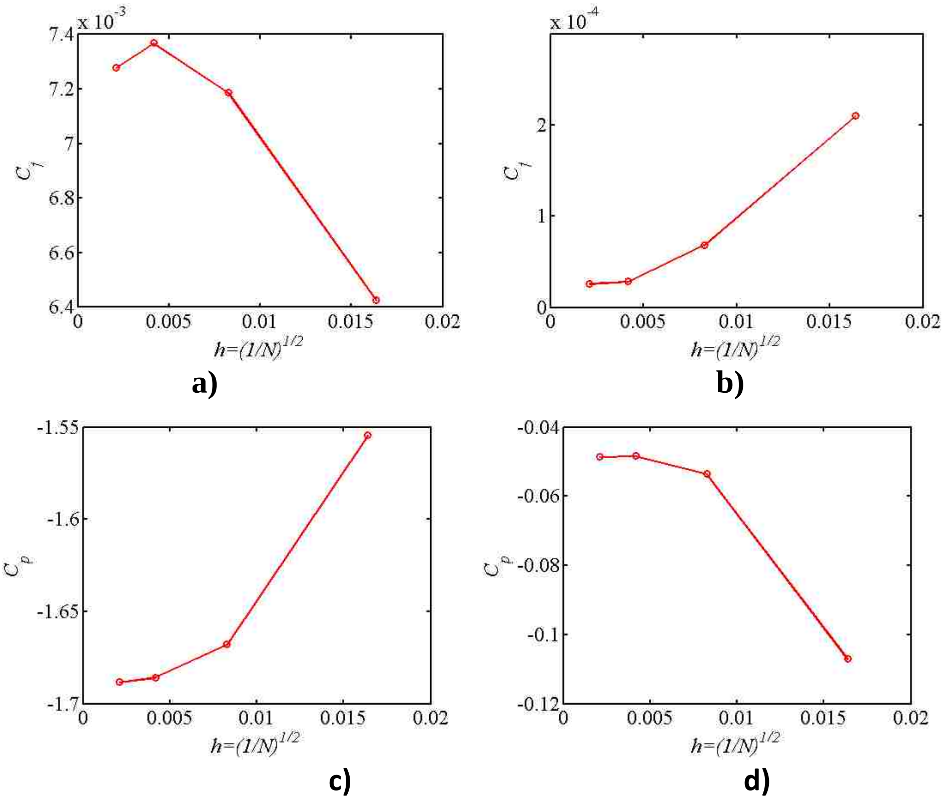


Figure 8. Richardson extrapolation analysis for *k-ω* a) C_f at x/c=0.3, b) C_f at x/c=0.9, c) C_p at x/c = 0.3 and, d) C_p at x/c =0.9.

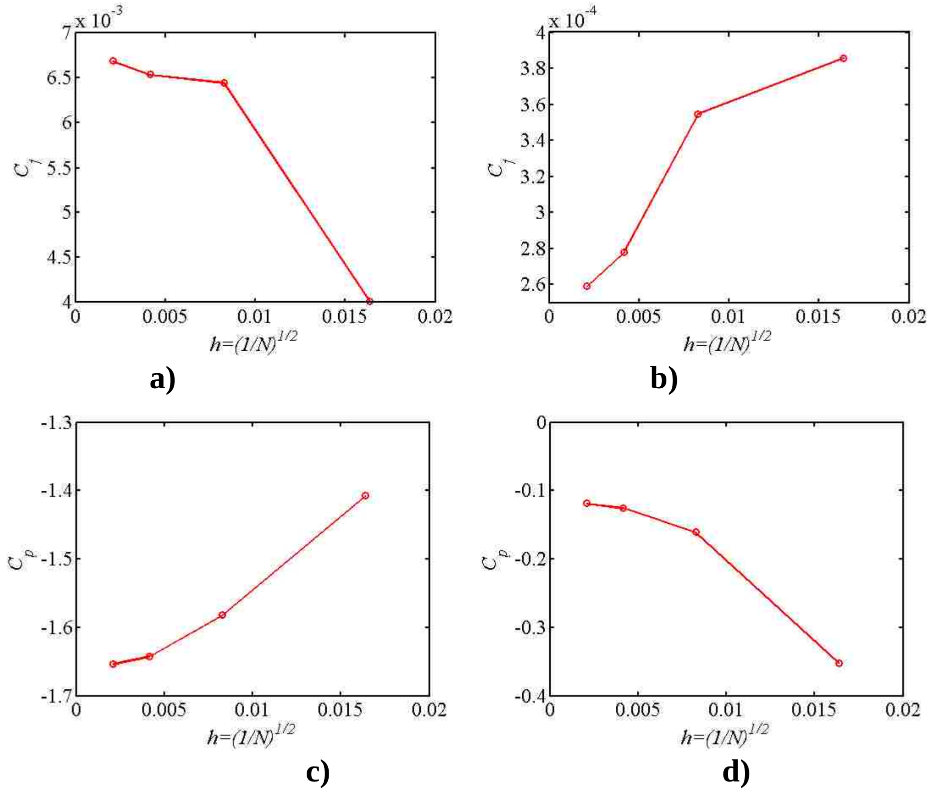
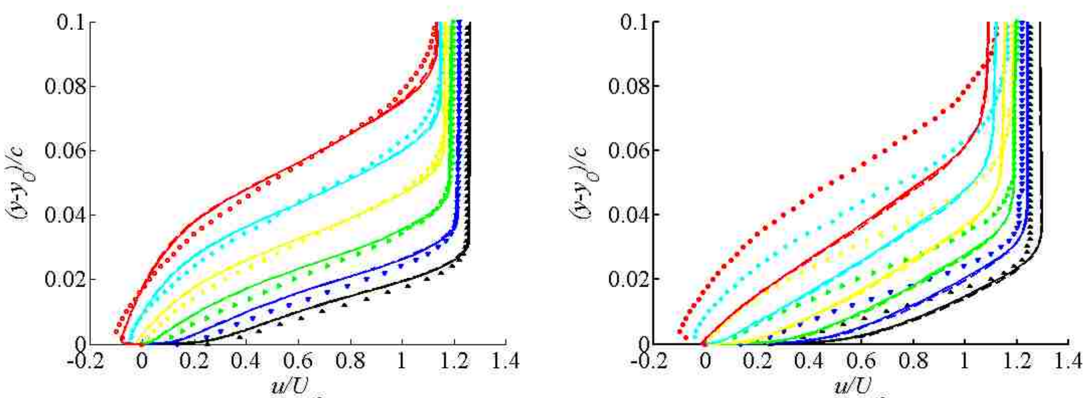
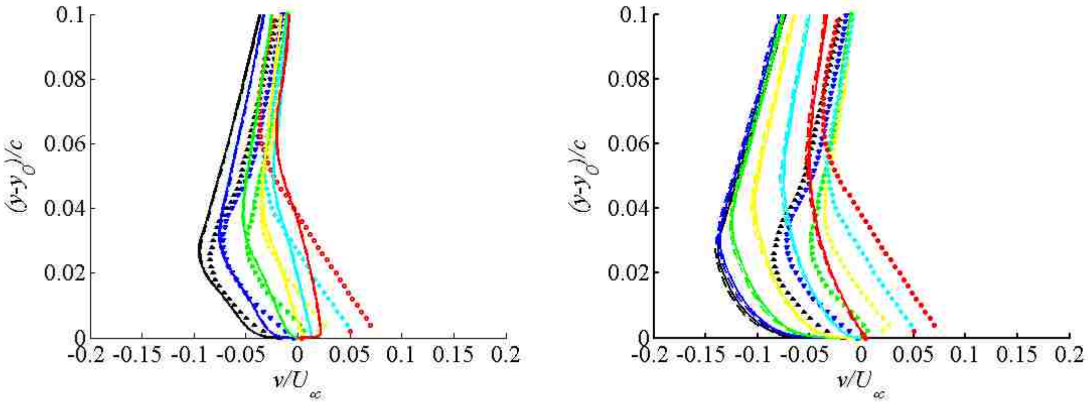


Figure 9. Richardson extrapolation analysis for Menter SST a) C_f at x/c=0.3, b) C_f at x/c=0.9, c) C_p at x/c = 0.3 and, d) C_p at x/c =0.9.



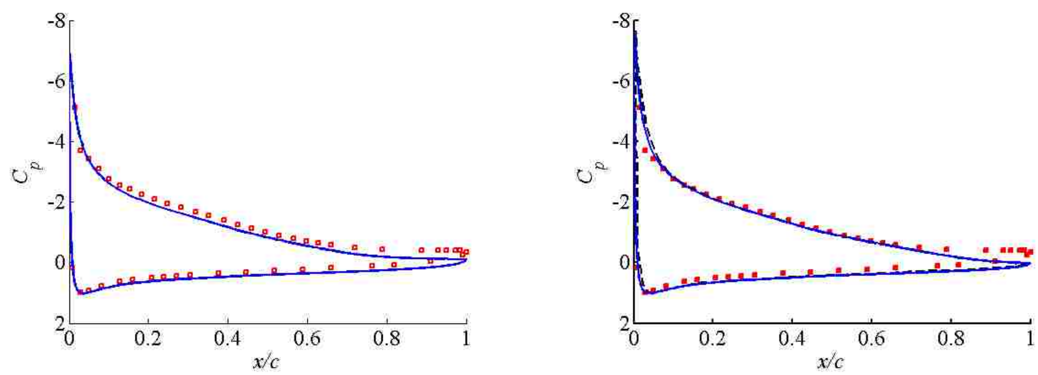
a) b)

Figure 10. Streamwise velocity profiles for flow over a 2D NACA 4412 profile on the fine grid 897×257 with a) Menter SST and b) *k-ω* turbulence models. Notations: experimental data ▲ x/c=0.6753, ▼ x/c=0.7308, ► x/c=0.7863, ◄ x/c=0.8418, ♦ x/c=0.8973, ● x/c=0.9528; computational profiles: - - - FUN3D and CFL3D, — OpenFOAM.



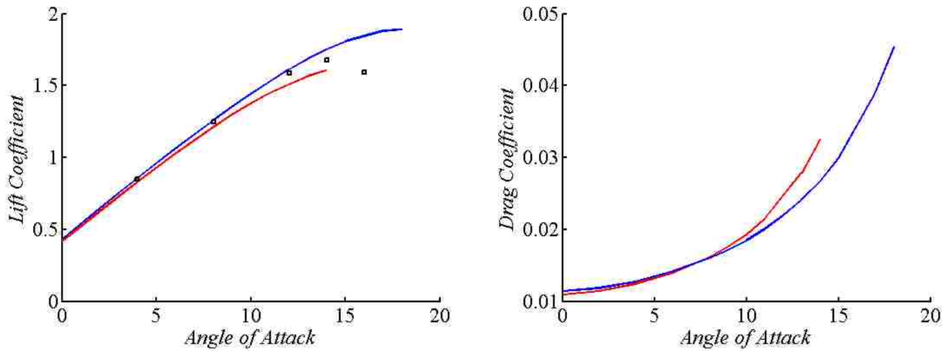
a) b)

Figure 11. Transverse velocity profiles for flow over a 2D NACA 4412 profile on the fine grid 897×257 with a) Menter SST and b) *k-ω* turbulence models. Notations: experimental data ▲ x/c=0.6753, ▼ x/c=0.7308, ▶ x/c=0.7863, ◀ x/c=0.8418, ♦ x/c=0.8973, ● x/c=0.9528; computational profiles: - - - FUN3D and CFL3D, — OpenFOAM.



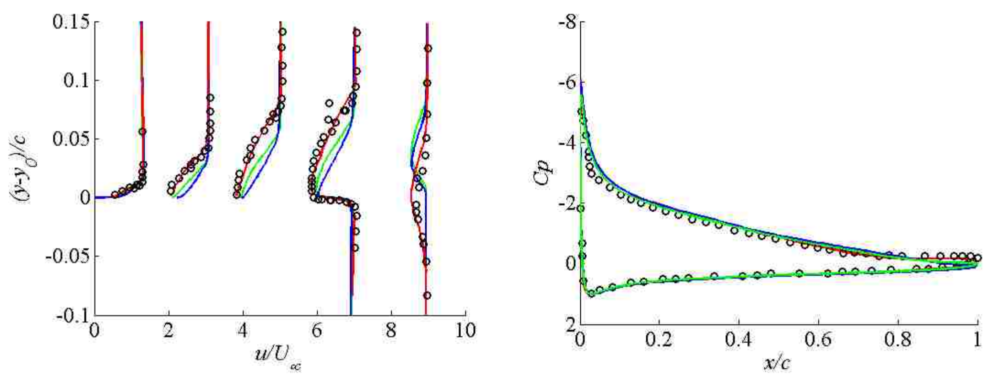
a) b)

Figure 12. Pressure coefficient distribution for flow over a 2D NACA 4412 profile on the fine grid 897×257 with a) Menter SST, b) *k-ω* turbulence models. Notations: ■ experimental data, - - - FUN3D and CFL3D, — OpenFOAM.



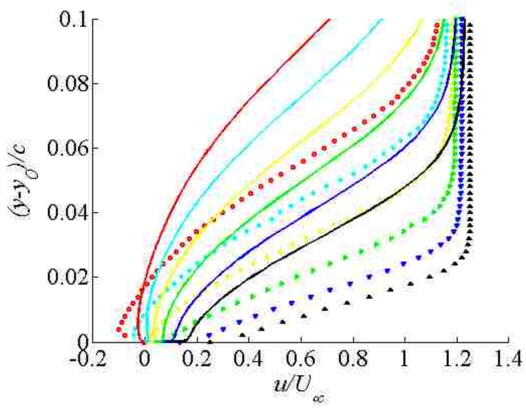
a) **b)**

Figure 13. a) Lift and b) Drag coefficients vs. angle of attack. Notations: □ experimental data at Re = 1.52×10^6, simulations; —— *k-ω*, —— Menter SST.



a) **b)**

Figure 14. Comparison data for the Second set of Flow parameters on the finest grid 897× 257 based on Wadcock [24]. a) From left to right, Streamwise velocity profiles for flow over a 2D NACA 4412 profile at x/c = 0.529, 0.815, 0.952; and velocity component normal to the tunnel axis in the wake at x_w/c = 0.007, 0.282. Profiles are shifted by 0, 2, 4, 6, 8 respectively along the abscissa. b) Pressure coefficient for flow over a 2D NACA 4412. Notations: ○ experimental data, —— LES, —— Menter SST, —— *k-ω*.



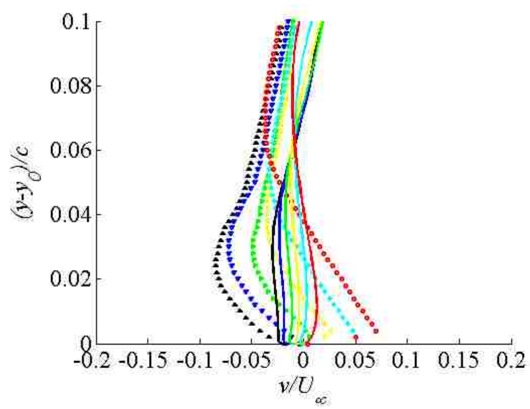


Figure 15. LRR-IP a) Streamwise and b) Transverse velocity profiles for flow over a 2D NACA 4412 profile on the fine grid 897×257. Notations: experimental data ▲ x/c=0.6753, ▼ x/c=0.7308, ► x/c=0.7863, ◄ x/c=0.8418, ♦ x/c=0.8973, ● x/c=0.9528; computational profiles: — OpenFOAM.

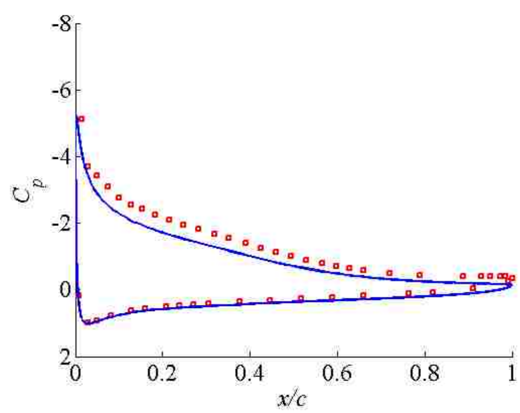


Figure 16. LRR-IP pressure coefficient profile for flow over a 2D NACA 4412 profile on the fine grid 897×257. Notations: ■ experimental data; computational profiles: — OpenFOAM.

V. References

[1] Freitas, C. J. "Perspective: selected benchmarks from commercial CFD codes." *Journal of Fluids Engineering* 117.2 (1995): 208-218.

[2] Iaccarino, G. "Predictions of a turbulent separated flow using commercial CFD codes." *Journal of Fluids Engineering* 123.4 (2001): 819-828.

[3] OpenFOAM, Open Field Operation and Manipulation, Software Package, www.openfoam.com/

[4] Rumsey, C. (2013, March 29). *CFL3D Home Page*. Retrieved from http://cfl3d.larc.nasa.gov/

[5] Carpenter, M. (2014, June 27). *FUN3D Home Page*. Retrieved from http://fun3d.larc.nasa.gov/

[6] Wilcox, D. C. (2006). *Turbulence modeling for CFD* 3rd edition, La Canada, CA: DCW Industries, pp. 40, 90, 109-112, 125-126, 322.

[7] Furbo, E. (2010). "Evaluation of RANS turbulence models for flow problems with significant impact of boundary layers". Master's thesis, Swedish Defense Research Agency, FOI. December 2010.

[8] CFD Online. (2013, January 3). *Boussinesq Eddy Viscosity Assumption*. Retrieved from http://www.cfd-online.com/Wiki/Boussinesq_eddy_viscosity_assumption

[9] Rumsey, C. L., Rubinstein, R., Salas, M. D., Thomas, J. L., "Turbulence Modeling Workshop", NASA/CR-2001-210841, ICASE Interim Report No. 37.

[10] CFD Online. (2011, June 13). *Specific Turbulence Dissipation Rate*. Retrieved from http://www.cfd-online.com/Wiki/Specific_turbulence_dissipation_rate, NASA/CR-2001-210841, ICASE Interim Report No. 37.

[11] Prandtl, L, "Übe rein neues Formelsystem fur die ausgebilldete Turbulenz," Nacr. Akad. Wiss. Göttingen, Math-Phys. Kl., (1945) pp. 6-19.

[12] Hanjalić, K., & Launder, B. (2011). *Modelling turbulence in engineering and the environment: second-moment routes to closure*. Cambridge University Press. Cambridge, UK. pp. 5, 25, 63, 75, 95, 195.

[13] Daly, B.J. and Harlow, F. H., "Transport equations in turbulence", *Phys. Fluids* 13, (1970) 2634-2649.

[14] Lumley, J. L., "Computational modeling of turbulent flows", *Adv. Appl. Mech*. 18, (1978) 123-176.

[15] Rotta, J. C., "Statistische Theorie nichthomogener Turbulenz", *Z. Phys*. 129, (1951) 547-572, 131, 51-77.

[16] Naot, D., Shavit, A. and Wolfshtein, M., "Interactions between components of the turbulent correlation tensor", *Isr. J. Technol*. 8, (1970) 259-269.

[17] Shir, C. C., "A preliminary numerical study of atmospheric turbulent flows in the idealized turbulent boundary layer". *J. Atmos. Sci*. 30, (1973) 1327-1339.

[18] Gibson, M. M. and Launder, B. E., "Ground effects on pressure fluctuations in the atmospheric boundary layer", *J. Fluid Mech. 86*, (1978) 491-511.

[19] Kurbatskii, A. F., & Poroseva, S. V., "Modeling turbulent diffusion in a rotating cylindrical pipe flow". *International Journal of Heat and Fluid Flow*, Vol. *20, No.* 3, pp. 341-348. 1999.

[20] Pinkerton, R. M., "Calculated and measured pressure distributions over the midspan section of the N.A.C.A. 4412 airfoil" *National Advisory Committee for Aeronautics,* Report No. 563, 1936.

[21] Pinkerton, R. M., "The variation with Reynolds number of pressure distribution over an airfoil section" *National Advisory Committee for Aeronautic*, Report No. 613, 1938.

[22] Wu, T.J., "Visualization of stall characteristics of airfoils using the smoke-wire technique" Master Thesis, The University of Texas at Arlington. December 1992.

[23] Coles, D. and Wadcock, A. J., "Flying-Hot-Wire Study of Flow Past an NACA 4412 Airfoil at Maximum Lift," *AIAA Journal*, Vol. 17, No. 4, April 1979, pp. 321-329

[24] Wadcock, A. J., "Structure of the Turbulent Separated Flow Around a Stalled Airfoil," *NASA-CR-152263*, February 1979.

[25] Hastings, R.C. and Williams, B.R., "Studies of the flow field near a NACA 4412 aerofoil at nearly maximum lift" No. RAE-TM-AERO-2026. Royal Aircraft establishment Farnborough, England, 1984.

[26] Wadcock, A. J., "Investigation of low-speed turbulent separated flow around airfoils," NASA Contractor Report 177450, 1987.

[27] Bradran, O.O., "A flying hot-wire study of separated flows" Ph.D thesis. University of Bradford, United Kingdom, 1993.

[28] Bradran, O.O., and Bruun, H.H., "Turbulent flow over a NACA 4412 airfoil at angle of attack 15 degree" *Proceedings of FEDSM'03,* 4th ASME_JSME Joint Fluids Engineering Conference, Honolulu, Hawaii, July 6-11, 2003.

[29] Chang, J.W., "Near-wake characteristics of an oscillating NACA 4412 airfoil" *Journal of aircraft*, 41, No. 5, pp. 1240-1244, (2004).

[30] Storms, B.L. and Jang, C.S., "Lift enhancement of an airfoil using a Gurney flap and vortex generators" *Journal of Aircraft 31*, No. 3, pp 542-547, 1994.

[31] Jang, C.S., James, R.C., and Russell, C.M., "Numerical investigation of an airfoil with a gurney flap" *Aircraft Design*, No. 2, 1998.

[32] Ekaterinaris, J.A., and Menter, F.R. "Computation of oscillating airfoil flows with one- and two-equation turbulence models" *AIAA journal*, Vol 32, No. 12, pp 2359-2365, 1994.

[33] Saxena, G., and Agrawal, M., "Aerodynamic analysis of NACA 4412 airfoil using CFD" *International Jounal of Emerging Trends in Engineering and Development*, Vol 4., No. 3, 2013.

[34] Saraf, A., Nazar, F.A., Singh, S.P., "Analysis of the k-ϵ and k-ω standard models for the simulation of the flow over a National Advisory Committee for Aeronautics (NACA) 4412 airfoil" *Iternational Journal of Scientific & Engineering Research*, Vol. 4, No. 6, 2013.

[35] Badran, O., "Formulation of two-equation turbulence models for turbulent flow over a NACA 4412 airfoil at angle of attack 15 degree" *6th International Colloquium on Bluff Bodies Aerodynamics and Applications*, Milano, Vol. 6., 2008.

[36] Rosen, A.M. "Turbulence modeling for subsonic separated flows over 2-D airfoils and 3-D wings" Master Thesis, Purdue University, 2013.

[37] Jansen, K., "Unstructured grid large eddy simulation of flow over an airfoil" *Annual Research Briefs 1994*. Center for Turbulence Research, NASA Ames/Stanford University., pp 161-174. 1994.

[38] Jansen, K., "Preliminary large-eddy simulations of flow around a NACA 4412 airfoil using unstructured grids" *Annual Research Briefs 1995.* Center for Turbulence Research, NASA Ames/Stanford University., pp 61-72. 1995.

[39] Jansen, K., "Large-eddy simulation of flow around a NACA 4412 airfoil using unstructured grids" *Annual Research Briefs 1996.* Center for Turbulence Research, NASA Ames/Stanford University., pp 225-232. 1996.

[40] Park, G.I. and Moin, P., "An improved dynamic non-equilibrium wall-model for large eddy simulation," *Physics of Fluids*, Vol. 26, January 2014. ID 015108.

[41] Park, G.I. and Moin, P., "A dynamic slip boundary condition for wall-modeled large eddy simulation," *Physics of Fluids*, Vol. 26, January 2014. ID 015104.

[42] Rumsey, C. (2014, April 7). *Turbulence Modeling Resource: Purpose*. Retrieved from http://turbmodels.larc.nasa.gov/

[43] Smith, B., Rumsey, C., Huang, G., (2014, June 27). *Turbulence Model Benchmarking Working Group*. Retrieved from http://turbmodels.larc.nasa.gov/tmbwg.html

[44] Williams, D. (No publication date available) *AIAA Fluid Dynamics Technical Committee.* Retrieved from https://info.aiaa.org/tac/ASG/FDTC/default.aspx

[45] American Institute of Aeronautics and Astronautics. *Home Page*. Retrieved from http://www.aiaa.org/

[46] Thigpen, W. (2015, March 19). *Pleiades Supercomputer Homepage.* Retrieved from http://www.nas.nasa.gov/hecc/resources/pleiades.html

[47] OpenFOAM Foundation. (No publication date available). *The OpenFOAM Foundation.* Retrieved from http://www.openfoam.org/

[48] OpenCFD Ltd. (No publication date available). *The open source CFD toolbox.* Retrieved from http://www.openfoam.com/about/

[49] Wilcox, D. C., "Turbulence Modeling for CFD, 3rd edition," DCW Industries, Inc., La Canada CA, 2006.

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

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

[52] Poroseva, S. V., and Murman, S. M., "FORANS Approach to Turbulence Modeling," AIAA 2014-2207, *Proc. AIAA Aviation and Aeronautics Forum and Exposition*, Atlanta, GA, June 16-20, 2014.

[53] Turbulence Model Benchmarking Working Group Turbulence Modeling Resource, NASA Langley, http://turbmodels.larc.nasa.gov/flatplate.html

[54] Gomez, S., Graves, B.J., Poroseva, S.V., "On the Accuracy of RANS Simulations of 2D Boundary Layers with OpenFOAM," AIAA 2014-2087, *Proc. AIAA Aviation and Aeronautics Forum and Exposition*, Atlanta, GA, June 16-20, 2014.

VI. Appendix

A. Pleiades Information

Pleiades is named after an astronomical open star cluster and it is one of the world's most powerful supercomputers. The system is a distributed-memory SGI ICE cluster connected with InfiniBand® in a dual-plane hypercube technology. The system contains the following types of Intel® Xeon® processors: E5-2680v2 (Ivy Bridge), E5-2670 (Sandy Bridge), and X5670 (Westmere). The cluster's information is as follows:

- System Architecture
  - Manufacturer: SGI
  - 163 racks (11,176 nodes)
  - 3.59 Pflop/s peak cluster
  - 1.54 Pflop/s LINPACK rating (November 2013)
  - 2 racks enhanced with NVIDIA graphics processing unit
  - Total cores: 184,800
  - Total memory: 502 TB
- Interconnects
  - Internode: InfiniBand®, with all nodes connected in partial hypercube topology
  - Two independent InfiniBand® fabrics
  - Infiniband® DDR, QDR and FDR
  - Gigabit Ethernet management network
- Storage
  - SGI® InfiniteStorege NEXIS 9000 home filesystem
  - 15 PB of RAID disk storage configured over several cluster-wide Listre filesystems
- Operating Environment
  - Operating system: SUSE® Linux®
  - Job scheduler: PBS®
  - Compilters: Intel and GNU C, C++ and Fortran
  - MPI SGI MPT, MVAPICH2, Intel MPI

The information presented above was obtained from NASA's Advanced Super Computer Division website [54]. More details on the specifics of each subcomponent are available at the same location.

B. Source Code for Turbulence Models

For all the models the .C file will come first then immediately following the header .H file.

k-ω Turbulence Model .C-file

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2012 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 "kOmega20062D.H"
#include "addToRunTimeSelectionTable.H"

#include "backwardsCompatibilityWallFunctions.H"

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

namespace Foam
{
namespace incompressible
{
namespace RASModels
{

// * * * * * * * * * * * * * * Static Data Members * * * * * * * * * * * * * //
```

```
defineTypeNameAndDebug(kOmega20062D, 0);
addToRunTimeSelectionTable(RASModel, kOmega20062D, dictionary);

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

kOmega20062D::kOmega20062D
(
    const volVectorField& U,
    const surfaceScalarField& phi,
    transportModel& transport,
    const word& turbulenceModelName,
    const word& modelName
)
:
    RASModel(modelName, U, phi, transport, turbulenceModelName),

    Cmu_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "betaStar",
            coeffDict_,
            0.09 //Beta=9/100 in 2006
        )
    ),
    /*beta_                                             ORIGINAL beta DEFINITION
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "beta",
            coeffDict_,
            0.0708 //(changed from 0.072)
        )
    ),*/
    alpha_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "alpha",
            coeffDict_,
            0.52 //alpha=13/25 in 2006
        )
    ),
    alphaK_
    (
```

```
    dimensioned<scalar>::lookupOrAddToDict
    (
        "alphaK",
        coeffDict_,
        0.6 //sigma*=3/5 in 2006
    )
),
alphaOmega_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "alphaOmega",
        coeffDict_,
        0.5 //sigma=1/2 in 2006
    )
),
Clim_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Clim",
        coeffDict_,
        0.875 //Clim=7/8
    )
),
k_
(
    IOobject
    (
        "k",
        runTime_.timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    autoCreateK("k", mesh_)
),
omega_
(
    IOobject
    (
        "omega",
        runTime_.timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
```

```
    ),
    autoCreateOmega("omega", mesh_)
  ),
  nut_
  (
    IOobject
    (
      "nut",
      runTime_.timeName(),
      mesh_,
      IOobject::NO_READ,
      IOobject::AUTO_WRITE
    ),
    autoCreateNut("nut", mesh_)
  ),
//*************************************************Change NO_WRITE to AUTO_WRITE to verify values

 fBeta_
  (
    IOobject
    (
      "fBeta",
      runTime_.timeName(),
      mesh_,
      IOobject::NO_READ,
      IOobject::NO_WRITE
    ),
    mesh_,
    dimless
  ),
 Chi_
  (
    IOobject
    (
      "Chi",
      runTime_.timeName(),
      mesh_,
      IOobject::NO_READ,
      IOobject::NO_WRITE
    ),
    mesh_, dimless
  ),
 absChi_
  (
    IOobject
```

```
        (
            "absChi",
            runTime_.timeName(),
            mesh_,
            IOobject::NO_READ,
            IOobject::NO_WRITE
        ),
        mesh_, dimless
    ),
   beta_
    (
        IOobject
        (
            "beta",
            runTime_.timeName(),
            mesh_,
            IOobject::NO_READ,
            IOobject::NO_WRITE
        ),
        mesh_,
        dimless
    ),
   alphad_
    (
        IOobject
        (
            "alphad",
            runTime_.timeName(),
            mesh_,
            IOobject::NO_READ,
            IOobject::NO_WRITE
        ),
        mesh_, dimensionedScalar("zero", dimless, 0.125)
    )
{
    bound(k_, kMin_);
    bound(omega_, omegaMin_);

   //nut_ = k_/omega_; //Standard OpenFOAM definition
    nut_ = k_/ max(omega_, Clim_*sqrt(2/0.09*magSqr(symm(fvc::grad(U_)))));;
    nut_.correctBoundaryConditions();


    printCoeffs();
}
```

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

tmp<volSymmTensorField> kOmega20062D::R() const
{
   return tmp<volSymmTensorField>
   (
      new volSymmTensorField
      (
         IOobject
         (
            "R",
            runTime_.timeName(),
            mesh_,
            IOobject::NO_READ,
            IOobject::NO_WRITE
         ),
         ((2.0/3.0)*I)*k_ - nut_*twoSymm(fvc::grad(U_)),
         k_.boundaryField().types()
      )
   );
}


tmp<volSymmTensorField> kOmega20062D::devReff() const
{
   return tmp<volSymmTensorField>
   (
      new volSymmTensorField
      (
         IOobject
         (
            "devRhoReff",
            runTime_.timeName(),
            mesh_,
            IOobject::NO_READ,
            IOobject::NO_WRITE
         ),
        -nuEff()*dev(twoSymm(fvc::grad(U_)))
      )
   );
}


tmp<fvVectorMatrix> kOmega20062D::divDevReff(volVectorField& U) const
{
```

```
    return
    (
      - fvm::laplacian(nuEff(), U)
      - fvc::div(nuEff()*dev(T(fvc::grad(U))))
    );
}


tmp<fvVectorMatrix> kOmega20062D::divDevRhoReff
(
    const volScalarField& rho,
    volVectorField& U
) const
{
    volScalarField muEff("muEff", rho*nuEff());

    return
    (
      - fvm::laplacian(muEff, U)
      - fvc::div(muEff*dev(T(fvc::grad(U))))
    );
}


bool kOmega20062D::read()
{
    if (RASModel::read())
    {
        Cmu_.readIfPresent(coeffDict());
        //beta_.readIfPresent(coeffDict());                    Must be commented for blending
function
        alphaK_.readIfPresent(coeffDict());
        alphaOmega_.readIfPresent(coeffDict());


        return true;
    }
    else
    {
        return false;
    }
}


void kOmega20062D::correct()
{
```

```
    RASModel::correct();

    if (!turbulence_)
    {
        return;
    }


    volTensorField GradU(fvc::grad(U_));
    volSymmTensorField Sij(symm(GradU));
    volTensorField Omij(-skew(GradU));
    volScalarField StressLim(Clim_*sqrt(2/Cmu_)*mag(Sij));
    volSymmTensorField tauij(2*nut_*Sij-((2/3)*I)*k_);
    volVectorField Gradk(fvc::grad(k_));
    volVectorField Gradomega(fvc::grad(omega_));

    volScalarField G(type() + ".G", tauij && GradU);
    //volScalarField G(GName(), tauij && GradU); //for newer OF versions


    // Update omega and G at the wall
    omega_.boundaryField().updateCoeffs();

//START NEW STUFF FOR 2006....................................

volScalarField alphadCheck_(Gradk & Gradomega);  //condition to change alphad_

forAll(alphad_,celli)
{
        if (alphadCheck_[celli] <= 0.0001)
        {
        alphad_[celli]=scalar(0);
        }else
        {
        alphad_[celli]=scalar(0.125);
        }
}


volScalarField CDkOmega(alphad_/omega_*(Gradk & Gradomega)); //last term in
NASA equations


//modify the values of Chi,fbeta, beta, for the corresponding cell, cellI:

//Alternate definition (yields the same results as what is used)
```

```
//volTensorField OmProd(Omij & Omij);
//volTensorField OmSProd(OmProd & Sij);
//Chi_ = tr(OmSProd)/pow((Cmu_*omega_),3);

Chi_ = (Omij & Omij) && Sij /pow((Cmu_*omega_),3);
absChi_ = mag(Chi_);
fBeta_ = 1.0; //This term should be (1.+85.*absChi_)/(1.+100.*absChi_); for 3D
beta_ = 0.0708*fBeta_;



    // Turbulence specific dissipation rate equation
    tmp<fvScalarMatrix> omegaEqn
    (
        fvm::ddt(omega_)
      + fvm::div(phi_, omega_)
      - fvm::laplacian(DomegaEff(), omega_)
     ==
        alpha_*G*omega_/k_
      - fvm::Sp(beta_*omega_, omega_)
      + CDkOmega //Crossflow diffusion term to match 2006
    );

    omegaEqn().relax();

    omegaEqn().boundaryManipulate(omega_.boundaryField());

    solve(omegaEqn);
    bound(omega_, omegaMin_);


    // Turbulent kinetic energy equation
    tmp<fvScalarMatrix> kEqn
    (
        fvm::ddt(k_)
      + fvm::div(phi_, k_)
      - fvm::laplacian(DkEff(), k_)
     ==
        G
      - fvm::Sp(Cmu_*omega_, k_)
    );

    kEqn().relax();
    solve(kEqn);
    bound(k_, kMin_);
```

```
    // Re-calculate viscosity
    //nut_ = k_/omega_; //Standard OpenFOAM definition
    nut_ = k_/ max(omega_, Clim_*sqrt(2/0.09*magSqr(symm(fvc::grad(U_)))));;
    nut_.correctBoundaryConditions();
}

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

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

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

<u>k-ω Turbulence Model .H-file</u>

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2012 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::incompressible::RASModels::kOmega20062DC2

Group
```

```
    grpIcoRASTurbulence

Description
    Standard high Reynolds-number k-omega turbulence model for
    incompressible flows.

    References:
        http://turbmodels.larc.nasa.gov/wilcox.html

        Turbulence Modeling for CFD (3rd Edition), David C. Wilcox, 2006

SourceFiles
    kOmega20062D.C

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

#ifndef kOmega20062D_H
#define kOmega20062D_H

#include "RASModel.H"

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

namespace Foam
{
namespace incompressible
{
namespace RASModels
{

/*---------------------------------------------------------------------------*\
                     Class kOmega20062DC2 Declaration
\*---------------------------------------------------------------------------*/

class kOmega20062D
:
    public RASModel
{

protected:

    // Protected data

        // Model coefficients

            dimensionedScalar Cmu_;
```

```
        //dimensionedScalar beta_;                     Commented for blending
function
        dimensionedScalar alpha_;
        dimensionedScalar alphaK_;
        dimensionedScalar alphaOmega_;
          dimensionedScalar Clim_;

    // Fields

        volScalarField k_;
        volScalarField omega_;
        volScalarField nut_;
          volScalarField fBeta_;
          volScalarField Chi_;
          volScalarField absChi_;
          volScalarField beta_;
        volScalarField alphad_;


public:

  //- Runtime type information
  TypeName("kOmega20062D");

  // Constructors

    //- Construct from components
    kOmega20062D
    (
      const volVectorField& U,
      const surfaceScalarField& phi,
      transportModel& transport,
      const word& turbulenceModelName = turbulenceModel::typeName,
      const word& modelName = typeName
    );


  //- Destructor
  virtual ~kOmega20062D()
  {}


  // Member Functions

    //- Return the turbulence viscosity
    virtual tmp<volScalarField> nut() const
```

```
{
   return nut_;
}

//- Return the effective diffusivity for k
tmp<volScalarField> DkEff() const
{
   return tmp<volScalarField>
   (
      //new volScalarField("DkEff", alphaK_*nut_ + nu())
            new volScalarField("DkEff", alphaK_*k_/omega_ + nu())
   );
}

//- Return the effective diffusivity for omega
tmp<volScalarField> DomegaEff() const
{
   return tmp<volScalarField>
   (
      //new volScalarField("DomegaEff", alphaOmega_*nut_ + nu())
            new volScalarField("DomegaEff", alphaOmega_*k_/omega_ + nu())
   );
}

//- Return the turbulence kinetic energy
virtual tmp<volScalarField> k() const
{
   return k_;
}

//- Return the turbulence specific dissipation rate
virtual tmp<volScalarField> omega() const
{
   return omega_;
}

//- Return the turbulence kinetic energy dissipation rate
virtual tmp<volScalarField> epsilon() const
{
   return tmp<volScalarField>
   (
      new volScalarField
      (
         IOobject
         (
            "epsilon",
```

```
                mesh_.time().timeName(),
                mesh_
            ),
            Cmu_*k_*omega_,
            omega_.boundaryField().types()
        )
    );
}

//- Return the Reynolds stress tensor
virtual tmp<volSymmTensorField> R() const;


//- Return the effective stress tensor including the laminar stress
virtual tmp<volSymmTensorField> devReff() const;

//- Return the source term for the momentum equation
virtual tmp<fvVectorMatrix> divDevReff(volVectorField& U) const;

//- Return the source term for the momentum equation
virtual tmp<fvVectorMatrix> divDevRhoReff
(
    const volScalarField& rho,
    volVectorField& U
) const;

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

//- Read RASProperties dictionary
virtual bool read();
};

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

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

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

#endif

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

Menter SST Turbulence Model .C-file

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2012 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 "BobbySST.H"
#include "addToRunTimeSelectionTable.H"

#include "backwardsCompatibilityWallFunctions.H"

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

namespace Foam
{
namespace incompressible
{
namespace RASModels
{

// * * * * * * * * * * * * * * Static Data Members * * * * * * * * * * * * * //

defineTypeNameAndDebug(BobbySST, 0);
addToRunTimeSelectionTable(RASModel, BobbySST, dictionary);
```

```
// * * * * * * * * * * * * Private Member Functions  * * * * * * * * * * * * //

tmp<volScalarField> BobbySST::F1(const volScalarField& CDkOmega) const
{
   tmp<volScalarField> CDkOmegaPlus = max //limiter, what is defined as CD_kOmega
(NASA)
   (
      CDkOmega,
      dimensionedScalar("1.0e-10", dimless/sqr(dimTime), 1.0e-10)
   );

   tmp<volScalarField> arg1 = min
   (
         max
         (
            (scalar(1)/betaStar_)*sqrt(k_)/(omega_*y_),
            scalar(500)*nu()/(sqr(y_)*omega_)
         ),
         (4*alphaOmega2_)*k_/(CDkOmegaPlus*sqr(y_))
   );

   return tanh(pow4(arg1));
}


tmp<volScalarField> BobbySST::F2() const
{
   tmp<volScalarField> arg2 =
      max
      (
         (scalar(2)/betaStar_)*sqrt(k_)/(omega_*y_),
         scalar(500)*nu()/(sqr(y_)*omega_)

   );

   return tanh(sqr(arg2));
}


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

BobbySST::BobbySST
(
   const volVectorField& U,
   const surfaceScalarField& phi,
   transportModel& transport,
```

```
    const word& turbulenceModelName,
    const word& modelName
)
:
    RASModel(modelName, U, phi, transport, turbulenceModelName),

    alphaK1_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "alphaK1",
            coeffDict_,
            0.85
        )
    ),
    alphaK2_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "alphaK2",
            coeffDict_,
            1.0
        )
    ),
    alphaOmega1_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "alphaOmega1",
            coeffDict_,
            0.5
        )
    ),
    alphaOmega2_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "alphaOmega2",
            coeffDict_,
            0.856
        )
    ),
    gamma1_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
```

```
        "gamma1",
        coeffDict_,
        0.55316666
    )
),
gamma2_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "gamma2",
        coeffDict_,
        0.44035466
    )
),
beta1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "beta1",
        coeffDict_,
        0.075
    )
),
beta2_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "beta2",
        coeffDict_,
        0.0828
    )
),
betaStar_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "betaStar",
        coeffDict_,
        0.09
    )
),
a1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "a1",
```

```
        coeffDict_,
        0.31
    )
),
b1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "b1",
        coeffDict_,
        1.0
    )
),
c1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "c1",
        coeffDict_,
        10.0
    )
),

y_(mesh_),

k_
(
    IOobject
    (
        "k",
        runTime_.timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    autoCreateK("k", mesh_)
),
omega_
(
    IOobject
    (
        "omega",
        runTime_.timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
```

```
        ),
        autoCreateOmega("omega", mesh_)
    ),
    nut_
    (
        IOobject
        (
            "nut",
            runTime_.timeName(),
            mesh_,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        autoCreateNut("nut", mesh_)
    )
{
    bound(k_, kMin_);
    bound(omega_, omegaMin_);

    nut_ =
    (
        a1_*k_
      / max
        (
            a1_*omega_,
            b1_*F2()*sqrt(2.0)*mag(skew(fvc::grad(U_)))
        )
    );
    nut_.correctBoundaryConditions();

    printCoeffs();
}


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

tmp<volSymmTensorField> BobbySST::R() const
{
    return tmp<volSymmTensorField>
    (
        new volSymmTensorField
        (
            IOobject
            (
                "R",
                runTime_.timeName(),
```

```
                mesh_,
                IOobject::NO_READ,
                IOobject::NO_WRITE
            ),
            ((2.0/3.0)*I)*k_ - nut_*twoSymm(fvc::grad(U_)),
            k_.boundaryField().types()
        )
    );
}


tmp<volSymmTensorField> BobbySST::devReff() const
{
    return tmp<volSymmTensorField>
    (
        new volSymmTensorField
        (
            IOobject
            (
                "devRhoReff",
                runTime_.timeName(),
                mesh_,
                IOobject::NO_READ,
                IOobject::NO_WRITE
            ),
           -nuEff()*dev(twoSymm(fvc::grad(U_)))
        )
    );
}


tmp<fvVectorMatrix> BobbySST::divDevReff(volVectorField& U) const
{
    return
    (
      - fvm::laplacian(nuEff(), U)
      - fvc::div(nuEff()*dev(T(fvc::grad(U))))
    );
}


tmp<fvVectorMatrix> BobbySST::divDevRhoReff
(
    const volScalarField& rho,
    volVectorField& U
) const
```

```
{
   volScalarField muEff("muEff", rho*nuEff());

   return
   (
     - fvm::laplacian(muEff, U)
     - fvc::div(muEff*dev(T(fvc::grad(U))))
   );
}


bool BobbySST::read()
{
   if (RASModel::read())
   {
      alphaK1_.readIfPresent(coeffDict());
      alphaK2_.readIfPresent(coeffDict());
      alphaOmega1_.readIfPresent(coeffDict());
      alphaOmega2_.readIfPresent(coeffDict());
      gamma1_.readIfPresent(coeffDict());
      gamma2_.readIfPresent(coeffDict());
      beta1_.readIfPresent(coeffDict());
      beta2_.readIfPresent(coeffDict());
      betaStar_.readIfPresent(coeffDict());
      a1_.readIfPresent(coeffDict());
      b1_.readIfPresent(coeffDict());
      c1_.readIfPresent(coeffDict());


      return true;
   }
   else
   {
      return false;
   }
}


void BobbySST::correct()
{
   RASModel::correct();

   if (!turbulence_)
   {
      return;
   }
```

```
    if (mesh_.changing())
    {
        y_.correct();
    }


    const volScalarField S2(2*magSqr(symm(fvc::grad(U_))));
    volTensorField GradU(fvc::grad(U_));
    volSymmTensorField Sij(symm(GradU));
    volSymmTensorField tauij(2*nut_*Sij-((2/3)*I)*k_);
    volScalarField G3(type() + ".G", tauij && GradU);
    volScalarField G(type() + ".G", nut_*S2);
    volScalarField G4(type() + ".G", min(tauij &&
GradU,scalar(20.0)*betaStar_*omega_*k_));
    //volScalarField G(GName(), nut_*2*magSqr(symm(fvc::grad(U_)))); for newer OF
versions

    volScalarField G2(type() + ".G", min(G,scalar(20.0)*betaStar_*omega_*k_));
//******************DIF



    // Update omega and G at the wall
    omega_.boundaryField().updateCoeffs();

    const volScalarField CDkOmega
    (
        (2*alphaOmega2_)*(fvc::grad(k_) & fvc::grad(omega_))/omega_
    );

    const volScalarField F1(this->F1(CDkOmega));





    // Turbulent frequency equation
 /*   tmp<fvScalarMatrix> omegaEqn
    (
        fvm::ddt(omega_)
      + fvm::div(phi_, omega_)
      - fvm::laplacian(DomegaEff(F1), omega_)
     ==
        gamma(F1)*S2
      - fvm::Sp(beta(F1)*omega_, omega_)
```

```
    - fvm::SuSp
      (
          (F1 - scalar(1))*CDkOmega/omega_,
          omega_
      )
  );*/

  // Turbulent frequency equation
  tmp<fvScalarMatrix> omegaEqn
  (
      fvm::ddt(omega_)
    + fvm::div(phi_, omega_)
    - fvm::laplacian(DomegaEff(F1), omega_)
   ==
      gamma(F1)*G/nut_
//*****************************************************DIFFERENT FROM
STD
    - fvm::Sp(beta(F1)*omega_, omega_)
    + fvm::Sp //changed this (see above)
      (
          (scalar(1)-F1)*CDkOmega/omega_,
          omega_
      )
  );

  omegaEqn().relax();

  omegaEqn().boundaryManipulate(omega_.boundaryField());

  solve(omegaEqn);
  bound(omega_, omegaMin_);

  // Turbulent kinetic energy equation
 /*  tmp<fvScalarMatrix> kEqn
  (
      fvm::ddt(k_)
    + fvm::div(phi_, k_)
    - fvm::laplacian(DkEff(F1), k_)
   ==
      G
    - fvm::Sp(betaStar_*omega_, k_)
  );*/

  // Turbulent kinetic energy equation ADDED LIMITER G2 (NASA)
  tmp<fvScalarMatrix> kEqn
  (
```

```
      fvm::ddt(k_)
    + fvm::div(phi_, k_)
    - fvm::laplacian(DkEff(F1), k_)
   ==
      G2
//*********************************************************************DI
FFERENT FROM STD
    - fvm::Sp(betaStar_*omega_, k_)
   );

   kEqn().relax();
   solve(kEqn);
   bound(k_, kMin_);


   // Re-calculate viscosity
   nut_ = a1_*k_/max(a1_*omega_, b1_*F2()*sqrt(2.0)*mag(skew(fvc::grad(U_))));
   nut_.correctBoundaryConditions();
}


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

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

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

Menter SST Turbulence Model .H-file

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2012 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::incompressible::RASModels::BobbySST

Description
    Implementation of the k-omega-SST turbulence model for incompressible
    flows.

    Turbulence model described in:
    \verbatim
        Menter, F., Esch, T.,
        "Elements of Industrial Heat Transfer Prediction",
        16th Brazilian Congress of Mechanical Engineering (COBEM),
        Nov. 2001.
    \endverbatim

    with the addition of the optional F3 term for rough walls from
    \verbatim
        Hellsten, A.
        "Some Improvements in Menter’s k-omega-SST turbulence model"
        29th AIAA Fluid Dynamics Conference,
        AIAA-98-2554,
        June 1998.
    \endverbatim

    Note that this implementation is written in terms of alpha diffusion
    coefficients rather than the more traditional sigma (alpha = 1/sigma) so
    that the blending can be applied to all coefficuients in a consistent
    manner.  The paper suggests that sigma is blended but this would not be
    consistent with the blending of the k-epsilon and k-omega models.

    Also note that the error in the last term of equation (2) relating to
    sigma has been corrected.

    Wall-functions are applied in this implementation by using equations (14)
    to specify the near-wall omega as appropriate.
```

```
    The blending functions (15) and (16) are not currently used because of the
    uncertainty in their origin, range of applicability and that is y+ becomes
    sufficiently small blending u_tau in this manner clearly becomes nonsense.

    The default model coefficients correspond to the following:
    \verbatim
        BobbySSTCoeffs
        {
            alphaK1     0.85034;
            alphaK2     1.0;
            alphaOmega1 0.5;
            alphaOmega2 0.85616;
            beta1       0.075;
            beta2       0.0828;
            betaStar    0.09;
            gamma1      0.5532;
            gamma2      0.4403;
            a1          0.31;
            b1          1.0;
            c1          10.0;
            F3          no;
        }
    \endverbatim

SourceFiles
    BobbySST.C

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

#ifndef BobbySST_H
#define BobbySST_H

#include "RASModel.H"
#include "wallDist.H"

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

namespace Foam
{
namespace incompressible
{
namespace RASModels
{

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

```
                    Class BobbySST Declaration
\*---------------------------------------------------------------------------*/

class BobbySST
:
    public RASModel
{

protected:

    // Protected data

        // Model coefficients
            dimensionedScalar alphaK1_;
            dimensionedScalar alphaK2_;

            dimensionedScalar alphaOmega1_;
            dimensionedScalar alphaOmega2_;

            dimensionedScalar gamma1_;
            dimensionedScalar gamma2_;

            dimensionedScalar beta1_;
            dimensionedScalar beta2_;

            dimensionedScalar betaStar_;

            dimensionedScalar a1_;
            dimensionedScalar b1_;
            dimensionedScalar c1_;

            Switch F3_;


        //- Wall distance field
        //  Note: different to wall distance in parent RASModel
        wallDist y_;

        // Fields

            volScalarField k_;
            volScalarField omega_;
            volScalarField nut_;


    // Protected Member Functions
```

```
        tmp<volScalarField> F1(const volScalarField& CDkOmega) const;
        tmp<volScalarField> F2() const;
        tmp<volScalarField> F3() const;
        tmp<volScalarField> F23() const;


        tmp<volScalarField> blend
        (
            const volScalarField& F1,
            const dimensionedScalar& psi1,
            const dimensionedScalar& psi2
        ) const
        {
            return F1*(psi1 - psi2) + psi2;
        }

        tmp<volScalarField> alphaK(const volScalarField& F1) const
        {
            return blend(F1, alphaK1_, alphaK2_);
        }

        tmp<volScalarField> alphaOmega(const volScalarField& F1) const
        {
            return blend(F1, alphaOmega1_, alphaOmega2_);
        }

        tmp<volScalarField> beta(const volScalarField& F1) const
        {
            return blend(F1, beta1_, beta2_);
        }

        tmp<volScalarField> gamma(const volScalarField& F1) const
        {
            return blend(F1, gamma1_, gamma2_);
        }


public:

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


    // Constructors
```

```
    //- Construct from components
    BobbySST
    (
        const volVectorField& U,
        const surfaceScalarField& phi,
        transportModel& transport,
        const word& turbulenceModelName = turbulenceModel::typeName,
        const word& modelName = typeName
    );


//- Destructor
virtual ~BobbySST()
{}


// Member Functions

    //- Return the turbulence viscosity
    virtual tmp<volScalarField> nut() const
    {
        return nut_;
    }

    //- Return the effective diffusivity for k
    tmp<volScalarField> DkEff(const volScalarField& F1) const
    {
        return tmp<volScalarField>
        (

            new volScalarField("DkEff", alphaK(F1)*nut_ + nu())
        );
    }

    //- Return the effective diffusivity for omega
    tmp<volScalarField> DomegaEff(const volScalarField& F1) const
    {
        return tmp<volScalarField>
        (

            new volScalarField("DomegaEff", alphaOmega(F1)*nut_ + nu())
        );
    }

    //- Return the turbulence kinetic energy
    virtual tmp<volScalarField> k() const
```

```
{
   return k_;
}

//- Return the turbulence specific dissipation rate
virtual tmp<volScalarField> omega() const
{
   return omega_;
}

//- Return the turbulence kinetic energy dissipation rate
virtual tmp<volScalarField> epsilon() const
{
   return tmp<volScalarField>
   (
      new volScalarField
      (
         IOobject
         (
            "epsilon",
            mesh_.time().timeName(),
            mesh_
         ),
         betaStar_*k_*omega_,
         omega_.boundaryField().types()
      )
   );
}

//- Return the Reynolds stress tensor
virtual tmp<volSymmTensorField> R() const;

//- Return the effective stress tensor including the laminar stress
virtual tmp<volSymmTensorField> devReff() const;

//- Return the source term for the momentum equation
virtual tmp<fvVectorMatrix> divDevReff(volVectorField& U) const;

//- Return the source term for the momentum equation
virtual tmp<fvVectorMatrix> divDevRhoReff
(
   const volScalarField& rho,
   volVectorField& U
) const;

//- Solve the turbulence equations and correct the turbulence viscosity
```

```
    virtual void correct();

    //- Read RASProperties dictionary
    virtual bool read();
};


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

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

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

#endif

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

Launder-Reece-Rodi Isotropization-of-Production Model .C-file

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2013 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 "myLRRIP.H"
#include "addToRunTimeSelectionTable.H"
#include "wallFvPatch.H"

#include "wallPointData.H"
#include "wallDistData.H"

#include "backwardsCompatibilityWallFunctions.H"


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

namespace Foam
{
namespace incompressible
{
namespace RASModels
{
```

```
// * * * * * * * * * * * * * * Static Data Members * * * * * * * * * * * * * //

defineTypeNameAndDebug(myLRRIP, 0);
addToRunTimeSelectionTable(RASModel, myLRRIP, dictionary);

// * * * * * * * * * * * * Private Member Functions  * * * * * * * * * * * * //
void myLRRIP::calcUTau(volScalarField& uTau)
{
   const volSymmTensorField Reff(this->devReff());

   forAll(uTau.boundaryField(), patchI)
   {
      uTau.boundaryField()[patchI] =
      sqrt(mag((-mesh_.Sf().boundaryField()[patchI]
              /mesh_.magSf().boundaryField()[patchI]
             ) & Reff.boundaryField()[patchI]));
   }

   wallDistData<wallPointData<scalar> > ytau_(mesh_, uTau, true);
}

tmp<volScalarField> myLRRIP::F1(volScalarField& uTau) const
{
   return exp(-0.5*yr_*uTau/nu());
}

tmp<volScalarField> myLRRIP::F2() const
{
   return 1.0 -(2.0/9.0)*exp(-sqr(sqr(k_)/(6.0*nu()*epsilon_)));
}

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

myLRRIP::myLRRIP
(
   const volVectorField& U,
   const surfaceScalarField& phi,
   transportModel& transport,
   const word& turbulenceModelName,
   const word& modelName
)
:
   RASModel(modelName, U, phi, transport, turbulenceModelName),

   Cmu_
   (
```

```
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Cmu",
        coeffDict_,
        0.09
    )
),
Cf1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Cf1",
        coeffDict_,
        0.2
           )
),
C1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "C1",
        coeffDict_,
        1.8
           )
),
C2_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "C2",
        coeffDict_,
        0.6
    )
),
Cs_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Cs",
        coeffDict_,
        0.2
    )
),
C1w_
(
    dimensioned<scalar>::lookupOrAddToDict
```

```
    (
        "C1w",
        coeffDict_,
        0.3
    )
),
C2w_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "C2w",
        coeffDict_,
        0.3
    )
),
Ceps_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Ceps",
        coeffDict_,
        0.15
    )
),
Ce1_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Ce1",
        coeffDict_,
       1.44
    )
),
Ce2_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        "Ce2",
        coeffDict_,
        1.92
    )
),
couplingFactor_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
```

```
        "couplingFactor",
        coeffDict_,
        0.0
    )
),

yr_(mesh_),

R_
(
    IOobject
    (
        "R",
        runTime_.timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    autoCreateR("R", mesh_)
),
k_
(
    IOobject
    (
        "k",
        runTime_.timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    autoCreateK("k", mesh_)
),
epsilon_
(
    IOobject
    (
        "epsilon",
        runTime_.timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    autoCreateEpsilon("epsilon", mesh_)
),
nut_
(
```

```
        IOobject
        (
            "nut",
            runTime_.timeName(),
            mesh_,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        autoCreateNut("nut", mesh_)
    )
{
    bound(k_, kMin_);
    bound(epsilon_, epsilonMin_);

    nut_ = Cmu_*sqr(k_)/epsilon_;
    nut_.correctBoundaryConditions();

    if (couplingFactor_.value() < 0.0 || couplingFactor_.value() > 1.0)
    {
        FatalErrorIn
        (
            "myLRRIP::myLRRIP"
            "(const volVectorField& U, const surfaceScalarField& phi,"
            "transportModel& transport)"
        )   << "couplingFactor = " << couplingFactor_
            << " is not in range 0 - 1" << nl
            << exit(FatalError);
    }

    printCoeffs();
}


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

tmp<volSymmTensorField> myLRRIP::devReff() const
{
    return tmp<volSymmTensorField>
    (
        new volSymmTensorField
        (
            IOobject
            (
                "devRhoReff",
                runTime_.timeName(),
                mesh_,
```

```
                IOobject::NO_READ,
                IOobject::NO_WRITE
            ),
            R_ - nu()*dev(twoSymm(fvc::grad(U_)))
        )
    );
}


tmp<fvVectorMatrix> myLRRIP::divDevReff(volVectorField& U) const
{
    if (couplingFactor_.value() > 0.0)
    {
        return
        (
            fvc::div(R_ + couplingFactor_*nut_*fvc::grad(U), "div(R)")
          + fvc::laplacian
            (
                (1.0 - couplingFactor_)*nut_,
                U,
                "laplacian(nuEff,U)"
            )
          - fvm::laplacian(nuEff(), U)
        );
    }
    else
    {
        return
        (
            fvc::div(R_)
          + fvc::laplacian(nut_, U, "laplacian(nuEff,U)")
          - fvm::laplacian(nuEff(), U)
        );
    }
}


tmp<fvVectorMatrix> myLRRIP::divDevRhoReff
(
    const volScalarField& rho,
    volVectorField& U
) const
{
    volScalarField muEff("muEff", rho*nuEff());

    if (couplingFactor_.value() > 0.0)
```

```
    {
        return
        (
            fvc::div
            (
                rho*R_ + couplingFactor_*(rho*nut_)*fvc::grad(U),
                "div((rho*R))"
            )
          + fvc::laplacian
            (
                (1.0 - couplingFactor_)*rho*nut_,
                U,
                "laplacian(muEff,U)"
            )
          - fvm::laplacian(muEff, U)
        );
    }
    else
    {
        return
        (
            fvc::div(rho*R_)
          + fvc::laplacian(rho*nut_, U, "laplacian(muEff,U)")
          - fvm::laplacian(muEff, U)
        );
    }
}


bool myLRRIP::read()
{
    if (RASModel::read())
    {

        Cmu_.readIfPresent(coeffDict());
        Cf1_.readIfPresent(coeffDict());
        C1_.readIfPresent(coeffDict());
        C2_.readIfPresent(coeffDict());
        Cs_.readIfPresent(coeffDict());
        C1w_.readIfPresent(coeffDict());
        C2w_.readIfPresent(coeffDict());
        Ceps_.readIfPresent(coeffDict());
        Ce1_.readIfPresent(coeffDict());
        Ce2_.readIfPresent(coeffDict());

        couplingFactor_.readIfPresent(coeffDict());
```

```
        if (couplingFactor_.value() < 0.0 || couplingFactor_.value() > 1.0)
        {
            FatalErrorIn("myLRRIP::read()")
                << "couplingFactor = " << couplingFactor_
                << " is not in range 0 - 1"
                << exit(FatalError);
        }

        return true;
    }
    else
    {
        return false;
    }
}


void myLRRIP::correct()
{
    RASModel::correct();

    if (!turbulence_)
    {
        return;
    }

    if (mesh_.changing())
    {
        yr_.correct();
    }

    volSymmTensorField P(-twoSymm(R_ & fvc::grad(U_)));
    volScalarField G(GName(), 0.5*mag(tr(P)));
    volScalarField magGradU(mag(fvc::grad(U_)));


    // Update epsilon and G at the wall
    epsilon_.boundaryField().updateCoeffs();

    volScalarField uTau
    (
        IOobject
        (
            "uTau",
            runTime_.timeName(),
```

```
        mesh_,
        IOobject::NO_READ,
        IOobject::NO_WRITE
    ),
    mesh_,
    dimensionedScalar
    (
        "uTau",
        dimLength/dimTime,
        0.0
    )
);

volScalarField Gwallfunc
(
    IOobject
    (
        "Gwallfunc",
        runTime_.timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::NO_WRITE
    ),
    (nut_+nu())
            *magGradU
            *pow(0.09, .25)*sqrt(k_)
            /(0.41*yr_)
);

            if( runTime_.outputTime() )
{
     Gwallfunc.write();
}

calcUTau(uTau);

const volScalarField F1(this->F1(uTau));

volScalarField Ce2s(Ce2_*F2());

forAll(Ce2s,cellI)
{
    Ce2s[cellI] = max( Ce2s[cellI], 1.4 );
}

// Dissipation equation
```

```
  tmp<fvScalarMatrix> epsEqn
  (
     fvm::ddt(epsilon_)
   + fvm::div(phi_, epsilon_)
  //- fvm::laplacian(Ceps*(k_/epsilon_)*R_, epsilon_)
   - fvm::laplacian(DepsilonEff(), epsilon_)
   ==
     Ce1_*G*epsilon_/k_
   - fvm::Sp(Ce2s*epsilon_/k_, epsilon_)
   //- fvm::Sp(Ce2_*epsilon_/k_, epsilon_)
   - fvm::Sp(2.0*nu()*F1/sqr(yr_), epsilon_ )
  );

  epsEqn().relax();

  epsEqn().boundaryManipulate(epsilon_.boundaryField());

  solve(epsEqn);

  bound(epsilon_, //epsilonMin_);

           dimensionedScalar
           (
                   "myEpsilonMin_",
                   kMin_.dimensions()/dimTime,
                   1e-15
           )
  );

volScalarField oldG
  (
     IOobject
     (
        "oldG",
        runTime_.timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::NO_WRITE
     ),
     G
  );

           if( runTime_.outputTime() )
  {
      oldG.write();
  }
```

```
if( runTime_.outputTime() )
{
    F1.write();
}

volScalarField F1term
(
   IOobject
   (
      "F1term",
      runTime_.timeName(),
      mesh_,
      IOobject::NO_READ,
      IOobject::NO_WRITE
   ),
   2.0*nu()*F1/sqr(yr_)*epsilon_
);

if( runTime_.outputTime() )
{
    F1term.write();
}

volScalarField F2
(
   IOobject
   (
      "F2",
      runTime_.timeName(),
      mesh_,
      IOobject::NO_READ,
      IOobject::NO_WRITE
   ),
   1.0 -(2.0/9.0)*exp(-sqr(sqr(k_)/(6.0*nu()*epsilon_)))
);

if( runTime_.outputTime() )
{
    F2.write();
}

volScalarField ce1
(
   IOobject
   (
```

```
        "ce1",
        runTime_.timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::NO_WRITE
    ),
    Ce1_*G*epsilon_/k_
);

if( runTime_.outputTime() )
{
     ce1.write();
}


volScalarField Ce2
(
    IOobject
    (
        "Ce2",
        runTime_.timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::NO_WRITE
    ),
    Ce2s
);

if( runTime_.outputTime() )
{
     Ce2.write();
}



// Reynolds stress equation

const fvPatchList& patches = mesh_.boundary();

forAll(patches, patchi)
{
    const fvPatch& curPatch = patches[patchi];

    if (isA<wallFvPatch>(curPatch))
    {
        forAll(curPatch, facei)
```

```
        {
            label faceCelli = curPatch.faceCells()[facei];
            P[faceCelli] *=
                min(G[faceCelli]/(0.5*mag(tr(P[faceCelli])) + SMALL), 1.0);
        }
    }
}

const volSymmTensorField PS2(-C2_*dev(P));
const volSymmTensorField reflect
(
    C1w_*epsilon_/k_*R_ + C2w_*PS2
);

tmp<fvSymmTensorMatrix> REqn
(
    fvm::ddt(R_)
  + fvm::div(phi_, R_)
//- fvm::laplacian(Cs*(k_/epsilon_)*R_, R_)
  - fvm::laplacian(DREff(), R_)
  + fvm::Sp(C1_*(epsilon_/k_), R_)
  ==
    P
  + (2.0/3.0)*(C1_-1.0)*I*epsilon_
  //- fvm::Sp(2.0*nu()/sqr(yr_), R_)  // Implicit
  + PS2
    // wall reflection terms
  + symm
    (
        I*((yr_.n() & reflect) & yr_.n())
      - 1.5*(yr_.n()*(reflect & yr_.n())
      + (yr_.n() & reflect)*yr_.n())
         )*Cf1_*pow(k_, 1.5)/(yr_*epsilon_)
);

volSymmTensorField epsOkTR
(
    IOobject
    (
        "epsOkTR",
        runTime_.timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::NO_WRITE
    ),
     epsilon_/k_*R_
```

```
);

if( runTime_.outputTime() )
{
    epsOkTR.write();
}

volSymmTensorField epsilonTensor
(
   IOobject
   (
      "epsilonTensor",
      runTime_.timeName(),
      mesh_,
      IOobject::NO_READ,
      IOobject::NO_WRITE
   ),
   (2.0/3.0)*(C1_-1.0)*I*epsilon_
  - 2.0*nu()*R_/sqr(yr_)
);

if( runTime_.outputTime() )
{
    epsilonTensor.write();
}

volScalarField fxn
(
   IOobject
   (
      "fxn",
      runTime_.timeName(),
      mesh_,
      IOobject::NO_READ,
      IOobject::NO_WRITE
   ),
   pow(k_, 1.5)/(yr_*epsilon_)
);

if( runTime_.outputTime() )
{
    fxn.write();
}
```

```
REqn().relax();
solve(REqn);

R_.max
(
   dimensionedSymmTensor
   (
      "zero",
      R_.dimensions(),
      symmTensor
      (
         kMin_.value(), -GREAT, -GREAT,
         kMin_.value(), -GREAT,
         kMin_.value()
      )
   )
);

k_ == 0.5*tr(R_);
bound(k_, kMin_);


// Re-calculate turbulent viscosity
nut_ = Cmu_*sqr(k_)/epsilon_;
nut_.correctBoundaryConditions();


// Correct wall shear stresses
forAll(patches, patchi)
{
   const fvPatch& curPatch = patches[patchi];

   if (isA<wallFvPatch>(curPatch))
   {
      symmTensorField& Rw = R_.boundaryField()[patchi];

      const scalarField& nutw = nut_.boundaryField()[patchi];

      const vectorField snGradU(U_.boundaryField()[patchi].snGrad());

      const vectorField& faceAreas
         = mesh_.Sf().boundaryField()[patchi];

      const scalarField& magFaceAreas
         = mesh_.magSf().boundaryField()[patchi];
```

```
          forAll(curPatch, facei)
          {
             // Calculate near-wall velocity gradient
             tensor gradUw
                 = (faceAreas[facei]/magFaceAreas[facei])*snGradU[facei];

             // Calculate near-wall shear-stress tensor
             tensor tauw = -nutw[facei]*2*symm(gradUw);

             // Reset the shear components of the stress tensor
             Rw[facei].xy() = tauw.xy();
             Rw[facei].xz() = tauw.xz();
             Rw[facei].yz() = tauw.yz();
          }
       }
    }
}


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

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

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

Launder-Reece-Rodi Isotropization-of-Production Model .H-file

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2012 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.
```

Class
Foam::incompressible::RASModels::myLRRIP

Group
grpIcoRASTurbulence

Description
Launder-Gibson Reynolds stress turbulence model for incompressible flows
with wall damping functions (So and Yoo, 1989). Coefficients were taken
from Kurbatskii et al. (1999).

References:

- M. Gibson, B. Launder. "Ground effects on pressure fluctuations in the atmospheric boundary layer". Journal of Fluid Mechanics, Vol 86, 1978.
- R. So, G. Yoo. "On the modeling of low-Reynolds number turbulence". NASA Contractor Report 3996, 1986.
- A. Kurbatskii, S. Poroseva. "Modeling turbulent diffusion in a rotating cylindrical pipe flow". International Journal of Heat and Fluid Flow, 20, 1999.

The default model coefficients correspond to the following:

```
\verbatim
    myLRRIPCoeffs
    {
        Cmu             0.09;
        Cf1             0.2;
        C1              1.8;
        C2              0.6;
        C1w             0.3;
        C2w             0.3;
        Cs              0.2;
        Ceps            0.15;
        Ce1             1.44;
        Ce2             1.92;
        couplingFactor  0;
    }
```

```
    \endverbatim

SourceFiles
    myLRRIP.C

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

#ifndef myLRRIP_H
#define myLRRIP_H

#include "RASModel.H"
#include "wallDistReflection.H"

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

namespace Foam
{
namespace incompressible
{
namespace RASModels
{

/*---------------------------------------------------------------------------*\
                           Class myLRRIP Declaration
\*---------------------------------------------------------------------------*/

class myLRRIP
:
    public RASModel
{

protected:

    // Protected data

        // Model coefficients

            dimensionedScalar Cmu_;

            dimensionedScalar Cf1_;
            dimensionedScalar C1_;
            dimensionedScalar C2_;
            dimensionedScalar Cs_;
            dimensionedScalar C1w_;
            dimensionedScalar C2w_;
            dimensionedScalar Ceps_;
```

```
            dimensionedScalar Ce1_;
            dimensionedScalar Ce2_;

            dimensionedScalar couplingFactor_;


        // Fields

            wallDistReflection yr_;

            volSymmTensorField R_;
            volScalarField k_;
            volScalarField epsilon_;
            volScalarField nut_;

    // Protected Member Functions

              void calcUTau(volScalarField& uTau);
        tmp<volScalarField> F1(volScalarField& uTau) const;
        tmp<volScalarField> F2() const;

public:

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

    // Constructors

        //- Construct from components
        myLRRIP
        (
            const volVectorField& U,
            const surfaceScalarField& phi,
            transportModel& transport,
            const word& turbulenceModelName = turbulenceModel::typeName,
            const word& modelName = typeName
        );


    //- Destructor
    virtual ~myLRRIP()
    {}


    // Member Functions
```

```
//- Return the turbulence viscosity
virtual tmp<volScalarField> nut() const
{
   return nut_;
}

//- Return the effective diffusivity for R
tmp<volSymmTensorField> DREff() const
{
   return tmp<volSymmTensorField>
   (
      new volSymmTensorField("DREff",Cs_*(k_/epsilon_)*R_+nu()*I)
   );
}

//- Return the effective diffusivity for epsilon
tmp<volSymmTensorField> DepsilonEff() const
{
   return tmp<volSymmTensorField>
   (
      new volSymmTensorField("DepsilonEff",
                             Ceps_*(k_/epsilon_)*R_+nu()*I)
   );
}

//- Return the turbulence kinetic energy
virtual tmp<volScalarField> k() const
{
   return k_;
}

//- Return the turbulence kinetic energy dissipation rate
virtual tmp<volScalarField> epsilon() const
{
   return epsilon_;
}

//- Return the Reynolds stress tensor
virtual tmp<volSymmTensorField> R() const
{
   return R_;
}

//- Return the effective stress tensor including the laminar stress
virtual tmp<volSymmTensorField> devReff() const;
```

```
    //- Return the source term for the momentum equation
    virtual tmp<fvVectorMatrix> divDevReff(volVectorField& U) const;

    //- Return the source term for the momentum equation
    virtual tmp<fvVectorMatrix> divDevRhoReff
    (
        const volScalarField& rho,
        volVectorField& U
    ) const;

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

    //- Read RASProperties dictionary
    virtual bool read();
};


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

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

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

#endif

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

B. Tables for OpenFOAM Data Shown in Plots

* Note only the OpenFOAM data is included here. The Experimental and NASA's CFL3D and FUN3D data can be obtained from http://turbmodels.larc.nasa.gov/index.html.

Final SST Results for Streamwise Velocity Profiles

x/c = 0.6753		x/c = 0.7308		x/c = 0.7863	
Probe Point	Streamwise Velocity	Probe Point	Streamwise Velocity	Probe Point	Streamwise Velocity
0.070614	0.330925	0.061402	0.010985	0.051038	-0.01272
0.070618	0.398689	0.061406	0.038822	0.051042	-0.01455
0.070622	0.468537	0.06141	0.065699	0.051046	-0.01606
0.070626	0.539223	0.061414	0.093157	0.051049	-0.0172
0.07063	0.610305	0.061418	0.121971	0.051053	-0.01816
0.070634	0.68668	0.061422	0.151236	0.051057	-0.0188
0.070638	0.76411	0.061426	0.181889	0.051061	-0.01898
0.070642	0.844343	0.06143	0.213848	0.051066	-0.01868
0.070647	0.926923	0.061435	0.24811	0.05107	-0.01809
0.070651	1.01289	0.061439	0.282831	0.051074	-0.0171
0.070656	1.10187	0.061444	0.32093	0.051079	-0.01559
0.070661	1.19522	0.061449	0.361208	0.051083	-0.01378
0.070665	1.29344	0.061453	0.402111	0.051088	-0.01125
0.070671	1.39468	0.061459	0.445854	0.051093	-0.00785
0.070676	1.50326	0.061464	0.492082	0.051099	-0.00391
0.070681	1.61594	0.061469	0.541993	0.051104	0.000784
0.070687	1.73686	0.061475	0.594616	0.05111	0.006271
0.070693	1.86336	0.061481	0.651126	0.051116	0.012981
0.0707	1.99657	0.061487	0.712495	0.051122	0.020832
0.070706	2.13477	0.061494	0.77542	0.051129	0.029687
0.070713	2.28098	0.061501	0.843659	0.051135	0.039899
0.07072	2.43212	0.061508	0.916747	0.051143	0.051975
0.070728	2.58744	0.061516	0.994759	0.05115	0.065578
0.070736	2.75014	0.061524	1.07693	0.051158	0.080877
0.070745	2.91809	0.061532	1.16301	0.051167	0.098386
0.070754	3.08486	0.061541	1.25265	0.051175	0.1177
0.070763	3.25673	0.06155	1.34764	0.051185	0.139292
0.070773	3.42315	0.06156	1.44487	0.051195	0.163161
0.070784	3.59193	0.061571	1.5428	0.051205	0.189063
0.070795	3.75799	0.061582	1.64191	0.051216	0.216958
0.070806	3.92083	0.061593	1.74212	0.051227	0.246586
0.070819	4.0792	0.061606	1.84152	0.05124	0.277737

0.070832	4.2334	0.061619	1.93795	0.051253	0.310165
0.070846	4.38292	0.061632	2.03316	0.051266	0.342967
0.07086	4.52934	0.061647	2.1255	0.051281	0.376103
0.070876	4.67141	0.061662	2.21531	0.051296	0.410092
0.070892	4.80783	0.061679	2.30211	0.051312	0.442873
0.07091	4.94216	0.061696	2.38629	0.051329	0.476153
0.070928	5.07185	0.061714	2.46769	0.051347	0.508923
0.070947	5.19829	0.061733	2.54668	0.051366	0.540816
0.070968	5.32209	0.061754	2.6237	0.051387	0.57301
0.07099	5.4429	0.061776	2.69862	0.051408	0.604753
0.071013	5.56116	0.061798	2.77189	0.051431	0.636298
0.071037	5.67723	0.061823	2.84368	0.051455	0.66756
0.071063	5.79237	0.061848	2.91506	0.051481	0.699004
0.071091	5.90511	0.061876	2.98471	0.051508	0.73027
0.07112	6.01734	0.061904	3.05396	0.051536	0.761587
0.07115	6.12755	0.061935	3.12277	0.051567	0.793142
0.071183	6.23825	0.061967	3.19148	0.051599	0.825315
0.071217	6.34787	0.062002	3.26006	0.051633	0.857707
0.071254	6.45729	0.062038	3.32887	0.051669	0.890252
0.071293	6.5672	0.062077	3.39775	0.051707	0.923699
0.071334	6.67703	0.062117	3.46725	0.051748	0.957772
0.071377	6.78678	0.062161	3.5374	0.051791	0.992523
0.071424	6.89811	0.062207	3.60866	0.051836	1.02804
0.071472	7.00992	0.062255	3.68052	0.051885	1.06467
0.071524	7.12295	0.062306	3.75379	0.051936	1.10227
0.071579	7.23747	0.062361	3.82851	0.05199	1.1408
0.071637	7.35346	0.062419	3.90452	0.052048	1.18077
0.071699	7.47108	0.06248	3.98251	0.052108	1.22191
0.071764	7.59122	0.062545	4.06243	0.052173	1.26446
0.071834	7.71358	0.062614	4.14418	0.052241	1.30876
0.071907	7.83782	0.062687	4.22856	0.052314	1.35464
0.071985	7.96553	0.062764	4.3154	0.052391	1.40233
0.072067	8.09598	0.062846	4.4048	0.052472	1.45192
0.072155	8.22968	0.062933	4.4973	0.052558	1.50356
0.072247	8.36719	0.063025	4.59289	0.05265	1.55763
0.072346	8.50856	0.063122	4.6921	0.052747	1.61396
0.07245	8.65385	0.063226	4.79498	0.052849	1.67301
0.07256	8.80383	0.063335	4.90179	0.052958	1.73473
0.072677	8.9588	0.063451	5.01284	0.053074	1.7996
0.072801	9.11861	0.063574	5.12866	0.053196	1.86768
0.072932	9.28439	0.063705	5.24939	0.053326	1.93912
0.073071	9.45603	0.063843	5.37536	0.053374	1.97603
0.073219	9.63398	0.06399	5.5071	0.053463	2.02486
0.073375	9.81906	0.064145	5.64487	0.053609	2.10441
0.073541	10.0114	0.06431	5.78932	0.053763	2.1883

0.073716	10.2118	0.064484	5.94072	0.053927	2.27686
0.073902	10.4207	0.064669	6.09973	0.0541	2.37039
0.0741	10.6386	0.064865	6.26674	0.054284	2.46944
0.074309	10.8663	0.065073	6.44237	0.054478	2.57427
0.07453	11.1044	0.065293	6.62731	0.054685	2.68551
0.074765	11.3535	0.065526	6.82219	0.054904	2.80348
0.075014	11.6149	0.065774	7.02785	0.055135	2.92883
0.075278	11.8892	0.066036	7.24475	0.055381	3.06216
0.075558	12.1773	0.066314	7.47395	0.055642	3.2041
0.075854	12.4804	0.066608	7.71633	0.055918	3.35536
0.076168	12.7996	0.066921	7.973	0.056211	3.51675
0.076501	13.1361	0.067252	8.24456	0.056521	3.68902
0.076855	13.4913	0.067603	8.53274	0.05685	3.87314
0.077229	13.8668	0.067974	8.83823	0.057199	4.07012
0.077625	14.2642	0.068368	9.16258	0.057568	4.28106
0.078046	14.6854	0.068786	9.50753	0.05796	4.50704
0.078491	15.1325	0.069229	9.87409	0.058375	4.74945
0.078963	15.6117	0.069698	10.2646	0.058815	5.00958
0.079464	16.1289	0.070195	10.6802	0.059281	5.28918
0.079994	16.6885	0.070723	11.1244	0.059776	5.58966
0.080557	17.2948	0.071282	11.6024	0.0603	5.91299
0.081153	17.9514	0.071874	12.1196	0.060855	6.26115
0.081784	18.6621	0.072501	12.6818	0.061444	6.63631
0.082454	19.4317	0.073167	13.2922	0.062068	7.04085
0.083163	20.2659	0.073872	13.9555	0.062729	7.47831
0.083916	21.1709	0.07462	14.6756	0.06343	7.95498
0.084713	22.154	0.075412	15.4578	0.064173	8.47815
0.085558	23.2215	0.075729	15.8161	0.064961	9.05393
0.085571	23.2851	0.076252	16.3474	0.065795	9.68693
0.086453	24.4238	0.077142	17.2769	0.06668	10.3816
0.087402	25.6608	0.078085	18.2912	0.067618	11.1429
0.088408	26.9595	0.079084	19.3996	0.068612	11.9766
0.089474	28.2836	0.080144	20.6098	0.069666	12.8904
0.090603	29.5856	0.081267	21.9234	0.070783	13.8937
0.091801	30.8343	0.082458	23.3301	0.071966	14.9977
0.09307	32.0329	0.083719	24.8009	0.073221	16.2146
0.094416	33.1833	0.085057	26.2867	0.074551	17.5544
0.095842	34.2465	0.086474	27.7382	0.075961	19.022
0.097353	35.1449	0.087977	29.1412	0.077455	20.608
0.098955	35.7977	0.089569	30.5069	0.079039	22.279
		0.091257	31.8161	0.080718	23.9753
		0.093047	32.9995	0.082497	25.6426
		0.094943	33.9584	0.08322	26.3159
		0.096953	34.6105	0.084383	27.3217
		0.099084	34.9486	0.086382	28.9243

				0.088501	30.4641
				0.090746	31.8494
				0.093126	32.9589
				0.095649	33.6983
				0.098324	34.0699

x/c = 0.8418		x/c = 0.8973		x/c = 0.9528	
Probe Point	Streamwise Velocity	Probe Point	Streamwise Velocity	Probe Point	Streamwise Velocity
0.039513	0.003404	0.026805	-0.08808	0.01287	-0.03151
0.039517	-0.01324	0.026809	-0.11156	0.012873	-0.06034
0.039521	-0.03025	0.026813	-0.13576	0.012877	-0.09163
0.039524	-0.04693	0.026816	-0.15992	0.012881	-0.12244
0.039528	-0.0645	0.02682	-0.18364	0.012885	-0.15351
0.039532	-0.08138	0.026824	-0.20811	0.012889	-0.18497
0.039536	-0.09836	0.026828	-0.23305	0.012893	-0.21662
0.03954	-0.11546	0.026832	-0.25804	0.012897	-0.24934
0.039545	-0.13389	0.026837	-0.28383	0.012901	-0.28265
0.039549	-0.15143	0.026841	-0.31016	0.012906	-0.31771
0.039554	-0.16915	0.026846	-0.33728	0.01291	-0.35127
0.039558	-0.18744	0.02685	-0.36447	0.012915	-0.38845
0.039563	-0.20699	0.026855	-0.39305	0.01292	-0.42526
0.039568	-0.22586	0.02686	-0.4225	0.012925	-0.46382
0.039573	-0.24515	0.026865	-0.45251	0.01293	-0.50464
0.039579	-0.26584	0.026871	-0.48408	0.012935	-0.54629
0.039585	-0.28617	0.026876	-0.51617	0.012941	-0.58903
0.03959	-0.30662	0.026882	-0.54948	0.012947	-0.63322
0.039597	-0.32814	0.026888	-0.58411	0.012953	-0.68034
0.039603	-0.35011	0.026895	-0.62	0.01296	-0.72926
0.03961	-0.3722	0.026902	-0.65737	0.012966	-0.77985
0.039617	-0.3954	0.026909	-0.69588	0.012974	-0.83291
0.039625	-0.4181	0.026916	-0.73577	0.012981	-0.88784
0.039633	-0.44148	0.026924	-0.77696	0.012989	-0.94495
0.039641	-0.46491	0.026932	-0.81929	0.012997	-1.00439
0.03965	-0.48843	0.026941	-0.86322	0.013006	-1.0654
0.039659	-0.51178	0.02695	-0.9078	0.013015	-1.12936
0.039669	-0.53504	0.02696	-0.95339	0.013025	-1.19628
0.039679	-0.55709	0.02697	-0.99936	0.013035	-1.26323
0.03969	-0.57857	0.026981	-1.04556	0.013046	-1.33334
0.039701	-0.59862	0.026993	-1.09161	0.013057	-1.40213
0.039714	-0.61719	0.027005	-1.13661	0.01307	-1.47289
0.039726	-0.63413	0.027017	-1.18037	0.013083	-1.54223
0.03974	-0.6486	0.027031	-1.22228	0.013096	-1.61012

0.039754	-0.66108	0.027045	-1.26164	0.01311	-1.67667
0.039769	-0.67122	0.02706	-1.29796	0.013125	-1.7402
0.039786	-0.67927	0.027076	-1.33124	0.013142	-1.80098
0.039803	-0.68517	0.027093	-1.36161	0.013159	-1.85743
0.039821	-0.68906	0.027111	-1.38852	0.013176	-1.90929
0.03984	-0.69117	0.02713	-1.41255	0.013196	-1.95779
0.03986	-0.69162	0.027151	-1.43361	0.013216	-2.00152
0.039881	-0.69055	0.027172	-1.45189	0.013237	-2.04158
0.039904	-0.68821	0.027194	-1.46782	0.01326	-2.07765
0.039928	-0.68478	0.027218	-1.48155	0.013284	-2.11021
0.039953	-0.68007	0.027243	-1.49327	0.013309	-2.13958
0.03998	-0.67437	0.02727	-1.50319	0.013336	-2.16593
0.040009	-0.66778	0.027299	-1.51141	0.013364	-2.18969
0.040039	-0.66015	0.027329	-1.51812	0.013394	-2.21087
0.040071	-0.65166	0.027361	-1.52342	0.013426	-2.22992
0.040105	-0.64235	0.027394	-1.52745	0.01346	-2.24682
0.040141	-0.6322	0.02743	-1.53028	0.013496	-2.26189
0.040179	-0.62117	0.027468	-1.53195	0.013534	-2.27516
0.040219	-0.60932	0.027508	-1.53254	0.013574	-2.28683
0.040262	-0.5966	0.027551	-1.53212	0.013617	-2.29707
0.040307	-0.58307	0.027596	-1.53068	0.013662	-2.30584
0.040355	-0.5685	0.027644	-1.52827	0.01371	-2.31331
0.040406	-0.5531	0.027694	-1.52491	0.013761	-2.31946
0.04046	-0.53672	0.027748	-1.52059	0.013815	-2.32441
0.040517	-0.5194	0.027805	-1.51537	0.013872	-2.32817
0.040577	-0.50099	0.027865	-1.50917	0.013932	-2.33077
0.040641	-0.48152	0.027929	-1.50205	0.013996	-2.33226
0.040709	-0.46093	0.027997	-1.49395	0.014064	-2.33264
0.040781	-0.43917	0.028069	-1.48489	0.014136	-2.3319
0.040858	-0.41612	0.028145	-1.47484	0.014212	-2.33008
0.040939	-0.3918	0.028226	-1.46375	0.014293	-2.32715
0.041025	-0.36604	0.028311	-1.45158	0.014379	-2.32311
0.041115	-0.33886	0.028402	-1.43835	0.01447	-2.31795
0.041212	-0.31007	0.028498	-1.42397	0.014566	-2.31164
0.041314	-0.2797	0.028599	-1.40844	0.014668	-2.30417
0.041422	-0.2475	0.028707	-1.39164	0.014777	-2.2955
0.041537	-0.2135	0.028822	-1.37358	0.014891	-2.28558
0.041659	-0.17753	0.028943	-1.35414	0.015013	-2.27443
0.041788	-0.13937	0.029071	-1.33329	0.015142	-2.26194
0.041924	-0.09901	0.029208	-1.31097	0.015278	-2.24812
0.042069	-0.05619	0.029352	-1.28705	0.015423	-2.23288
0.042223	-0.01083	0.029505	-1.26147	0.015576	-2.21615
0.042385	0.037286	0.029667	-1.2341	0.015739	-2.19792
0.042558	0.088396	0.029839	-1.20492	0.015911	-2.17805
0.042741	0.142637	0.030021	-1.17368	0.016094	-2.15653

0.042934	0.200334	0.030214	-1.14038	0.016288	-2.13321
0.04314	0.261726	0.030419	-1.10479	0.016448	-2.11258
0.043358	0.327177	0.030636	-1.06682	0.016493	-2.10707
0.043588	0.39683	0.030866	-1.02625	0.01671	-2.07992
0.043833	0.471123	0.03111	-0.9829	0.016941	-2.05067
0.044092	0.550524	0.031368	-0.9366	0.017186	-2.01927
0.044367	0.63527	0.031642	-0.88712	0.017445	-1.98557
0.044658	0.726033	0.031932	-0.8342	0.017719	-1.94943
0.044967	0.823285	0.03224	-0.77752	0.018011	-1.91067
0.04502	0.84764	0.032566	-0.71691	0.018319	-1.86916
0.045294	0.935339	0.032912	-0.65189	0.018647	-1.82471
0.045641	1.04762	0.033279	-0.58218	0.018993	-1.77714
0.046009	1.16836	0.033667	-0.50726	0.019361	-1.72622
0.046398	1.29835	0.034079	-0.42682	0.019751	-1.67172
0.046811	1.43839	0.034515	-0.34023	0.020164	-1.61337
0.047249	1.58961	0.034978	-0.24692	0.020601	-1.55089
0.047713	1.75295	0.035469	-0.14624	0.021065	-1.48397
0.048205	1.92975	0.035988	-0.03748	0.021557	-1.41224
0.048726	2.12138	0.036539	0.080195	0.022079	-1.33531
0.049279	2.3294	0.037123	0.207823	0.022632	-1.25271
0.049864	2.55548	0.037742	0.346389	0.023218	-1.16398
0.050485	2.80156	0.038022	0.416957	0.023838	-1.06862
0.051143	3.06984	0.038398	0.503658	0.024497	-0.96586
0.051841	3.36273	0.039094	0.668675	0.025195	-0.85508
0.05258	3.68291	0.039831	0.848949	0.025935	-0.73543
0.053364	4.0335	0.040613	1.04635	0.026719	-0.60601
0.054194	4.41793	0.041441	1.26299	0.02755	-0.46572
0.055075	4.8401	0.042319	1.50132	0.028431	-0.31333
0.056008	5.3063	0.04325	1.76413	0.028916	-0.22192
0.056997	5.82729	0.044236	2.05482	0.029365	-0.14195
0.058046	6.41411	0.045282	2.37698	0.030355	0.039633
0.059157	7.07588	0.04639	2.73515	0.031405	0.238445
0.060335	7.81933	0.047565	3.13437	0.032518	0.456763
0.061584	8.65192	0.048811	3.58041	0.033697	0.697254
0.062908	9.58309	0.050131	4.08234	0.034947	0.963064
0.064311	10.6239	0.051531	4.65491	0.036273	1.25809
0.065798	11.7883	0.053014	5.31648	0.037678	1.58682
0.066398	12.3338	0.053513	5.58865	0.039167	1.9546
0.067374	13.1433	0.054586	6.11538	0.040746	2.36803
0.069045	14.6042	0.056253	7.01053	0.041646	2.63469
0.070816	16.2229	0.05802	8.04523	0.04242	2.8513
0.072693	17.987	0.059893	9.23872	0.044194	3.38381
0.074682	19.855	0.061878	10.6096	0.046075	3.99471
0.076791	21.7592	0.063983	12.173	0.04807	4.70901
0.079027	23.6517	0.066214	13.9313	0.050184	5.55854

0.081397	25.5383	0.068579	15.8622	0.052425	6.57634
0.083909	27.4237	0.069068	16.3278	0.054675	7.77465
0.086571	29.2535	0.071085	17.9751	0.054801	7.8393
0.087701	29.9563	0.073743	20.0622	0.05732	9.29435
0.089394	30.9418	0.07656	22.1644	0.059991	10.9903
0.092385	32.2437	0.079546	24.3045	0.062822	12.9134
0.095556	33.0954	0.082711	26.4723	0.065823	15.0134
0.098917	33.5067	0.084687	27.7726	0.068013	16.595
		0.086067	28.6368	0.069006	17.2801
		0.089624	30.4956	0.07238	19.5218
		0.093396	31.9211	0.075957	21.8284
		0.097393	32.7924	0.07975	24.2219
				0.081707	25.474
				0.083771	26.7147
				0.088035	28.9785
				0.092555	30.8454
				0.095782	31.7148
				0.097349	32.1226

Final SST Results for Transverse Velocity Profiles

x/c = 0.6753		x/c = 0.7308		x/c = 0.7863	
Probe Point	Transverse Velocity	Probe Point	Transverse Velocity	Probe Point	Transverse Velocity
0.070614	-0.05327	0.061402	-0.00189	0.051038	0.003052
0.070618	-0.06146	0.061406	-0.00694	0.051042	0.003028
0.070622	-0.07219	0.06141	-0.01174	0.051046	0.003313
0.070626	-0.08449	0.061414	-0.0166	0.051049	0.003645
0.07063	-0.09514	0.061418	-0.0217	0.051053	0.003425
0.070634	-0.1057	0.061422	-0.02682	0.051057	0.003297
0.070638	-0.11671	0.061426	-0.03221	0.051061	0.003688
0.070642	-0.13001	0.06143	-0.03793	0.051066	0.003975
0.070647	-0.14432	0.061435	-0.04429	0.05107	0.003802
0.070651	-0.15754	0.061439	-0.05009	0.051074	0.003587
0.070656	-0.17024	0.061444	-0.05624	0.051079	0.003055
0.070661	-0.18394	0.061449	-0.06394	0.051083	0.00246
0.070665	-0.20007	0.061453	-0.07142	0.051088	0.002325
0.070671	-0.21572	0.061459	-0.07878	0.051093	0.001927
0.070676	-0.23187	0.061464	-0.08694	0.051099	0.001201
0.070681	-0.24994	0.061469	-0.09598	0.051104	0.000141
0.070687	-0.26793	0.061475	-0.10498	0.05111	-0.0013

0.070693	-0.2864	0.061481	-0.11432	0.051116	-0.00224
0.0707	-0.30772	0.061487	-0.1256	0.051122	-0.00336
0.070706	-0.3298	0.061494	-0.13699	0.051129	-0.00542
0.070713	-0.35165	0.061501	-0.14889	0.051135	-0.00777
0.07072	-0.37484	0.061508	-0.16137	0.051143	-0.00955
0.070728	-0.39936	0.061516	-0.1745	0.05115	-0.01187
0.070736	-0.42409	0.061524	-0.18945	0.051158	-0.01533
0.070745	-0.44879	0.061532	-0.20482	0.051167	-0.01846
0.070754	-0.47422	0.061541	-0.22001	0.051175	-0.02212
0.070763	-0.50164	0.06155	-0.23641	0.051185	-0.02658
0.070773	-0.52739	0.06156	-0.25357	0.051195	-0.03073
0.070784	-0.55271	0.061571	-0.27071	0.051205	-0.0356
0.070795	-0.57729	0.061582	-0.28769	0.051216	-0.04122
0.070806	-0.60201	0.061593	-0.3048	0.051227	-0.04671
0.070819	-0.62665	0.061606	-0.32223	0.05124	-0.05252
0.070832	-0.65029	0.061619	-0.33897	0.051253	-0.05855
0.070846	-0.67272	0.061632	-0.35506	0.051266	-0.06486
0.07086	-0.69412	0.061647	-0.37087	0.051281	-0.07109
0.070876	-0.71573	0.061662	-0.38618	0.051296	-0.07723
0.070892	-0.73654	0.061679	-0.40089	0.051312	-0.08329
0.07091	-0.75601	0.061696	-0.41506	0.051329	-0.08942
0.070928	-0.7753	0.061714	-0.42868	0.051347	-0.09537
0.070947	-0.79406	0.061733	-0.44183	0.051366	-0.1011
0.070968	-0.81209	0.061754	-0.45454	0.051387	-0.10682
0.07099	-0.82992	0.061776	-0.46683	0.051408	-0.11243
0.071013	-0.84694	0.061798	-0.47877	0.051431	-0.11795
0.071037	-0.86355	0.061823	-0.49026	0.051455	-0.12333
0.071063	-0.88005	0.061848	-0.5016	0.051481	-0.12871
0.071091	-0.89601	0.061876	-0.51276	0.051508	-0.13398
0.07112	-0.9119	0.061904	-0.52359	0.051536	-0.13918
0.07115	-0.92725	0.061935	-0.53413	0.051567	-0.14434
0.071183	-0.94255	0.061967	-0.54461	0.051599	-0.14951
0.071217	-0.95768	0.062002	-0.555	0.051633	-0.15474
0.071254	-0.97245	0.062038	-0.56527	0.051669	-0.15988
0.071293	-0.98718	0.062077	-0.57541	0.051707	-0.165
0.071334	-1.00186	0.062117	-0.58547	0.051748	-0.17023
0.071377	-1.01624	0.062161	-0.5955	0.051791	-0.17548
0.071424	-1.03069	0.062207	-0.60554	0.051836	-0.18072
0.071472	-1.04509	0.062255	-0.61552	0.051885	-0.18605
0.071524	-1.05939	0.062306	-0.62552	0.051936	-0.19145
0.071579	-1.07372	0.062361	-0.63556	0.05199	-0.1969
0.071637	-1.08806	0.062419	-0.64562	0.052048	-0.20245
0.071699	-1.10234	0.06248	-0.65575	0.052108	-0.20806
0.071764	-1.11672	0.062545	-0.66597	0.052173	-0.21376
0.071834	-1.13123	0.062614	-0.67621	0.052241	-0.2196

0.071907	-1.14567	0.062687	-0.68657	0.052314	-0.22555
0.071985	-1.16025	0.062764	-0.6971	0.052391	-0.23162
0.072067	-1.17496	0.062846	-0.70771	0.052472	-0.23782
0.072155	-1.18976	0.062933	-0.71846	0.052558	-0.24417
0.072247	-1.20472	0.063025	-0.72935	0.05265	-0.25068
0.072346	-1.21985	0.063122	-0.74044	0.052747	-0.25734
0.07245	-1.23508	0.063226	-0.7517	0.052849	-0.2642
0.07256	-1.25054	0.063335	-0.76316	0.052958	-0.27124
0.072677	-1.26624	0.063451	-0.77482	0.053074	-0.27848
0.072801	-1.28209	0.063574	-0.78672	0.053196	-0.28594
0.072932	-1.2982	0.063705	-0.79887	0.053326	-0.29362
0.073071	-1.3146	0.063843	-0.81126	0.053374	-0.29768
0.073219	-1.33123	0.06399	-0.82394	0.053463	-0.30282
0.073375	-1.34819	0.064145	-0.83691	0.053609	-0.31102
0.073541	-1.36547	0.06431	-0.85019	0.053763	-0.3195
0.073716	-1.38306	0.064484	-0.8638	0.053927	-0.32827
0.073902	-1.40104	0.064669	-0.87777	0.0541	-0.33733
0.0741	-1.41941	0.064865	-0.89212	0.054284	-0.34674
0.074309	-1.43821	0.065073	-0.90685	0.054478	-0.35648
0.07453	-1.45745	0.065293	-0.922	0.054685	-0.36659
0.074765	-1.47715	0.065526	-0.93758	0.054904	-0.37708
0.075014	-1.49738	0.065774	-0.95365	0.055135	-0.38798
0.075278	-1.5182	0.066036	-0.9702	0.055381	-0.39931
0.075558	-1.5396	0.066314	-0.98726	0.055642	-0.4111
0.075854	-1.56162	0.066608	-1.00487	0.055918	-0.42337
0.076168	-1.58438	0.066921	-1.02309	0.056211	-0.43616
0.076501	-1.60787	0.067252	-1.0419	0.056521	-0.44947
0.076855	-1.63217	0.067603	-1.06138	0.05685	-0.46335
0.077229	-1.65736	0.067974	-1.08154	0.057199	-0.47782
0.077625	-1.6835	0.068368	-1.10242	0.057568	-0.49292
0.078046	-1.71069	0.068786	-1.12411	0.05796	-0.50866
0.078491	-1.73912	0.069229	-1.1466	0.058375	-0.5251
0.078963	-1.76929	0.069698	-1.16998	0.058815	-0.54224
0.079464	-1.80176	0.070195	-1.19427	0.059281	-0.56014
0.079994	-1.83688	0.070723	-1.21973	0.059776	-0.57881
0.080557	-1.87487	0.071282	-1.24681	0.0603	-0.59829
0.081153	-1.9159	0.071874	-1.27596	0.060855	-0.61862
0.081784	-1.96008	0.072501	-1.30766	0.061444	-0.63982
0.082454	-2.0076	0.073167	-1.34205	0.062068	-0.66195
0.083163	-2.05874	0.073872	-1.37928	0.062729	-0.68522
0.083916	-2.1138	0.07462	-1.41945	0.06343	-0.71011
0.084713	-2.17309	0.075412	-1.46275	0.064173	-0.73713
0.085558	-2.23677	0.075729	-1.48305	0.064961	-0.76666
0.085571	-2.23882	0.076252	-1.51221	0.065795	-0.79886
0.086453	-2.30551	0.077142	-1.56278	0.06668	-0.83378

0.087402	-2.37621	0.078085	-1.61745	0.067618	-0.87147
0.088408	-2.44761	0.079084	-1.67658	0.068612	-0.91205
0.089474	-2.51601	0.080144	-1.74025	0.069666	-0.95572
0.090603	-2.57755	0.081267	-1.80789	0.070783	-1.00282
0.091801	-2.62989	0.082458	-1.87773	0.071966	-1.05373
0.09307	-2.67329	0.083719	-1.94631	0.073221	-1.10882
0.094416	-2.70779	0.085057	-2.00943	0.074551	-1.16808
0.095842	-2.73109	0.086474	-2.06371	0.075961	-1.23077
0.097353	-2.7395	0.087977	-2.10844	0.077455	-1.29488
0.098955	-2.73004	0.089569	-2.14406	0.079039	-1.3563
		0.091257	-2.16954	0.080718	-1.41074
		0.093047	-2.18225	0.082497	-1.45493
		0.094943	-2.17923	0.08322	-1.46994
		0.096953	-2.16033	0.084383	-1.49071
		0.099084	-2.12867	0.086382	-1.51411
				0.088501	-1.52595
				0.090746	-1.52481
				0.093126	-1.51015
				0.095649	-1.48521
				0.098324	-1.45472

x/c = 0.8418		x/c = 0.8973		x/c = 0.9528	
Probe Point	Transverse Velocity	Probe Point	Transverse Velocity	Probe Point	Transverse Velocity
0.039513	-0.00068	0.026805	0.021075	0.01287	0.008349
0.039517	0.002844	0.026809	0.026769	0.012873	0.01602
0.039521	0.006472	0.026813	0.032636	0.012877	0.024304
0.039524	0.010016	0.026816	0.038398	0.012881	0.032452
0.039528	0.013791	0.02682	0.044099	0.012885	0.040693
0.039532	0.017515	0.026824	0.050032	0.012889	0.049035
0.039536	0.021197	0.026828	0.055985	0.012893	0.057445
0.03954	0.024848	0.026832	0.061992	0.012897	0.066103
0.039545	0.028753	0.026837	0.068222	0.012901	0.07495
0.039549	0.032525	0.026841	0.074535	0.012906	0.084261
0.039554	0.036444	0.026846	0.081098	0.01291	0.093136
0.039558	0.040338	0.02685	0.087631	0.012915	0.102969
0.039563	0.044487	0.026855	0.094473	0.01292	0.112707
0.039568	0.048632	0.02686	0.101578	0.012925	0.122968
0.039573	0.052827	0.026865	0.108796	0.01293	0.133811
0.039579	0.057233	0.026871	0.116393	0.012935	0.144791
0.039585	0.061554	0.026876	0.124127	0.012941	0.156088
0.03959	0.066108	0.026882	0.132158	0.012947	0.167839
0.039597	0.070908	0.026888	0.140485	0.012953	0.180338

0.039603	0.075492	0.026895	0.149119	0.01296	0.193272
0.03961	0.080211	0.026902	0.158111	0.012966	0.206674
0.039617	0.085252	0.026909	0.167422	0.012974	0.220719
0.039625	0.090256	0.026916	0.177043	0.012981	0.235257
0.039633	0.095443	0.026924	0.186909	0.012989	0.250392
0.039641	0.100459	0.026932	0.197171	0.012997	0.266143
0.03965	0.105655	0.026941	0.20779	0.013006	0.282298
0.039659	0.110707	0.02695	0.218456	0.013015	0.299243
0.039669	0.115697	0.02696	0.229466	0.013025	0.316969
0.039679	0.120571	0.02697	0.24063	0.013035	0.334701
0.03969	0.125377	0.026981	0.251789	0.013046	0.353287
0.039701	0.129931	0.026993	0.262903	0.013057	0.371543
0.039714	0.133896	0.027005	0.273801	0.01307	0.390335
0.039726	0.137608	0.027017	0.284397	0.013083	0.408749
0.03974	0.141114	0.027031	0.294548	0.013096	0.426769
0.039754	0.144025	0.027045	0.304136	0.01311	0.444455
0.039769	0.146268	0.02706	0.312998	0.013125	0.46141
0.039786	0.148168	0.027076	0.321127	0.013142	0.477647
0.039803	0.14972	0.027093	0.328579	0.013159	0.492711
0.039821	0.150853	0.027111	0.335214	0.013176	0.5066
0.03984	0.151598	0.02713	0.341182	0.013196	0.519666
0.03986	0.152075	0.027151	0.346446	0.013216	0.531489
0.039881	0.152204	0.027172	0.351065	0.013237	0.542349
0.039904	0.151994	0.027194	0.355144	0.01326	0.55219
0.039928	0.151679	0.027218	0.358711	0.013284	0.561144
0.039953	0.151178	0.027243	0.361824	0.013309	0.569279
0.03998	0.150405	0.02727	0.364527	0.013336	0.576664
0.040009	0.149518	0.027299	0.366845	0.013364	0.583407
0.040039	0.148483	0.027329	0.368832	0.013394	0.589495
0.040071	0.147267	0.027361	0.370504	0.013426	0.595068
0.040105	0.145927	0.027394	0.371903	0.01346	0.600122
0.040141	0.144476	0.02743	0.373048	0.013496	0.604741
0.040179	0.142898	0.027468	0.373956	0.013534	0.608926
0.040219	0.141181	0.027508	0.374635	0.013574	0.612738
0.040262	0.139342	0.027551	0.375112	0.013617	0.616239
0.040307	0.137417	0.027596	0.37539	0.013662	0.619407
0.040355	0.135349	0.027644	0.375477	0.01371	0.622298
0.040406	0.133163	0.027694	0.37539	0.013761	0.624904
0.04046	0.130859	0.027748	0.375123	0.013815	0.627264
0.040517	0.128448	0.027805	0.374696	0.013872	0.629387
0.040577	0.125907	0.027865	0.374095	0.013932	0.631284
0.040641	0.123229	0.027929	0.373337	0.013996	0.632974
0.040709	0.120426	0.027997	0.372416	0.014064	0.634457
0.040781	0.117501	0.028069	0.371336	0.014136	0.635747
0.040858	0.114433	0.028145	0.370097	0.014212	0.636847

0.040939	0.111227	0.028226	0.368698	0.014293	0.637767
0.041025	0.107865	0.028311	0.367135	0.014379	0.638506
0.041115	0.10436	0.028402	0.365413	0.01447	0.639071
0.041212	0.10069	0.028498	0.363522	0.014566	0.639466
0.041314	0.096868	0.028599	0.361471	0.014668	0.639692
0.041422	0.092861	0.028707	0.35924	0.014777	0.639748
0.041537	0.088678	0.028822	0.356837	0.014891	0.639632
0.041659	0.084315	0.028943	0.354253	0.015013	0.639355
0.041788	0.079748	0.029071	0.351482	0.015142	0.6389
0.041924	0.07498	0.029208	0.348526	0.015278	0.63828
0.042069	0.06999	0.029352	0.345369	0.015423	0.637481
0.042223	0.06478	0.029505	0.342008	0.015576	0.636503
0.042385	0.059332	0.029667	0.338432	0.015739	0.635347
0.042558	0.053628	0.029839	0.334644	0.015911	0.634002
0.042741	0.047666	0.030021	0.330617	0.016094	0.632466
0.042934	0.041424	0.030214	0.326359	0.016288	0.630725
0.04314	0.03489	0.030419	0.321846	0.016448	0.628832
0.043358	0.028032	0.030636	0.317077	0.016493	0.628411
0.043588	0.020858	0.030866	0.312032	0.01671	0.626267
0.043833	0.01334	0.03111	0.306697	0.016941	0.623888
0.044092	0.005442	0.031368	0.301062	0.017186	0.621278
0.044367	-0.00283	0.031642	0.295108	0.017445	0.618419
0.044658	-0.01152	0.031932	0.288821	0.017719	0.6153
0.044967	-0.02066	0.03224	0.282172	0.018011	0.611899
0.04502	-0.0223	0.032566	0.27516	0.018319	0.608207
0.045294	-0.03039	0.032912	0.267744	0.018647	0.604202
0.045641	-0.04053	0.033279	0.259912	0.018993	0.599869
0.046009	-0.05121	0.033667	0.251624	0.019361	0.595184
0.046398	-0.06245	0.034079	0.242871	0.019751	0.590122
0.046811	-0.07427	0.034515	0.23361	0.020164	0.584657
0.047249	-0.08673	0.034978	0.223807	0.020601	0.578755
0.047713	-0.09984	0.035469	0.213428	0.021065	0.572388
0.048205	-0.11367	0.035988	0.202438	0.021557	0.565515
0.048726	-0.12823	0.036539	0.190788	0.022079	0.558102
0.049279	-0.14359	0.037123	0.178428	0.022632	0.5501
0.049864	-0.15977	0.037742	0.165322	0.023218	0.541461
0.050485	-0.17681	0.038022	0.158851	0.023838	0.532138
0.051143	-0.19477	0.038398	0.150861	0.024497	0.522051
0.051841	-0.21369	0.039094	0.136049	0.025195	0.511146
0.05258	-0.23359	0.039831	0.120313	0.025935	0.499348
0.053364	-0.25453	0.040613	0.103588	0.026719	0.486568
0.054194	-0.27654	0.041441	0.085806	0.02755	0.472706
0.055075	-0.29972	0.042319	0.066897	0.028431	0.457663
0.056008	-0.32436	0.04325	0.046795	0.028916	0.448272
0.056997	-0.35104	0.044236	0.025408	0.029365	0.440379

0.058046	-0.38031	0.045282	0.002676	0.030355	0.422529
0.059157	-0.41242	0.04639	-0.02149	0.031405	0.403104
0.060335	-0.44741	0.047565	-0.04715	0.032518	0.381942
0.061584	-0.48529	0.048811	-0.07446	0.033697	0.358869
0.062908	-0.52623	0.050131	-0.10374	0.034947	0.333698
0.064311	-0.57047	0.051531	-0.13551	0.036273	0.306197
0.065798	-0.61842	0.053014	-0.17041	0.037678	0.276141
0.066398	-0.64065	0.053513	-0.18396	0.039167	0.243287
0.067374	-0.67292	0.054586	-0.21021	0.040746	0.207362
0.069045	-0.72889	0.056253	-0.2525	0.041646	0.184822
0.070816	-0.78748	0.05802	-0.29887	0.04242	0.16666
0.072693	-0.84568	0.059893	-0.3496	0.044194	0.123792
0.074682	-0.89842	0.061878	-0.40467	0.046075	0.076366
0.076791	-0.94143	0.063983	-0.46326	0.04807	0.022817
0.079027	-0.97217	0.066214	-0.52309	0.050184	-0.03786
0.081397	-0.99021	0.068579	-0.57912	0.052425	-0.1055
0.083909	-0.99583	0.069068	-0.58988	0.054675	-0.17799
0.086571	-0.98785	0.071085	-0.62714	0.054801	-0.18191
0.087701	-0.97919	0.073743	-0.65966	0.05732	-0.26071
0.089394	-0.96622	0.07656	-0.67742	0.059991	-0.34161
0.092385	-0.93438	0.079546	-0.67881	0.062822	-0.4199
0.095556	-0.90034	0.082711	-0.66286	0.065823	-0.48797
0.098917	-0.87041	0.084687	-0.64257	0.068013	-0.52578
		0.086067	-0.62897	0.069006	-0.54225
		0.089624	-0.58111	0.07238	-0.57648
		0.093396	-0.52871	0.075957	-0.58949
		0.097393	-0.48547	0.07975	-0.57933
				0.081707	-0.56111
				0.083771	-0.54304
				0.088035	-0.48489
				0.092555	-0.41414
				0.095782	-0.36975
				0.097349	-0.34894

Final k-ω Results for Streamwise Velocity Profiles

x/c = 0.6753		x/c = 0.7308		x/c = 0.7863	
Probe Point	Streamwise Velocity	Probe Point	Streamwise Velocity	Probe Point	Streamwise Velocity
0.072613	16.3624	0.0634	13.3625	0.053034	9.89957
0.072616	16.369	0.063453	13.4465	0.053076	9.95557

0.072676	16.4744	0.063513	13.5419	0.053136	10.0381
0.072736	16.5802	0.063576	13.6384	0.053198	10.1225
0.072799	16.6872	0.06364	13.7358	0.053262	10.2072
0.072864	16.7952	0.063706	13.8342	0.053328	10.2938
0.07293	16.9044	0.063774	13.9339	0.053395	10.3809
0.072999	17.0146	0.063844	14.0347	0.053451	10.4535
0.073069	17.1258	0.063917	14.1367	0.053465	10.4714
0.073142	17.2384	0.063991	14.2398	0.053537	10.5613
0.073217	17.3522	0.064067	14.3443	0.053611	10.6525
0.073294	17.4673	0.064146	14.45	0.053687	10.7449
0.073373	17.5835	0.064227	14.5573	0.053766	10.8387
0.073454	17.7014	0.064311	14.6658	0.053846	10.934
0.073538	17.8203	0.064397	14.7758	0.053929	11.0307
0.073625	17.9411	0.064485	14.8874	0.054015	11.1288
0.073714	18.0634	0.064576	15.0004	0.054103	11.2285
0.073805	18.1869	0.06467	15.1151	0.054193	11.3299
0.073899	18.3124	0.064767	15.2314	0.054287	11.4329
0.073997	18.4395	0.064866	15.3494	0.054383	11.5374
0.074097	18.5683	0.064969	15.4692	0.054482	11.6438
0.074199	18.699	0.065074	15.5908	0.054583	11.752
0.074305	18.8316	0.065183	15.7143	0.054688	11.8621
0.074414	18.966	0.065294	15.8397	0.054796	11.9739
0.074527	19.1024	0.065409	15.9672	0.054907	12.0878
0.074642	19.241	0.065528	16.0966	0.055022	12.2037
0.074761	19.3816	0.06565	16.2281	0.05514	12.3216
0.074884	19.5245	0.065775	16.3621	0.055261	12.4417
0.07501	19.6698	0.065904	16.498	0.055386	12.5639
0.07514	19.8172	0.066037	16.6364	0.055514	12.6884
0.075273	19.9671	0.066174	16.7772	0.055646	12.8153
0.075411	20.1197	0.066315	16.9204	0.055783	12.9445
0.075553	20.2747	0.066461	17.0663	0.055923	13.0762
0.075699	20.4324	0.06661	17.2147	0.056067	13.2104
0.075849	20.5928	0.066764	17.3657	0.056216	13.3471
0.076003	20.7561	0.066922	17.5196	0.056369	13.4866
0.076163	20.9224	0.067085	17.6765	0.056527	13.6288
0.076326	21.0916	0.067253	17.8362	0.056689	13.7739
0.076495	21.264	0.067426	17.999	0.056856	13.9217
0.076669	21.4397	0.067604	18.165	0.057028	14.0727
0.076848	21.6186	0.067787	18.3341	0.057205	14.2267
0.077032	21.8009	0.067976	18.5065	0.057387	14.3839
0.077221	21.9867	0.06817	18.6825	0.057575	14.5443
0.077417	22.1761	0.06837	18.8619	0.057768	14.7081
0.077617	22.3693	0.068576	19.0449	0.057967	14.8753
0.077824	22.5662	0.068788	19.2318	0.058171	15.046
0.078037	22.767	0.069006	19.4224	0.058382	15.2204

0.078256	22.9719	0.06923	19.6168	0.058599	15.3984
0.078482	23.1808	0.069462	19.8157	0.058823	15.5804
0.078715	23.3941	0.0697	20.0185	0.059053	15.7663
0.078954	23.6117	0.069945	20.2255	0.059289	15.9563
0.0792	23.8338	0.070197	20.437	0.059533	16.1504
0.079454	24.0605	0.070457	20.6531	0.059784	16.3488
0.079715	24.2918	0.070724	20.8738	0.060043	16.5517
0.079983	24.5277	0.071	21.0993	0.060309	16.7589
0.08026	24.7686	0.071283	21.3297	0.060583	16.9709
0.080545	25.0145	0.071575	21.5649	0.060865	17.1877
0.080838	25.2654	0.071875	21.8054	0.061155	17.4094
0.08114	25.5215	0.072185	22.051	0.061454	17.636
0.081451	25.7828	0.072503	22.302	0.061762	17.868
0.081771	26.0493	0.072831	22.5584	0.062078	18.1051
0.0821	26.3211	0.073169	22.8204	0.062405	18.3477
0.082439	26.5982	0.073516	23.0882	0.062741	18.5957
0.082789	26.8808	0.073874	23.3615	0.063086	18.8495
0.083148	27.1688	0.074242	23.6405	0.063442	19.109
0.083518	27.462	0.074621	23.9256	0.063809	19.3746
0.083899	27.7604	0.075012	24.2166	0.064186	19.6462
0.084291	28.0641	0.075414	24.5134	0.064574	19.9241
0.084695	28.373	0.075801	24.8025	0.064974	20.2081
0.084734	28.4081	0.075828	24.8218	0.065386	20.4986
0.085111	28.6928	0.076254	25.1307	0.06581	20.7957
0.085539	29.0113	0.076692	25.4455	0.066246	21.0994
0.08598	29.3344	0.077144	25.7661	0.066696	21.4096
0.086433	29.6615	0.077608	26.0924	0.067158	21.7265
0.0869	29.9923	0.078087	26.4242	0.067634	22.0502
0.087381	30.3266	0.07858	26.7614	0.068125	22.3809
0.087876	30.6636	0.079087	27.1036	0.068629	22.7183
0.088386	31.0046	0.079609	27.4507	0.068881	22.8905
0.08891	31.3545	0.080147	27.802	0.069149	23.0681
0.08945	31.7188	0.0807	28.1571	0.069684	23.4188
0.090007	32.1	0.08127	28.5155	0.070235	23.7759
0.090579	32.497	0.081857	28.879	0.070802	24.1394
0.091168	32.9065	0.082461	29.2527	0.071386	24.5088
0.091775	33.3236	0.083083	29.6415	0.071987	24.8839
0.0924	33.7426	0.083723	30.0475	0.072605	25.2642
0.093043	34.158	0.084382	30.4701	0.073243	25.6492
0.093705	34.5645	0.08506	30.9063	0.073898	26.0383
0.094387	34.9569	0.085759	31.3519	0.074574	26.433
0.095089	35.3303	0.086478	31.8013	0.075269	26.837
0.095811	35.68	0.087219	32.2487	0.075985	27.2551
0.096555	36.0019	0.087981	32.688	0.076721	27.69
0.097321	36.2925	0.088766	33.1135	0.07748	28.1426

0.098109	36.5498	0.089574	33.5193	0.078261	28.6118
0.098921	36.7732	0.090406	33.8998	0.079065	29.0946
0.099757	36.963	0.091262	34.2503	0.079893	29.5866
		0.092144	34.5667	0.080745	30.0822
		0.093052	34.8465	0.081623	30.5747
		0.093987	35.0887	0.082526	31.0573
		0.094949	35.2941	0.083457	31.5228
		0.095939	35.4647	0.084249	31.8939
		0.096606	35.5566	0.084414	31.97
		0.096959	35.6046	0.0854	32.3795
		0.098009	35.7139	0.086415	32.7526
		0.09909	35.7978	0.08746	33.0852
				0.088536	33.3749
				0.089643	33.6217
				0.090783	33.8272
				0.091957	33.9943
				0.093166	34.1267
				0.09441	34.2288
				0.095691	34.3043
				0.09701	34.3572
				0.098368	34.3912
				0.099556	34.4085
				0.099766	34.4114

x/c = 0.8418		x/c = 0.8973		x/c = 0.9528	
Probe Point	Streamwise Velocity	Probe Point	Streamwise Velocity	Probe Point	Streamwise Velocity
0.04151	6.05369	0.028802	2.47165	0.014866	0.082725
0.041542	6.08879	0.028827	2.48862	0.014897	0.09396
0.041602	6.1549	0.028886	2.53085	0.014957	0.116138
0.041664	6.22167	0.028948	2.57391	0.015019	0.138909
0.041728	6.28985	0.029011	2.618	0.015083	0.162368
0.041793	6.3591	0.029077	2.66331	0.015148	0.186429
0.041861	6.42973	0.029144	2.70938	0.015216	0.211125
0.04193	6.50148	0.029213	2.75675	0.015285	0.236463
0.042002	6.57461	0.029284	2.80516	0.015357	0.262609
0.042075	6.64895	0.029358	2.85489	0.01543	0.289453
0.042151	6.72482	0.029433	2.90581	0.015506	0.31699
0.042229	6.8021	0.029511	2.95768	0.015584	0.345358
0.04231	6.88075	0.029591	3.01123	0.015664	0.374579
0.042392	6.96095	0.029674	3.06599	0.015747	0.4046
0.042478	7.04277	0.029758	3.12206	0.015832	0.435549
0.042565	7.12619	0.029846	3.17967	0.01592	0.467294

0.042656	7.21119	0.029936	3.23867	0.01601	0.500101
0.042748	7.29789	0.030028	3.29929	0.016103	0.533723
0.042844	7.3866	0.030124	3.36148	0.016199	0.568465
0.042943	7.4767	0.030222	3.42536	0.016297	0.604289
0.043044	7.56895	0.030323	3.4908	0.016399	0.641061
0.043149	7.66312	0.030427	3.55812	0.016503	0.679149
0.043256	7.75914	0.030519	3.61909	0.01661	0.718286
0.043367	7.8573	0.030534	3.62889	0.016721	0.758565
0.043481	7.95751	0.030644	3.69986	0.016796	0.786724
0.043598	8.05976	0.030758	3.77284	0.016835	0.801169
0.043719	8.16436	0.030875	3.84782	0.016953	0.844154
0.043843	8.27116	0.030995	3.92483	0.017073	0.888469
0.043971	8.38021	0.031119	4.00397	0.017198	0.934218
0.044103	8.49186	0.031247	4.08538	0.017326	0.981422
0.044239	8.60572	0.031378	4.16905	0.017457	1.03019
0.044378	8.72211	0.031513	4.25516	0.017593	1.08057
0.044522	8.84154	0.031652	4.34364	0.017733	1.13252
0.04467	8.96316	0.031796	4.43465	0.017877	1.18626
0.044823	9.08783	0.031943	4.52819	0.018025	1.24177
0.04498	9.21508	0.032095	4.62451	0.018177	1.29919
0.045141	9.34536	0.032252	4.7236	0.018334	1.35851
0.045308	9.47871	0.032413	4.82555	0.018496	1.41985
0.045424	9.57355	0.032578	4.93044	0.018662	1.48329
0.045479	9.61758	0.032749	5.03837	0.018834	1.54895
0.045655	9.75707	0.032925	5.1495	0.01901	1.61687
0.045837	9.89988	0.033106	5.26389	0.019192	1.68715
0.046024	10.0459	0.033292	5.38154	0.019379	1.75994
0.046216	10.1953	0.033484	5.50282	0.019571	1.83533
0.046414	10.3484	0.033681	5.62752	0.019769	1.91338
0.046618	10.5049	0.033884	5.75592	0.019973	1.99427
0.046828	10.6651	0.034094	5.88816	0.020183	2.07798
0.047044	10.8293	0.034309	6.02424	0.0204	2.16482
0.047267	10.9972	0.034531	6.16443	0.020622	2.25485
0.047496	11.1691	0.034759	6.30872	0.020852	2.34821
0.047732	11.3451	0.034994	6.45728	0.021088	2.44497
0.047975	11.5254	0.035237	6.61026	0.021331	2.54533
0.048225	11.7099	0.035486	6.7678	0.021581	2.64944
0.048482	11.899	0.035742	6.92995	0.021839	2.75759
0.048747	12.0925	0.036007	7.09679	0.022104	2.86964
0.04902	12.2908	0.036278	7.26857	0.022377	2.98599
0.049301	12.4938	0.036559	7.4455	0.022658	3.10683
0.049591	12.7018	0.036847	7.6276	0.022947	3.23212
0.049888	12.9148	0.037144	7.81501	0.023245	3.36242
0.050195	13.1329	0.037449	8.00794	0.023552	3.49757
0.05051	13.3564	0.037764	8.20657	0.023586	3.51522

0.050835	13.5853	0.038088	8.41089	0.023868	3.64051
0.05117	13.8198	0.038421	8.62112	0.024193	3.78649
0.051514	14.06	0.038693	8.79628	0.024528	3.938
0.051869	14.3062	0.038765	8.84141	0.024873	4.09544
0.052234	14.5583	0.039118	9.06409	0.025228	4.25903
0.05261	14.8166	0.039482	9.29319	0.025593	4.42891
0.052997	15.0812	0.039857	9.52883	0.025969	4.60532
0.053395	15.3521	0.040243	9.7712	0.026357	4.78859
0.053806	15.6298	0.04064	10.0204	0.026756	4.97897
0.054228	15.9141	0.041049	10.2768	0.027166	5.1767
0.054663	16.2055	0.04147	10.5403	0.027589	5.38202
0.055111	16.5039	0.041903	10.8111	0.028024	5.59518
0.055571	16.8092	0.042349	11.0895	0.028473	5.81663
0.056046	17.1222	0.042809	11.3755	0.028934	6.04631
0.056534	17.4424	0.043282	11.6695	0.029409	6.2849
0.056893	17.682	0.043769	11.9716	0.029898	6.53231
0.057037	17.7758	0.04427	12.2817	0.030402	6.78899
0.057555	18.1114	0.044787	12.6005	0.030457	6.82182
0.058088	18.4549	0.045318	12.9277	0.03092	7.05987
0.058637	18.8062	0.045865	13.2636	0.031454	7.33612
0.059202	19.1655	0.046429	13.6082	0.032004	7.6226
0.059784	19.5326	0.046882	13.8906	0.032569	7.91937
0.060383	19.9078	0.047009	13.9678	0.033152	8.22681
0.060999	20.2908	0.047606	14.331	0.033752	8.54519
0.061634	20.6818	0.048221	14.7034	0.03437	8.87471
0.062288	21.0803	0.048854	15.0851	0.035005	9.21557
0.062961	21.4865	0.049505	15.4765	0.03566	9.56825
0.063654	21.8999	0.050176	15.8776	0.036334	9.93262
0.064367	22.32	0.050867	16.2883	0.037028	10.3093
0.065101	22.7467	0.051578	16.7089	0.037417	10.527
0.065857	23.1791	0.052311	17.1393	0.037743	10.7045
0.066636	23.6168	0.053065	17.5795	0.038479	11.1061
0.067437	24.0614	0.053841	18.0293	0.039236	11.5203
0.068262	24.5165	0.05464	18.4886	0.040016	11.9473
0.068353	24.5745	0.055094	18.7548	0.040819	12.3872
0.069112	24.9937	0.055463	18.9651	0.041646	12.8403
0.069986	25.4804	0.05631	19.4432	0.042497	13.3064
0.070887	25.9855	0.057182	19.9297	0.043374	13.7859
0.071814	26.5089	0.05808	20.4246	0.044276	14.2784
0.072768	27.049	0.059005	20.927	0.044462	14.3872
0.073751	27.6026	0.059957	21.4361	0.045205	14.792
0.074762	28.1643	0.060937	21.9529	0.046162	15.3108
0.075804	28.727	0.061946	22.4809	0.047147	15.8426
0.076876	29.2828	0.062985	23.0239	0.048161	16.3869
0.07798	29.8223	0.063327	23.2122	0.049205	16.9438

0.079117	30.3363	0.064054	23.5941	0.05028	17.5124
0.079795	30.6202	0.065155	24.1754	0.051387	18.0925
0.080287	30.8215	0.066289	24.7775	0.051596	18.2098
0.081492	31.2569	0.067457	25.3994	0.052527	18.6925
0.082732	31.6435	0.068659	26.0386	0.0537	19.293
0.084009	31.9784	0.069896	26.6895	0.054909	19.9023
0.085324	32.2607	0.07117	27.3445	0.056153	20.5207
0.086678	32.4926	0.071581	27.5573	0.057434	21.1514
0.088071	32.6781	0.072482	28.0026	0.058753	21.7999
0.089506	32.8224	0.073832	28.6314	0.058826	21.8465
0.090984	32.9315	0.075223	29.2276	0.060111	22.4806
0.091225	32.9453	0.076655	29.7783	0.061509	23.1749
0.092504	33.012	0.078128	30.2723	0.062949	23.8936
0.09407	33.0676	0.079646	30.7023	0.064431	24.6351
0.095682	33.1053	0.079858	30.7563	0.065958	25.3945
0.097342	33.1311	0.081209	31.0687	0.066166	25.5084
0.099051	33.1505	0.082818	31.3632	0.067529	26.1747
		0.084474	31.5958	0.069147	26.9386
		0.086179	31.774	0.070814	27.6808
		0.087935	31.9064	0.072529	28.382
		0.08816	31.9197	0.073601	28.7792
		0.089743	32.0032	0.074296	29.0311
		0.091605	32.0706	0.076115	29.593
		0.093521	32.1179	0.077988	30.0678
		0.095495	32.1523	0.079916	30.4527
		0.096486	32.1661	0.081154	30.6431
		0.097526	32.1801	0.081902	30.7559
		0.099618	32.2044	0.083946	30.9789
				0.086052	31.1402
				0.08822	31.2539
				0.088818	31.2761
				0.090452	31.3337
				0.09275	31.3888
				0.095117	31.4289
				0.096609	31.4488
				0.097554	31.4611

Final k-ω Results for Transverse Velocity Profiles

x/c = 0.6753		x/c = 0.7308		x/c = 0.7863	
Probe Point	Transverse Velocity	Probe Point	Transverse Velocity	Probe Point	Transverse Velocity
0.072613	-2.45874	0.0634	-2.25662	0.053034	-1.8372

0.072616	-2.45928	0.063453	-2.26815	0.053076	-1.84575
0.072676	-2.47264	0.063513	-2.28138	0.053136	-1.8579
0.072736	-2.486	0.063576	-2.29467	0.053198	-1.87021
0.072799	-2.49943	0.06364	-2.30803	0.053262	-1.88249
0.072864	-2.5129	0.063706	-2.32142	0.053328	-1.89494
0.07293	-2.52643	0.063774	-2.33489	0.053395	-1.90738
0.072999	-2.54003	0.063844	-2.34844	0.053451	-1.91744
0.073069	-2.55367	0.063917	-2.36206	0.053465	-1.91998
0.073142	-2.56737	0.063991	-2.37574	0.053537	-1.93263
0.073217	-2.58115	0.064067	-2.3895	0.053611	-1.94537
0.073294	-2.59501	0.064146	-2.40334	0.053687	-1.9582
0.073373	-2.60891	0.064227	-2.41728	0.053766	-1.97111
0.073454	-2.62292	0.064311	-2.43128	0.053846	-1.98412
0.073538	-2.63697	0.064397	-2.44538	0.053929	-1.99723
0.073625	-2.65113	0.064485	-2.45959	0.054015	-2.01042
0.073714	-2.66539	0.064576	-2.47385	0.054103	-2.02372
0.073805	-2.6797	0.06467	-2.48825	0.054193	-2.03713
0.073899	-2.69412	0.064767	-2.50272	0.054287	-2.05066
0.073997	-2.70864	0.064866	-2.5173	0.054383	-2.06428
0.074097	-2.72326	0.064969	-2.53201	0.054482	-2.07802
0.074199	-2.73798	0.065074	-2.54681	0.054583	-2.09187
0.074305	-2.7528	0.065183	-2.56173	0.054688	-2.10585
0.074414	-2.76774	0.065294	-2.57676	0.054796	-2.11995
0.074527	-2.78279	0.065409	-2.59192	0.054907	-2.13416
0.074642	-2.79795	0.065528	-2.6072	0.055022	-2.14851
0.074761	-2.81322	0.06565	-2.62259	0.05514	-2.16298
0.074884	-2.82863	0.065775	-2.63815	0.055261	-2.17759
0.07501	-2.84418	0.065904	-2.65381	0.055386	-2.19234
0.07514	-2.85984	0.066037	-2.66962	0.055514	-2.20723
0.075273	-2.87562	0.066174	-2.68558	0.055646	-2.22226
0.075411	-2.89158	0.066315	-2.70168	0.055783	-2.23743
0.075553	-2.90765	0.066461	-2.71794	0.055923	-2.25275
0.075699	-2.92388	0.06661	-2.73434	0.056067	-2.26823
0.075849	-2.94026	0.066764	-2.7509	0.056216	-2.28385
0.076003	-2.9568	0.066922	-2.76762	0.056369	-2.29964
0.076163	-2.97349	0.067085	-2.78452	0.056527	-2.3156
0.076326	-2.99035	0.067253	-2.80157	0.056689	-2.33172
0.076495	-3.00736	0.067426	-2.81881	0.056856	-2.348
0.076669	-3.02456	0.067604	-2.83622	0.057028	-2.36445
0.076848	-3.04194	0.067787	-2.85381	0.057205	-2.38109
0.077032	-3.05948	0.067976	-2.87156	0.057387	-2.3979
0.077221	-3.07719	0.06817	-2.88953	0.057575	-2.41489
0.077417	-3.09509	0.06837	-2.90768	0.057768	-2.43208
0.077617	-3.11318	0.068576	-2.92602	0.057967	-2.44944
0.077824	-3.13145	0.068788	-2.94457	0.058171	-2.467

0.078037	-3.14991	0.069006	-2.96333	0.058382	-2.48476
0.078256	-3.16857	0.06923	-2.98227	0.058599	-2.5027
0.078482	-3.18742	0.069462	-3.00145	0.058823	-2.52085
0.078715	-3.20646	0.0697	-3.02082	0.059053	-2.53921
0.078954	-3.2257	0.069945	-3.04042	0.059289	-2.55779
0.0792	-3.24513	0.070197	-3.06024	0.059533	-2.57658
0.079454	-3.26477	0.070457	-3.08027	0.059784	-2.59557
0.079715	-3.28459	0.070724	-3.10052	0.060043	-2.6148
0.079983	-3.30459	0.071	-3.121	0.060309	-2.63423
0.08026	-3.32478	0.071283	-3.14169	0.060583	-2.65388
0.080545	-3.34515	0.071575	-3.1626	0.060865	-2.67376
0.080838	-3.3657	0.071875	-3.18374	0.061155	-2.69387
0.08114	-3.38642	0.072185	-3.20509	0.061454	-2.7142
0.081451	-3.4073	0.072503	-3.22666	0.061762	-2.73478
0.081771	-3.42832	0.072831	-3.24843	0.062078	-2.75557
0.0821	-3.44948	0.073169	-3.27039	0.062405	-2.77661
0.082439	-3.47075	0.073516	-3.29255	0.062741	-2.79786
0.082789	-3.49212	0.073874	-3.31491	0.063086	-2.81934
0.083148	-3.51359	0.074242	-3.33743	0.063442	-2.84104
0.083518	-3.53512	0.074621	-3.36012	0.063809	-2.86296
0.083899	-3.55666	0.075012	-3.38296	0.064186	-2.88508
0.084291	-3.5782	0.075414	-3.4059	0.064574	-2.90744
0.084695	-3.59972	0.075801	-3.42749	0.064974	-2.92998
0.084734	-3.60137	0.075828	-3.42896	0.065386	-2.95271
0.085111	-3.62079	0.076254	-3.45206	0.06581	-2.9756
0.085539	-3.64205	0.076692	-3.47519	0.066246	-2.99866
0.08598	-3.66314	0.077144	-3.49833	0.066696	-3.02186
0.086433	-3.684	0.077608	-3.52144	0.067158	-3.04516
0.0869	-3.70454	0.078087	-3.54444	0.067634	-3.06858
0.087381	-3.72468	0.07858	-3.56732	0.068125	-3.0921
0.087876	-3.74438	0.079087	-3.59	0.068629	-3.11564
0.088386	-3.76379	0.079609	-3.6124	0.068881	-3.12729
0.08891	-3.78339	0.080147	-3.63446	0.069149	-3.13943
0.08945	-3.80368	0.0807	-3.65607	0.069684	-3.1629
0.090007	-3.82487	0.08127	-3.6772	0.070235	-3.18628
0.090579	-3.84684	0.081857	-3.69806	0.070802	-3.20951
0.091168	-3.86917	0.082461	-3.71916	0.071386	-3.23253
0.091775	-3.89133	0.083083	-3.74103	0.071987	-3.25523
0.0924	-3.91271	0.083723	-3.76385	0.072605	-3.27756
0.093043	-3.93276	0.084382	-3.78753	0.073243	-3.29942
0.093705	-3.95097	0.08506	-3.81164	0.073898	-3.3207
0.094387	-3.96682	0.085759	-3.83567	0.074574	-3.34161
0.095089	-3.97989	0.086478	-3.859	0.075269	-3.36257
0.095811	-3.98978	0.087219	-3.88099	0.075985	-3.38408
0.096555	-3.99614	0.087981	-3.90103	0.076721	-3.4064

0.097321	-3.99875	0.088766	-3.91853	0.07748	-3.42955
0.098109	-3.99747	0.089574	-3.93295	0.078261	-3.45334
0.098921	-3.99233	0.090406	-3.94381	0.079065	-3.47733
0.099757	-3.98346	0.091262	-3.95069	0.079893	-3.50102
		0.092144	-3.9533	0.080745	-3.5238
		0.093052	-3.95152	0.081623	-3.54497
		0.093987	-3.94533	0.082526	-3.56389
		0.094949	-3.93488	0.083457	-3.5799
		0.095939	-3.92043	0.084249	-3.59015
		0.096606	-3.90778	0.084414	-3.59229
		0.096959	-3.90147	0.0854	-3.60057
		0.098009	-3.87992	0.086415	-3.60438
		0.09909	-3.85533	0.08746	-3.60346
				0.088536	-3.59775
				0.089643	-3.58734
				0.090783	-3.57248
				0.091957	-3.5535
				0.093166	-3.53079
				0.09441	-3.50476
				0.095691	-3.47575
				0.09701	-3.44415
				0.098368	-3.41036
				0.099556	-3.37939
				0.099766	-3.37402

x/c = 0.8418		x/c = 0.8973		x/c = 0.9528	
Probe Point	Transverse Velocity	Probe Point	Transverse Velocity	Probe Point	Transverse Velocity
0.04151	-1.20445	0.028802	-0.50379	0.014866	0.027817
0.041542	-1.2098	0.028827	-0.50643	0.014897	0.025869
0.041602	-1.21983	0.028886	-0.51305	0.014957	0.022032
0.041664	-1.22987	0.028948	-0.51973	0.015019	0.018117
0.041728	-1.24004	0.029011	-0.52653	0.015083	0.014105
0.041793	-1.25028	0.029077	-0.53346	0.015148	0.010012
0.041861	-1.26064	0.029144	-0.54045	0.015216	0.005834
0.04193	-1.27109	0.029213	-0.54758	0.015285	0.001574
0.042002	-1.28164	0.029284	-0.5548	0.015357	-0.0028
0.042075	-1.29228	0.029358	-0.56216	0.01543	-0.00726
0.042151	-1.30305	0.029433	-0.56965	0.015506	-0.0118
0.042229	-1.31394	0.029511	-0.5772	0.015584	-0.01646
0.04231	-1.32493	0.029591	-0.58495	0.015664	-0.02121
0.042392	-1.33603	0.029674	-0.5928	0.015747	-0.02607
0.042478	-1.34727	0.029758	-0.60077	0.015832	-0.03105

0.042565	-1.35863	0.029846	-0.6089	0.01592	-0.03612
0.042656	-1.37011	0.029936	-0.61715	0.01601	-0.04134
0.042748	-1.3817	0.030028	-0.62555	0.016103	-0.04665
0.042844	-1.39347	0.030124	-0.6341	0.016199	-0.0521
0.042943	-1.40531	0.030222	-0.6428	0.016297	-0.05768
0.043044	-1.41733	0.030323	-0.65163	0.016399	-0.06336
0.043149	-1.42949	0.030427	-0.66063	0.016503	-0.0692
0.043256	-1.44177	0.030519	-0.66877	0.01661	-0.07517
0.043367	-1.45421	0.030534	-0.67007	0.016721	-0.08127
0.043481	-1.4668	0.030644	-0.67939	0.016796	-0.08553
0.043598	-1.47952	0.030758	-0.6889	0.016835	-0.0877
0.043719	-1.49241	0.030875	-0.69857	0.016953	-0.09411
0.043843	-1.50545	0.030995	-0.70842	0.017073	-0.10067
0.043971	-1.51864	0.031119	-0.71844	0.017198	-0.1074
0.044103	-1.53203	0.031247	-0.72865	0.017326	-0.11428
0.044239	-1.54554	0.031378	-0.73905	0.017457	-0.12134
0.044378	-1.55921	0.031513	-0.74965	0.017593	-0.12858
0.044522	-1.57313	0.031652	-0.76044	0.017733	-0.13598
0.04467	-1.58715	0.031796	-0.77144	0.017877	-0.14357
0.044823	-1.60137	0.031943	-0.78262	0.018025	-0.15135
0.04498	-1.61574	0.032095	-0.79402	0.018177	-0.15932
0.045141	-1.63031	0.032252	-0.80563	0.018334	-0.16749
0.045308	-1.64509	0.032413	-0.81746	0.018496	-0.17586
0.045424	-1.65552	0.032578	-0.82951	0.018662	-0.18445
0.045479	-1.66035	0.032749	-0.84178	0.018834	-0.19325
0.045655	-1.67548	0.032925	-0.85428	0.01901	-0.20228
0.045837	-1.69081	0.033106	-0.86702	0.019192	-0.21152
0.046024	-1.70632	0.033292	-0.87997	0.019379	-0.22101
0.046216	-1.72204	0.033484	-0.89318	0.019571	-0.23074
0.046414	-1.73797	0.033681	-0.90661	0.019769	-0.24071
0.046618	-1.75409	0.033884	-0.92029	0.019973	-0.25094
0.046828	-1.77041	0.034094	-0.93422	0.020183	-0.26141
0.047044	-1.78696	0.034309	-0.9484	0.0204	-0.27217
0.047267	-1.8037	0.034531	-0.96284	0.020622	-0.28319
0.047496	-1.82066	0.034759	-0.97752	0.020852	-0.2945
0.047732	-1.83783	0.034994	-0.99246	0.021088	-0.30608
0.047975	-1.85522	0.035237	-1.00766	0.021331	-0.31796
0.048225	-1.87282	0.035486	-1.02313	0.021581	-0.33014
0.048482	-1.89065	0.035742	-1.03887	0.021839	-0.34263
0.048747	-1.9087	0.036007	-1.05485	0.022104	-0.35542
0.04902	-1.92697	0.036278	-1.07111	0.022377	-0.36852
0.049301	-1.94547	0.036559	-1.08766	0.022658	-0.38197
0.049591	-1.96421	0.036847	-1.10447	0.022947	-0.39572
0.049888	-1.98317	0.037144	-1.12154	0.023245	-0.40983
0.050195	-2.00235	0.037449	-1.13889	0.023552	-0.42426

0.05051	-2.02177	0.037764	-1.15652	0.023586	-0.42611
0.050835	-2.04143	0.038088	-1.17441	0.023868	-0.43928
0.05117	-2.06131	0.038421	-1.19256	0.024193	-0.45441
0.051514	-2.08143	0.038693	-1.2075	0.024528	-0.46987
0.051869	-2.10178	0.038765	-1.21134	0.024873	-0.48569
0.052234	-2.12236	0.039118	-1.23003	0.025228	-0.50186
0.05261	-2.14315	0.039482	-1.24898	0.025593	-0.51838
0.052997	-2.16418	0.039857	-1.26819	0.025969	-0.53525
0.053395	-2.18541	0.040243	-1.28766	0.026357	-0.55248
0.053806	-2.20689	0.04064	-1.3074	0.026756	-0.57006
0.054228	-2.22856	0.041049	-1.32738	0.027166	-0.58801
0.054663	-2.25046	0.04147	-1.34761	0.027589	-0.60629
0.055111	-2.27256	0.041903	-1.36808	0.028024	-0.62491
0.055571	-2.29481	0.042349	-1.3888	0.028473	-0.64388
0.056046	-2.31729	0.042809	-1.40974	0.028934	-0.66317
0.056534	-2.33992	0.043282	-1.4309	0.029409	-0.68281
0.056893	-2.35662	0.043769	-1.45228	0.029898	-0.70274
0.057037	-2.36314	0.04427	-1.47386	0.030402	-0.72297
0.057555	-2.38607	0.044787	-1.49566	0.030457	-0.72554
0.058088	-2.40911	0.045318	-1.51763	0.03092	-0.74389
0.058637	-2.43225	0.045865	-1.53976	0.031454	-0.76472
0.059202	-2.45545	0.046429	-1.56204	0.032004	-0.78581
0.059784	-2.47867	0.046882	-1.58006	0.032569	-0.80716
0.060383	-2.50191	0.047009	-1.58495	0.033152	-0.82873
0.060999	-2.52511	0.047606	-1.60755	0.033752	-0.85052
0.061634	-2.54823	0.048221	-1.63024	0.03437	-0.87251
0.062288	-2.5712	0.048854	-1.653	0.035005	-0.89467
0.062961	-2.59396	0.049505	-1.67584	0.03566	-0.91698
0.063654	-2.61645	0.050176	-1.6987	0.036334	-0.93939
0.064367	-2.63858	0.050867	-1.72157	0.037028	-0.96193
0.065101	-2.66025	0.051578	-1.7444	0.037417	-0.97463
0.065857	-2.68137	0.052311	-1.76716	0.037743	-0.98494
0.066636	-2.70189	0.053065	-1.78978	0.038479	-1.00758
0.067437	-2.72198	0.053841	-1.81225	0.039236	-1.03022
0.068262	-2.74201	0.05464	-1.83447	0.040016	-1.05284
0.068353	-2.74458	0.055094	-1.84703	0.040819	-1.07538
0.069112	-2.76271	0.055463	-1.85687	0.041646	-1.09782
0.069986	-2.78355	0.05631	-1.87845	0.042497	-1.12011
0.070887	-2.80495	0.057182	-1.89955	0.043374	-1.14221
0.071814	-2.82681	0.05808	-1.92012	0.044276	-1.16405
0.072768	-2.84886	0.059005	-1.94005	0.044462	-1.16875
0.073751	-2.8707	0.059957	-1.95928	0.045205	-1.186
0.074762	-2.89181	0.060937	-1.97791	0.046162	-1.20718
0.075804	-2.91154	0.061946	-1.99617	0.047147	-1.22794
0.076876	-2.92924	0.062985	-2.01427	0.048161	-1.24819

0.07798	-2.94417	0.063327	-2.02049	0.049205	-1.26788
0.079117	-2.95564	0.064054	-2.03286	0.05028	-1.2869
0.079795	-2.96002	0.065155	-2.05117	0.051387	-1.30517
0.080287	-2.96312	0.066289	-2.06953	0.051596	-1.30871
0.081492	-2.96592	0.067457	-2.08775	0.052527	-1.32294
0.082732	-2.96391	0.068659	-2.10553	0.0537	-1.33938
0.084009	-2.95703	0.069896	-2.1224	0.054909	-1.35482
0.085324	-2.94539	0.07117	-2.1378	0.056153	-1.36929
0.086678	-2.92926	0.071581	-2.14228	0.057434	-1.38288
0.088071	-2.90904	0.072482	-2.15139	0.058753	-1.39575
0.089506	-2.88519	0.073832	-2.1618	0.058826	-1.3967
0.090984	-2.85817	0.075223	-2.16874	0.060111	-1.40831
0.091225	-2.85301	0.076655	-2.17167	0.061509	-1.42003
0.092504	-2.82796	0.078128	-2.17019	0.062949	-1.43105
0.09407	-2.79592	0.079646	-2.16416	0.064431	-1.44113
0.095682	-2.76207	0.079858	-2.16267	0.065958	-1.44989
0.097342	-2.72678	0.081209	-2.1535	0.066166	-1.451
0.099051	-2.69035	0.082818	-2.13863	0.067529	-1.45697
		0.084474	-2.11989	0.069147	-1.46143
		0.086179	-2.09775	0.070814	-1.46285
		0.087935	-2.07271	0.072529	-1.46075
		0.08816	-2.06898	0.073601	-1.45707
		0.089743	-2.04493	0.074296	-1.4547
		0.091605	-2.01553	0.076115	-1.4446
		0.093521	-1.9846	0.077988	-1.43051
		0.095495	-1.95244	0.079916	-1.4127
		0.096486	-1.93582	0.081154	-1.39924
		0.097526	-1.9188	0.081902	-1.39127
		0.099618	-1.88457	0.083946	-1.36718
				0.086052	-1.34069
				0.08822	-1.31223
				0.088818	-1.3038
				0.090452	-1.28181
				0.09275	-1.25051
				0.095117	-1.21814
				0.096609	-1.19724
				0.097554	-1.1843

Coefficient of Pressure Data for Both Turbulence Models

Points along x axis	Pressure	
	Menter SST	Wilcox 2006 k-ω
0.999498	101275	101331
0.99849	101285	101339

0.99747	101290	101343
0.996439	101296	101347
0.995396	101300	101350
0.994341	101303	101353
0.993273	101307	101355
0.992193	101310	101358
0.991101	101313	101360
0.989996	101315	101362
0.988877	101318	101364
0.987747	101320	101366
0.986602	101323	101367
0.985443	101325	101369
0.984271	101327	101371
0.983085	101329	101372
0.981885	101331	101374
0.98067	101333	101375
0.97944	101335	101377
0.978196	101337	101378
0.976936	101339	101379
0.975661	101340	101381
0.974369	101342	101382
0.973062	101344	101383
0.971738	101345	101384
0.970398	101347	101386
0.96904	101349	101387
0.967665	101350	101388
0.966272	101352	101389
0.964862	101353	101390
0.963435	101355	101391
0.961987	101356	101393
0.960522	101358	101394
0.959037	101359	101395
0.957532	101361	101396
0.956008	101362	101397
0.954464	101363	101398
0.952898	101365	101399
0.951312	101366	101400
0.949703	101367	101401
0.948074	101369	101402
0.946422	101370	101403
0.944747	101371	101404
0.943049	101373	101405
0.941327	101374	101406
0.939582	101375	101407
0.937811	101376	101408

0.936016	101378	101409
0.934195	101379	101410
0.932348	101380	101411
0.930474	101381	101412
0.928574	101383	101413
0.926646	101384	101414
0.924689	101385	101415
0.922703	101386	101416
0.920687	101388	101417
0.918642	101389	101418
0.916566	101390	101419
0.914458	101391	101420
0.912318	101392	101421
0.910146	101394	101422
0.90794	101395	101423
0.905699	101396	101424
0.903423	101397	101425
0.901112	101398	101426
0.898764	101400	101427
0.896378	101401	101428
0.893953	101402	101429
0.89149	101403	101430
0.888986	101404	101431
0.886441	101406	101431
0.883854	101407	101432
0.881224	101408	101433
0.878549	101409	101434
0.87583	101410	101435
0.873064	101412	101436
0.870251	101413	101437
0.867389	101414	101438
0.864476	101415	101440
0.861512	101416	101441
0.858497	101418	101442
0.855427	101419	101443
0.852301	101420	101444
0.84912	101421	101445
0.845879	101422	101446
0.84258	101424	101447
0.839218	101425	101448
0.835794	101426	101449
0.832305	101428	101450
0.828748	101429	101451
0.825124	101430	101452
0.82143	101431	101453

0.817662	101433	101454
0.813821	101434	101456
0.809902	101435	101457
0.805905	101436	101458
0.801826	101438	101459
0.797664	101439	101460
0.793416	101440	101461
0.789078	101442	101462
0.784649	101443	101464
0.780125	101445	101465
0.775505	101446	101466
0.770783	101447	101467
0.765958	101449	101469
0.761026	101450	101470
0.755984	101452	101471
0.750826	101453	101472
0.745551	101454	101474
0.740153	101456	101475
0.734629	101457	101476
0.728973	101459	101478
0.723183	101460	101479
0.717253	101462	101481
0.711177	101464	101482
0.704951	101465	101484
0.698569	101467	101485
0.692024	101468	101487
0.685312	101470	101488
0.678425	101472	101490
0.671357	101473	101491
0.6641	101475	101493
0.656647	101477	101494
0.648991	101479	101496
0.641122	101481	101498
0.633032	101483	101500
0.624712	101484	101502
0.616151	101486	101503
0.607341	101488	101505
0.598269	101491	101507
0.588923	101493	101509
0.579292	101495	101512
0.569361	101497	101514
0.559119	101500	101516
0.548548	101502	101518
0.537634	101504	101521
0.52636	101507	101524

0.514708	101510	101526
0.502658	101513	101529
0.49019	101516	101532
0.477282	101520	101536
0.463912	101523	101539
0.450051	101527	101543
0.435676	101531	101547
0.420755	101536	101552
0.405257	101542	101558
0.38915	101549	101564
0.372399	101554	101570
0.354965	101560	101575
0.336804	101565	101581
0.318302	101571	101587
0.300337	101577	101593
0.283353	101583	101599
0.267293	101589	101605
0.252109	101596	101612
0.237751	101602	101618
0.224174	101608	101625
0.211336	101615	101631
0.199196	101622	101638
0.187717	101629	101645
0.176861	101636	101653
0.166596	101643	101660
0.156888	101651	101668
0.147709	101659	101676
0.13903	101666	101684
0.130823	101675	101692
0.123063	101683	101700
0.115726	101691	101708
0.10879	101700	101716
0.102233	101709	101725
0.096034	101717	101733
0.090175	101726	101742
0.084638	101735	101750
0.079404	101743	101758
0.074459	101752	101766
0.069788	101760	101773
0.065374	101768	101780
0.061206	101775	101787
0.057269	101783	101793
0.053553	101789	101799
0.050045	101795	101803
0.046734	101801	101807

0.043611	101805	101809
0.040666	101808	101810
0.037889	101810	101810
0.035272	101811	101808
0.032806	101809	101804
0.030484	101807	101798
0.028299	101801	101790
0.026244	101794	101778
0.024312	101784	101764
0.022497	101771	101746
0.020793	101755	101725
0.019195	101735	101699
0.017697	101711	101669
0.016295	101683	101635
0.014983	101650	101595
0.013758	101613	101550
0.012615	101571	101499
0.011549	101522	101443
0.010557	101469	101380
0.009636	101410	101312
0.008781	101346	101237
0.007988	101276	101157
0.007256	101201	101071
0.006579	101122	100980
0.005956	101038	100884
0.005382	100949	100784
0.004855	100857	100681
0.004373	100763	100575
0.003931	100667	100467
0.003527	100569	100358
0.003159	100471	100249
0.002824	100374	100141
0.002519	100277	100033
0.002242	100182	99928.4
0.001991	100089	99825.7
0.001764	99998.1	99726.3
0.001558	99910.7	99630.4
0.001372	99825.9	99537.7
0.001204	99745.5	99449.9
0.001053	99667.7	99365.1
0.000916	99594	99285.1
0.000793	99524.5	99209.7
0.000682	99457.9	99137.5
0.000581	99395.4	99069.9
0.000491	99335.8	99005.7

0.00041	99280.2	98945.7
0.000337	99226.7	98888.3
0.000271	99177.8	98835.7
0.000212	99129.5	98784.1
0.000158	99086.1	98737.6
0.00011	99044.2	98692.9
6.64E-05	99004.1	98650
2.72E-05	98967.2	98610.7
-8.22E-06	98932.2	98573.4
-4.02E-05	98897.7	98536.8
-6.90E-05	98866.2	98503.2
-9.50E-05	98836	98471.3
-0.00012	98807	98440.5
-0.00014	98779.5	98411.4
-0.00016	98753.4	98383.8
-0.00018	98728.7	98357.7
-0.00019	98705.1	98332.8
-0.00021	98682.7	98309.2
-0.00022	98660.5	98285.7
-0.00023	98639.6	98263.8
-0.00024	98621.1	98244.3
-0.00025	98599.5	98221.8
-0.00026	98583.5	98204.6
-0.00027	98563.4	98184
-0.00027	98548	98167.5
-0.00028	98528.9	98147.9
-0.00028	98515	98133
-0.00029	98495.8	98113.4
-0.00029	98482.9	98099.5
-0.00029	98465.2	98081.5
-0.0003	98451	98066.5
-0.0003	98435.8	98050.8
-0.0003	98419.9	98034.4
-0.0003	98406.6	98020.4
-0.0003	98389.6	98003.1
-0.0003	98377.3	97990
-0.0003	98360.3	97972.8
-0.0003	98347.3	97959.2
-0.00029	98331.7	97943.3
-0.00029	98316.8	97928
-0.00028	98302.8	97913.6
-0.00028	98287.1	97897.7
-0.00027	98272	97882.3
-0.00026	98257.6	97867.6
-0.00025	98242.4	97852.2

-0.00024	98227.3	97836.9
-0.00023	98212.2	97821.7
-0.00021	98196.9	97806.4
-0.00019	98182.2	97791.6
-0.00017	98165.8	97775.3
-0.00015	98152.1	97761.7
-0.00013	98136.7	97746.5
-9.79E-05	98122.5	97732.6
-6.62E-05	98107.3	97717.8
-3.06E-05	98094.1	97705
9.25E-06	98080.3	97691.9
5.39E-05	98067.2	97679.6
0.000104	98055.1	97668.4
0.000159	98042.3	97656.7
0.000221	98033	97648.7
0.000291	98022	97639.5
0.000367	98013.7	97633.1
0.000453	98006.5	97628.5
0.000548	97999.7	97625
0.000653	97996	97625.4
0.000769	97992.8	97627.4
0.000898	97992.8	97633.2
0.00104	97995.1	97641.8
0.001197	98001.9	97654
0.00137	98012.9	97668.4
0.00156	98031.8	97687.1
0.001769	98052.5	97707.4
0.001999	98074.7	97731.4
0.002251	98097.3	97757.6
0.002527	98121.2	97787.5
0.002829	98147.1	97819.9
0.003159	98176.1	97855.7
0.003519	98208.4	97894.8
0.003912	98242.6	97936.2
0.004338	98279.5	97980.7
0.004802	98320.3	98028.3
0.005306	98361.8	98077.5
0.005852	98406	98129.6
0.006443	98452.7	98183.3
0.007082	98500.4	98238.9
0.007773	98549.2	98295.8
0.008518	98601.2	98354.3
0.009321	98652.7	98413.5
0.010186	98704.3	98473.3
0.011116	98758.1	98533.7

0.012115	98812.1	98594.5
0.013188	98865.2	98655
0.014338	98919	98715.6
0.015571	98972.4	98775.7
0.016891	99025.7	98835.4
0.018304	99078.7	98894.4
0.019814	99130	98952.4
0.021427	99181.2	99009.9
0.02315	99232.3	99066.2
0.024989	99281.5	99121.3
0.026949	99329.4	99175.4
0.029039	99377.9	99228.5
0.031265	99425.6	99279.9
0.033635	99472.5	99330.3
0.036159	99515.7	99379.2
0.038843	99557.4	99427
0.041698	99600.4	99473.6
0.044733	99641.3	99518.8
0.047958	99680.8	99563
0.051385	99719.7	99605.8
0.055024	99759.6	99647.4
0.058888	99797.3	99687.7
0.062989	99831.3	99726.9
0.067341	99865.7	99765.2
0.071959	99900	99802.5
0.076857	99933.6	99838.7
0.082051	99966.5	99873.9
0.087558	99997.9	99908.2
0.093395	100028	99941.8
0.099582	100059	99974.5
0.106138	100088	100006
0.113084	100115	100038
0.120443	100149	100069
0.128236	100182	100099
0.136488	100205	100129
0.145227	100230	100158
0.154477	100259	100188
0.164269	100286	100217
0.174631	100314	100246
0.185596	100341	100275
0.197196	100369	100304
0.209468	100397	100333
0.222445	100425	100363
0.236169	100454	100393
0.250678	100483	100423

0.266014	100513	100454
0.282222	100544	100486
0.299346	100576	100519
0.317436	100609	100552
0.336539	100644	100588
0.356709	100681	100625
0.377427	100719	100663
0.397618	100756	100701
0.416794	100794	100740
0.435026	100825	100771
0.452385	100854	100799
0.468928	100880	100824
0.484714	100903	100847
0.499793	100924	100868
0.514209	100945	100888
0.528007	100964	100906
0.541225	100983	100924
0.553899	101000	100940
0.566061	101016	100956
0.577743	101031	100971
0.588972	101046	100985
0.599773	101060	100998
0.610171	101072	101011
0.620189	101085	101023
0.629847	101097	101035
0.639163	101108	101046
0.648157	101118	101057
0.656843	101128	101067
0.665239	101138	101077
0.673358	101146	101087
0.681214	101154	101096
0.68882	101162	101105
0.696186	101169	101114
0.703326	101176	101122
0.710249	101182	101130
0.716964	101188	101138
0.723481	101193	101145
0.72981	101198	101152
0.735957	101202	101159
0.741932	101207	101166
0.747741	101210	101172
0.75339	101214	101178
0.758887	101217	101184
0.764238	101220	101190
0.769448	101223	101196

0.774523	101226	101201
0.779468	101228	101206
0.784288	101230	101211
0.788989	101232	101216
0.793574	101234	101221
0.798048	101236	101225
0.802415	101238	101230
0.806678	101239	101234
0.810843	101241	101238
0.814911	101242	101241
0.818886	101244	101245
0.822771	101245	101249
0.82657	101246	101252
0.830286	101247	101255
0.83392	101248	101258
0.837477	101249	101261
0.840959	101250	101264
0.844367	101251	101267
0.847704	101252	101269
0.850972	101253	101272
0.854174	101253	101274
0.857312	101254	101276
0.860388	101255	101279
0.863403	101255	101281
0.86636	101256	101283
0.869258	101257	101285
0.872103	101257	101286
0.874893	101258	101288
0.877631	101258	101290
0.880319	101259	101292
0.882958	101259	101293
0.885549	101260	101295
0.888094	101260	101296
0.890592	101261	101297
0.893048	101261	101299
0.895461	101262	101300
0.897832	101262	101301
0.900162	101262	101302
0.902453	101263	101303
0.904706	101263	101304
0.906921	101263	101305
0.909099	101264	101306
0.911243	101264	101307
0.913352	101264	101308
0.915427	101265	101309

0.917469	101265	101310
0.919479	101265	101311
0.921458	101266	101312
0.923405	101266	101312
0.925324	101266	101313
0.927213	101266	101314
0.929074	101267	101314
0.930907	101267	101315
0.932713	101267	101316
0.934493	101267	101316
0.936246	101268	101317
0.937975	101268	101317
0.939678	101268	101318
0.941358	101268	101319
0.943014	101268	101319
0.944647	101269	101320
0.946257	101269	101320
0.947845	101269	101321
0.949412	101269	101321
0.950958	101269	101321
0.952483	101269	101322
0.953987	101270	101322
0.955473	101270	101323
0.956938	101270	101323
0.958384	101270	101323
0.959812	101270	101324
0.961222	101270	101324
0.962614	101270	101325
0.963987	101271	101325
0.965345	101271	101325
0.966685	101271	101326
0.968009	101271	101326
0.969317	101271	101326
0.970609	101271	101327
0.971885	101272	101327
0.973147	101272	101327
0.974392	101272	101327
0.975625	101272	101328
0.976842	101272	101328
0.978045	101273	101328
0.979234	101273	101329
0.98041	101273	101329
0.981573	101273	101329
0.982722	101274	101329
0.983859	101274	101330

0.984983	101274	101330
0.986095	101275	101330
0.987194	101275	101330
0.988282	101275	101331
0.989358	101276	101331
0.990422	101276	101331
0.991476	101276	101331
0.992518	101277	101332
0.993549	101277	101332
0.994569	101277	101332
0.99558	101278	101333
0.996579	101278	101333
0.997568	101279	101333
0.998548	101279	101334
0.999518	101278	101333

Final LRR-IP Results for Streamwise Velocity Profiles

x/c = 0.6753		x/c = 0.7308		x/c = 0.7863	
Probe Point	Streamwise Velocity	Probe Point	Streamwise Velocity	Probe Point	Streamwise Velocity
0.070614	1.11837	0.061402	0.071928	0.051038	0.799788
0.070618	1.33626	0.061406	0.109813	0.051042	0.889766
0.070622	1.54867	0.06141	0.314891	0.051046	0.978552
0.070626	1.74854	0.061414	0.497368	0.051049	1.05836
0.07063	1.93183	0.061418	0.666443	0.051053	1.13725
0.070634	2.10953	0.061422	0.825395	0.051057	1.20509
0.070638	2.27123	0.061426	0.968966	0.051061	1.26784
0.070642	2.42154	0.06143	1.10209	0.051066	1.32698
0.070647	2.5593	0.061435	1.22471	0.05107	1.37931
0.070651	2.68649	0.061439	1.34058	0.051074	1.42656
0.070656	2.80356	0.061444	1.44422	0.051079	1.47152
0.070661	2.9133	0.061449	1.54379	0.051083	1.51059
0.070665	3.0165	0.061453	1.63675	0.051088	1.54732
0.070671	3.11077	0.061459	1.72053	0.051093	1.58243
0.070676	3.2015	0.061464	1.79928	0.051099	1.61423
0.070681	3.28557	0.061469	1.87298	0.051104	1.64337
0.070687	3.36595	0.061475	1.94338	0.05111	1.67091
0.070693	3.44185	0.061481	2.00913	0.051116	1.69708
0.0707	3.51378	0.061487	2.07132	0.051122	1.72088
0.070706	3.5807	0.061494	2.13114	0.051129	1.74301
0.070713	3.64474	0.061501	2.18588	0.051135	1.76394
0.07072	3.70539	0.061508	2.23848	0.051143	1.78389

0.070728	3.7622	0.061516	2.28874	0.05115	1.80197
0.070736	3.81703	0.061524	2.33652	0.051158	1.81912
0.070745	3.8698	0.061532	2.3817	0.051167	1.83534
0.070754	3.91949	0.061541	2.42459	0.051175	1.85022
0.070763	3.96763	0.06155	2.46516	0.051185	1.86438
0.070773	4.01192	0.06156	2.50457	0.051195	1.87764
0.070784	4.05517	0.061571	2.5419	0.051205	1.88998
0.070795	4.09657	0.061582	2.57713	0.051216	1.90162
0.070806	4.13655	0.061593	2.61087	0.051227	1.91248
0.070819	4.1744	0.061606	2.6435	0.05124	1.92269
0.070832	4.21061	0.061619	2.67472	0.051253	1.93231
0.070846	4.24509	0.061632	2.70428	0.051266	1.94121
0.07086	4.27868	0.061647	2.7329	0.051281	1.94952
0.070876	4.31126	0.061662	2.76027	0.051296	1.95751
0.070892	4.3417	0.061679	2.78665	0.051312	1.96468
0.07091	4.37161	0.061696	2.81193	0.051329	1.97156
0.070928	4.40042	0.061714	2.83631	0.051347	1.97793
0.070947	4.4279	0.061733	2.85972	0.051366	1.98377
0.070968	4.4547	0.061754	2.88229	0.051387	1.98932
0.07099	4.4804	0.061776	2.90409	0.051408	1.99445
0.071013	4.50513	0.061798	2.9251	0.051431	1.9992
0.071037	4.52919	0.061823	2.94538	0.051455	2.00358
0.071063	4.55251	0.061848	2.96498	0.051481	2.00766
0.071091	4.57503	0.061876	2.98414	0.051508	2.01139
0.07112	4.59693	0.061904	3.00242	0.051536	2.01482
0.07115	4.61791	0.061935	3.02021	0.051567	2.01797
0.071183	4.63859	0.061967	3.03749	0.051599	2.02091
0.071217	4.65845	0.062002	3.05429	0.051633	2.02353
0.071254	4.67777	0.062038	3.07053	0.051669	2.02581
0.071293	4.69677	0.062077	3.08631	0.051707	2.02799
0.071334	4.71515	0.062117	3.10157	0.051748	2.02992
0.071377	4.73303	0.062161	3.11646	0.051791	2.03157
0.071424	4.75067	0.062207	3.13093	0.051836	2.03306
0.071472	4.76783	0.062255	3.14508	0.051885	2.0344
0.071524	4.78472	0.062306	3.15877	0.051936	2.03553
0.071579	4.80139	0.062361	3.17219	0.05199	2.0365
0.071637	4.81781	0.062419	3.18532	0.052048	2.03733
0.071699	4.83404	0.06248	3.19809	0.052108	2.03806
0.071764	4.85032	0.062545	3.2107	0.052173	2.03872
0.071834	4.86653	0.062614	3.22308	0.052241	2.0393
0.071907	4.88266	0.062687	3.23526	0.052314	2.03982
0.071985	4.89911	0.062764	3.24738	0.052391	2.04033
0.072067	4.91572	0.062846	3.25937	0.052472	2.04085
0.072155	4.93261	0.062933	3.27126	0.052558	2.04141
0.072247	4.95	0.063025	3.28322	0.05265	2.04206

0.072346	4.96795	0.063122	3.29524	0.052747	2.04286
0.07245	4.98655	0.063226	3.30741	0.052849	2.04386
0.07256	5.00601	0.063335	3.31981	0.052958	2.04505
0.072677	5.02643	0.063451	3.3325	0.053074	2.04657
0.072801	5.04791	0.063574	3.34556	0.053196	2.04842
0.072932	5.07085	0.063705	3.35914	0.053326	2.05072
0.073071	5.09527	0.063843	3.37332	0.053374	2.05806
0.073219	5.1214	0.06399	3.38824	0.053463	2.06101
0.073375	5.14957	0.064145	3.40403	0.053609	2.06482
0.073541	5.17991	0.06431	3.42081	0.053763	2.06927
0.073716	5.21288	0.064484	3.43881	0.053927	2.07446
0.073902	5.2487	0.064669	3.45821	0.0541	2.0805
0.0741	5.28763	0.064865	3.47922	0.054284	2.08758
0.074309	5.33014	0.065073	3.50198	0.054478	2.09578
0.07453	5.37655	0.065293	3.52681	0.054685	2.10535
0.074765	5.42728	0.065526	3.554	0.054904	2.11641
0.075014	5.48291	0.065774	3.58384	0.055135	2.12921
0.075278	5.54386	0.066036	3.6167	0.055381	2.14395
0.075558	5.61065	0.066314	3.65286	0.055642	2.1609
0.075854	5.68394	0.066608	3.69276	0.055918	2.18033
0.076168	5.7644	0.066921	3.73689	0.056211	2.20254
0.076501	5.85266	0.067252	3.78571	0.056521	2.22786
0.076855	5.94957	0.067603	3.83967	0.05685	2.25667
0.077229	6.05592	0.067974	3.8995	0.057199	2.2894
0.077625	6.17264	0.068368	3.96568	0.057568	2.32646
0.078046	6.30083	0.068786	4.03901	0.05796	2.36835
0.078491	6.44136	0.069229	4.12028	0.058375	2.41562
0.078963	6.59555	0.069698	4.21017	0.058815	2.46884
0.079464	6.76477	0.070195	4.30978	0.059281	2.52871
0.079994	6.95034	0.070723	4.41996	0.059776	2.59591
0.080557	7.15388	0.071282	4.54192	0.0603	2.67127
0.081153	7.37701	0.071874	4.67683	0.060855	2.75563
0.081784	7.62159	0.072501	4.8259	0.061444	2.84999
0.082454	7.88973	0.073167	4.99079	0.062068	2.95537
0.083163	8.18365	0.073872	5.17284	0.062729	3.07295
0.083916	8.50565	0.07462	5.37396	0.06343	3.20401
0.084713	8.85844	0.075412	5.59598	0.064173	3.35001
0.085558	9.24494	0.075729	5.84102	0.064961	3.5125
0.085571	9.27751	0.076252	5.95581	0.065795	3.69318
0.086453	9.70151	0.077142	6.1248	0.06668	3.89401
0.087402	10.167	0.078085	6.42483	0.067618	4.11715
0.088408	10.6767	0.079084	6.75559	0.068612	4.36494
0.089474	11.2345	0.080144	7.12016	0.069666	4.63994
0.090603	11.8448	0.081267	7.52192	0.070783	4.94509
0.091801	12.512	0.082458	7.96454	0.071966	5.28353

0.09307	13.2411	0.083719	8.45197	0.073221	5.65887
0.094416	14.0365	0.085057	8.98867	0.074551	6.0749
0.095842	14.9032	0.086474	9.57948	0.075961	6.53604
0.097353	15.8454	0.087977	10.2295	0.077455	7.04703
0.098955	16.8668	0.089569	10.9441	0.079039	7.61311
		0.091257	11.7293	0.080718	8.24013
		0.093047	12.5905	0.082497	8.93429
		0.094943	13.5338	0.08322	9.2471
		0.096953	14.564	0.084383	9.73587
		0.099084	15.6854	0.086382	10.5884
				0.088501	11.5292
				0.090746	12.5651
				0.093126	13.7021
				0.095649	14.9441
				0.098324	16.293

x/c = 0.8418		x/c = 0.8973		x/c = 0.9528	
Probe Point	Streamwise Velocity	Probe Point	Streamwise Velocity	Probe Point	Streamwise Velocity
0.039513	-0.00778	0.026805	0.094019	0.012901	-0.16489
0.039517	0.065111	0.026809	0.118683	0.012906	-0.16692
0.039521	0.136106	0.026813	0.142876	0.01291	-0.18008
0.039524	0.201287	0.026816	0.165368	0.012915	-0.1917
0.039528	0.264614	0.02682	0.185696	0.01292	-0.20343
0.039532	0.320286	0.026824	0.204839	0.012925	-0.21418
0.039536	0.371216	0.026828	0.222328	0.01293	-0.2245
0.03954	0.417614	0.026832	0.237965	0.012935	-0.23441
0.039545	0.462806	0.026837	0.252385	0.012941	-0.24377
0.039549	0.501533	0.026841	0.265419	0.012947	-0.2528
0.039554	0.537228	0.026846	0.277364	0.012953	-0.2614
0.039558	0.570754	0.02685	0.288036	0.01296	-0.26978
0.039563	0.603062	0.026855	0.297951	0.012966	-0.27794
0.039568	0.631561	0.02686	0.307072	0.012974	-0.28579
0.039573	0.658258	0.026865	0.315333	0.012981	-0.29344
0.039579	0.684371	0.026871	0.323062	0.012989	-0.30089
0.039585	0.707791	0.026876	0.330059	0.012997	-0.30815
0.03959	0.729382	0.026882	0.336538	0.013006	-0.31521
0.039597	0.750703	0.026888	0.342538	0.013015	-0.32207
0.039603	0.770657	0.026895	0.348041	0.013025	-0.32888
0.03961	0.788893	0.026902	0.353146	0.013035	-0.33566
0.039617	0.80686	0.026909	0.357796	0.013046	-0.34218
0.039625	0.823013	0.026916	0.362115	0.013057	-0.3488
0.039633	0.838925	0.026924	0.366028	0.01307	-0.35512

0.039641	0.853725	0.026932	0.369579	0.013083	-0.36151
0.03965	0.867766	0.026941	0.372895	0.013096	-0.36773
0.039659	0.881204	0.02695	0.375831	0.01311	-0.37389
0.039669	0.893797	0.02696	0.378465	0.013125	-0.38015
0.039679	0.905545	0.02697	0.380848	0.013142	-0.38633
0.03969	0.916789	0.026981	0.382972	0.013159	-0.3925
0.039701	0.927542	0.026993	0.384835	0.013176	-0.39864
0.039714	0.937774	0.027005	0.386439	0.013196	-0.4048
0.039726	0.947275	0.027017	0.387813	0.013216	-0.41105
0.03974	0.956289	0.027031	0.388951	0.013237	-0.4172
0.039754	0.96517	0.027045	0.38986	0.01326	-0.42344
0.039769	0.97333	0.02706	0.390566	0.013284	-0.42974
0.039786	0.981078	0.027076	0.391047	0.013309	-0.43606
0.039803	0.988401	0.027093	0.391334	0.013336	-0.44249
0.039821	0.995399	0.027111	0.391405	0.013364	-0.44898
0.03984	1.00199	0.02713	0.391284	0.013394	-0.45555
0.03986	1.0083	0.027151	0.390959	0.013426	-0.46217
0.039881	1.01432	0.027172	0.390438	0.01346	-0.46895
0.039904	1.01998	0.027194	0.389728	0.013496	-0.47577
0.039928	1.0251	0.027218	0.388818	0.013534	-0.48269
0.039953	1.03014	0.027243	0.38771	0.013574	-0.48972
0.03998	1.03485	0.02727	0.38641	0.013617	-0.49687
0.040009	1.03915	0.027299	0.384906	0.013662	-0.50419
0.040039	1.04328	0.027329	0.383201	0.01371	-0.51159
0.040071	1.04708	0.027361	0.381298	0.013761	-0.51913
0.040105	1.05056	0.027394	0.379187	0.013815	-0.52677
0.040141	1.05372	0.02743	0.376861	0.013872	-0.53457
0.040179	1.05661	0.027468	0.37433	0.013932	-0.5425
0.040219	1.05922	0.027508	0.371584	0.013996	-0.55054
0.040262	1.0615	0.027551	0.368602	0.014064	-0.55876
0.040307	1.06347	0.027596	0.365407	0.014136	-0.5671
0.040355	1.06519	0.027644	0.361975	0.014212	-0.5756
0.040406	1.06657	0.027694	0.358301	0.014293	-0.58419
0.04046	1.06766	0.027748	0.354387	0.014379	-0.59291
0.040517	1.06844	0.027805	0.350236	0.01447	-0.60173
0.040577	1.06894	0.027865	0.345845	0.014566	-0.61065
0.040641	1.06911	0.027929	0.341204	0.014668	-0.61969
0.040709	1.06896	0.027997	0.336324	0.014777	-0.6288
0.040781	1.06853	0.028069	0.331201	0.014891	-0.63794
0.040858	1.06784	0.028145	0.325834	0.015013	-0.64712
0.040939	1.06687	0.028226	0.320224	0.015142	-0.65635
0.041025	1.06564	0.028311	0.314394	0.015278	-0.66556
0.041115	1.06415	0.028402	0.308342	0.015423	-0.67475
0.041212	1.06243	0.028498	0.302054	0.015576	-0.68383
0.041314	1.06052	0.028599	0.295573	0.015739	-0.69277

0.041422	1.05843	0.028707	0.288894	0.015911	-0.70151
0.041537	1.05617	0.028822	0.282044	0.016094	-0.71001
0.041659	1.05382	0.028943	0.275037	0.016288	-0.71823
0.041788	1.05144	0.029071	0.267908	0.016448	-0.726
0.041924	1.04907	0.029208	0.260685	0.016493	-0.73335
0.042069	1.04674	0.029352	0.253399	0.01671	-0.734
0.042223	1.04454	0.029505	0.246083	0.016941	-0.73974
0.042385	1.04257	0.029667	0.238769	0.017186	-0.74582
0.042558	1.04088	0.029839	0.231534	0.017445	-0.75118
0.042741	1.03958	0.030021	0.224447	0.017719	-0.75566
0.042934	1.03879	0.030214	0.217565	0.018011	-0.75916
0.04314	1.03866	0.030419	0.210974	0.018319	-0.76152
0.043358	1.03928	0.030636	0.204752	0.018647	-0.76263
0.043588	1.04082	0.030866	0.198987	0.018993	-0.76226
0.043833	1.04345	0.03111	0.1938	0.019361	-0.76029
0.044092	1.04734	0.031368	0.189306	0.019751	-0.75651
0.044367	1.05267	0.031642	0.185647	0.020164	-0.7507
0.044658	1.05969	0.031932	0.182984	0.020601	-0.74266
0.044967	1.06865	0.03224	0.181519	0.021065	-0.73215
0.04502	1.0696	0.032566	0.181386	0.021557	-0.7189
0.045294	1.07929	0.032912	0.182812	0.022079	-0.70263
0.045641	1.09323	0.033279	0.18603	0.022632	-0.68307
0.046009	1.10997	0.033667	0.191293	0.023218	-0.65989
0.046398	1.12989	0.034079	0.198888	0.023838	-0.63279
0.046811	1.15331	0.034515	0.209135	0.024497	-0.60143
0.047249	1.18072	0.034978	0.222327	0.025195	-0.56533
0.047713	1.21254	0.035469	0.238937	0.025935	-0.52414
0.048205	1.24929	0.035988	0.259316	0.026719	-0.47736
0.048726	1.29154	0.036539	0.283924	0.02755	-0.4245
0.049279	1.33992	0.037123	0.313366	0.028431	-0.36503
0.049864	1.3951	0.037742	0.348083	0.028916	-0.29826
0.050485	1.45784	0.038022	0.368535	0.029365	-0.2502
0.051143	1.52899	0.038398	0.391042	0.030355	-0.22019
0.051841	1.60947	0.039094	0.438409	0.031405	-0.13629
0.05258	1.70035	0.039831	0.493067	0.032518	-0.04264
0.053364	1.80274	0.040613	0.55583	0.033697	0.061983
0.054194	1.91788	0.041441	0.627582	0.034947	0.178494
0.055075	2.04716	0.042319	0.709351	0.036273	0.307626
0.056008	2.19205	0.04325	0.802191	0.037678	0.450882
0.056997	2.3543	0.044236	0.907404	0.039167	0.609946
0.058046	2.53578	0.045282	1.02626	0.040746	0.786552
0.059157	2.73871	0.04639	1.16031	0.041646	0.982707
0.060335	2.96531	0.047565	1.31123	0.04242	1.11647
0.061584	3.21824	0.048811	1.48086	0.044194	1.20787
0.062908	3.50055	0.050131	1.67138	0.046075	1.45111

0.064311	3.81538	0.051531	1.88509	0.04807	1.72178
0.065798	4.16655	0.053014	2.12461	0.050184	2.02334
0.066398	4.33024	0.053513	2.22142	0.052425	2.35978
0.067374	4.57417	0.054586	2.40293	0.054675	2.7355
0.069045	5.01269	0.056253	2.70468	0.054801	3.16039
0.070816	5.50164	0.05802	3.04242	0.05732	3.16995
0.072693	6.04669	0.059893	3.42055	0.059991	3.64256
0.074682	6.6543	0.061878	3.84375	0.062822	4.17291
0.076791	7.33158	0.063983	4.31761	0.065823	4.76874
0.079027	8.08619	0.066214	4.84825	0.068013	5.43919
0.081397	8.92654	0.068579	5.44278	0.069006	6.00051
0.083909	9.86125	0.069068	5.59842	0.07238	6.22001
0.086571	10.8989	0.071085	6.13305	0.075957	7.07489
0.087701	11.3986	0.073743	6.88287	0.07975	8.03841
0.089394	12.0904	0.07656	7.72315	0.081707	9.12332
0.092385	13.361	0.079546	8.66404	0.083771	9.77687
0.095556	14.7527	0.082711	9.71614	0.088035	10.3836
0.098917	16.264	0.084687	10.4518	0.092555	11.7512
		0.086067	10.931	0.095782	13.2685
		0.089624	12.2391	0.097349	14.4681
		0.093396	13.6823		
		0.097393	15.2603		

<u>Final LRR-IP Results for Transverse Velocity Profiles</u>

x/c = 0.6753		x/c = 0.7308		x/c = 0.7863	
Probe Point	Transverse Velocity	Probe Point	Transverse Velocity	Probe Point	Transverse Velocity
0.070614	-0.1764	0.061402	-0.01331	0.051038	-0.15821
0.070618	-0.20732	0.061406	-0.02018	0.051042	-0.17513
0.070622	-0.23973	0.06141	-0.05634	0.051046	-0.19244
0.070626	-0.27162	0.061414	-0.08864	0.051049	-0.20818
0.07063	-0.29966	0.061418	-0.11862	0.051053	-0.22413
0.070634	-0.32632	0.061422	-0.14681	0.051057	-0.23775
0.070638	-0.35075	0.061426	-0.17226	0.051061	-0.24989
0.070642	-0.3743	0.06143	-0.19588	0.051066	-0.26138
0.070647	-0.39611	0.061435	-0.21765	0.05107	-0.27183
0.070651	-0.41552	0.061439	-0.23837	0.051074	-0.28117
0.070656	-0.43319	0.061444	-0.25672	0.051079	-0.29011
0.070661	-0.44972	0.061449	-0.27419	0.051083	-0.29796
0.070665	-0.46583	0.061453	-0.29092	0.051088	-0.30507

0.070671	-0.48019	0.061459	-0.30586	0.051093	-0.31197
0.070676	-0.49375	0.061464	-0.31976	0.051099	-0.31826
0.070681	-0.50678	0.061469	-0.33289	0.051104	-0.32399
0.070687	-0.51882	0.061475	-0.34544	0.05111	-0.32954
0.070693	-0.53011	0.061481	-0.35714	0.051116	-0.3346
0.0707	-0.54109	0.061487	-0.36807	0.051122	-0.33923
0.070706	-0.55125	0.061494	-0.37882	0.051129	-0.34362
0.070713	-0.56069	0.061501	-0.38856	0.051135	-0.34779
0.07072	-0.56975	0.061508	-0.39789	0.051143	-0.35162
0.070728	-0.57826	0.061516	-0.40685	0.05115	-0.3551
0.070736	-0.58633	0.061524	-0.41524	0.051158	-0.35857
0.070745	-0.59402	0.061532	-0.42335	0.051167	-0.36164
0.070754	-0.60125	0.061541	-0.43097	0.051175	-0.36453
0.070763	-0.6084	0.06155	-0.43813	0.051185	-0.36733
0.070773	-0.61475	0.06156	-0.44511	0.051195	-0.36981
0.070784	-0.62098	0.061571	-0.45173	0.051205	-0.37219
0.070795	-0.6268	0.061582	-0.45796	0.051216	-0.37446
0.070806	-0.63245	0.061593	-0.46391	0.051227	-0.37646
0.070819	-0.63783	0.061606	-0.46963	0.05124	-0.37843
0.070832	-0.64281	0.061619	-0.47514	0.051253	-0.38018
0.070846	-0.64761	0.061632	-0.48034	0.051266	-0.38187
0.07086	-0.65204	0.061647	-0.48529	0.051281	-0.38338
0.070876	-0.65646	0.061662	-0.49009	0.051296	-0.3848
0.070892	-0.66057	0.061679	-0.49463	0.051312	-0.3861
0.07091	-0.66431	0.061696	-0.49902	0.051329	-0.38729
0.070928	-0.6681	0.061714	-0.50318	0.051347	-0.38839
0.070947	-0.67154	0.061733	-0.50718	0.051366	-0.38936
0.070968	-0.67479	0.061754	-0.51098	0.051387	-0.39026
0.07099	-0.67799	0.061776	-0.51464	0.051408	-0.39106
0.071013	-0.68083	0.061798	-0.51812	0.051431	-0.39178
0.071037	-0.68361	0.061823	-0.52146	0.051455	-0.39238
0.071063	-0.68623	0.061848	-0.52463	0.051481	-0.39293
0.071091	-0.68863	0.061876	-0.52767	0.051508	-0.39336
0.07112	-0.69097	0.061904	-0.53058	0.051536	-0.39373
0.07115	-0.69304	0.061935	-0.5333	0.051567	-0.39401
0.071183	-0.69503	0.061967	-0.53591	0.051599	-0.3942
0.071217	-0.69689	0.062002	-0.53837	0.051633	-0.39433
0.071254	-0.69852	0.062038	-0.54071	0.051669	-0.39435
0.071293	-0.70008	0.062077	-0.54289	0.051707	-0.39428
0.071334	-0.70151	0.062117	-0.54494	0.051748	-0.39416
0.071377	-0.70273	0.062161	-0.54683	0.051791	-0.39392
0.071424	-0.70389	0.062207	-0.5486	0.051836	-0.3936
0.071472	-0.70488	0.062255	-0.55022	0.051885	-0.3932
0.071524	-0.70573	0.062306	-0.5517	0.051936	-0.39272
0.071579	-0.70648	0.062361	-0.55302	0.05199	-0.39212

0.071637	-0.7071	0.062419	-0.55422	0.052048	-0.39145
0.071699	-0.70757	0.06248	-0.55526	0.052108	-0.39066
0.071764	-0.70794	0.062545	-0.55614	0.052173	-0.3898
0.071834	-0.70822	0.062614	-0.55689	0.052241	-0.38882
0.071907	-0.70834	0.062687	-0.55747	0.052314	-0.38774
0.071985	-0.70838	0.062764	-0.55791	0.052391	-0.38656
0.072067	-0.70833	0.062846	-0.55819	0.052472	-0.38527
0.072155	-0.70815	0.062933	-0.5583	0.052558	-0.38388
0.072247	-0.70789	0.063025	-0.55827	0.05265	-0.38237
0.072346	-0.70755	0.063122	-0.55807	0.052747	-0.38075
0.07245	-0.70713	0.063226	-0.55772	0.052849	-0.37903
0.07256	-0.70663	0.063335	-0.55721	0.052958	-0.37719
0.072677	-0.70608	0.063451	-0.55655	0.053074	-0.37523
0.072801	-0.70546	0.063574	-0.55574	0.053196	-0.37316
0.072932	-0.70479	0.063705	-0.55477	0.053326	-0.37099
0.073071	-0.70413	0.063843	-0.55367	0.053374	-0.36641
0.073219	-0.70339	0.06399	-0.55241	0.053463	-0.36438
0.073375	-0.70268	0.064145	-0.55103	0.053609	-0.36204
0.073541	-0.70195	0.06431	-0.54951	0.053763	-0.35958
0.073716	-0.70124	0.064484	-0.54788	0.053927	-0.35704
0.073902	-0.70058	0.064669	-0.54615	0.0541	-0.35439
0.0741	-0.69995	0.064865	-0.5443	0.054284	-0.35168
0.074309	-0.69939	0.065073	-0.5424	0.054478	-0.34888
0.07453	-0.69891	0.065293	-0.54039	0.054685	-0.34602
0.074765	-0.69854	0.065526	-0.53836	0.054904	-0.34312
0.075014	-0.69828	0.065774	-0.53628	0.055135	-0.34018
0.075278	-0.6982	0.066036	-0.53422	0.055381	-0.33722
0.075558	-0.69825	0.066314	-0.53215	0.055642	-0.33427
0.075854	-0.69851	0.066608	-0.53012	0.055918	-0.33133
0.076168	-0.69899	0.066921	-0.52816	0.056211	-0.32845
0.076501	-0.69972	0.067252	-0.5263	0.056521	-0.32563
0.076855	-0.70071	0.067603	-0.52456	0.05685	-0.32291
0.077229	-0.70201	0.067974	-0.52299	0.057199	-0.32032
0.077625	-0.70363	0.068368	-0.52163	0.057568	-0.31789
0.078046	-0.70563	0.068786	-0.52049	0.05796	-0.31565
0.078491	-0.70801	0.069229	-0.51966	0.058375	-0.31365
0.078963	-0.71081	0.069698	-0.51913	0.058815	-0.31192
0.079464	-0.71406	0.070195	-0.51897	0.059281	-0.3105
0.079994	-0.71781	0.070723	-0.51924	0.059776	-0.30945
0.080557	-0.72206	0.071282	-0.51995	0.0603	-0.30879
0.081153	-0.72685	0.071874	-0.5212	0.060855	-0.3086
0.081784	-0.73219	0.072501	-0.52296	0.061444	-0.3089
0.082454	-0.73811	0.073167	-0.52535	0.062068	-0.30978
0.083163	-0.74463	0.073872	-0.52838	0.062729	-0.31124
0.083916	-0.75172	0.07462	-0.53209	0.06343	-0.31338

0.084713	-0.75942	0.075412	-0.53652	0.064173	-0.31621
0.085558	-0.76765	0.075729	-0.5417	0.064961	-0.31983
0.085571	-0.76905	0.076252	-0.54559	0.065795	-0.32423
0.086453	-0.77757	0.077142	-0.54935	0.06668	-0.3295
0.087402	-0.78657	0.078085	-0.55616	0.067618	-0.33564
0.088408	-0.79595	0.079084	-0.56375	0.068612	-0.34273
0.089474	-0.80559	0.080144	-0.57208	0.069666	-0.35075
0.090603	-0.81532	0.081267	-0.58112	0.070783	-0.35972
0.091801	-0.82492	0.082458	-0.59081	0.071966	-0.36962
0.09307	-0.83412	0.083719	-0.60105	0.073221	-0.38042
0.094416	-0.84258	0.085057	-0.61169	0.074551	-0.39204
0.095842	-0.8498	0.086474	-0.62258	0.075961	-0.40439
0.097353	-0.85529	0.087977	-0.63347	0.077455	-0.41733
0.098955	-0.85832	0.089569	-0.64404	0.079039	-0.43066
		0.091257	-0.65394	0.080718	-0.44413
		0.093047	-0.66263	0.082497	-0.45738
		0.094943	-0.66949	0.08322	-0.46257
		0.096953	-0.67376	0.084383	-0.47054
		0.099084	-0.67446	0.086382	-0.48186
				0.088501	-0.49121
				0.090746	-0.49771
				0.093126	-0.50026
				0.095649	-0.4975
				0.098324	-0.48791

x/c = 0.8418		x/c = 0.8973		x/c = 0.9528	
Probe Point	Transverse Velocity	Probe Point	Transverse Velocity	Probe Point	Transverse Velocity
0.039513	0.001659	0.026805	-0.02343	0.012901	0.043704
0.039517	-0.01395	0.026809	-0.02862	0.012906	0.044557
0.039521	-0.02921	0.026813	-0.03419	0.01291	0.048141
0.039524	-0.04326	0.026816	-0.03968	0.012915	0.051272
0.039528	-0.05686	0.02682	-0.04455	0.01292	0.054473
0.039532	-0.06871	0.026824	-0.04905	0.012925	0.057396
0.039536	-0.0796	0.026828	-0.05327	0.01293	0.060261
0.03954	-0.08954	0.026832	-0.05701	0.012935	0.063015
0.039545	-0.09922	0.026837	-0.06041	0.012941	0.065597
0.039549	-0.10748	0.026841	-0.06355	0.012947	0.068109
0.039554	-0.115	0.026846	-0.06636	0.012953	0.070546
0.039558	-0.12213	0.02685	-0.06889	0.01296	0.072921
0.039563	-0.12899	0.026855	-0.07126	0.012966	0.075234
0.039568	-0.13495	0.02686	-0.07339	0.012974	0.077492
0.039573	-0.14055	0.026865	-0.07535	0.012981	0.079703

0.039579	-0.14602	0.026871	-0.07716	0.012989	0.081869
0.039585	-0.15091	0.026876	-0.07879	0.012997	0.084003
0.03959	-0.15536	0.026882	-0.08029	0.013006	0.086096
0.039597	-0.15972	0.026888	-0.08169	0.013015	0.088143
0.039603	-0.16386	0.026895	-0.08295	0.013025	0.090201
0.03961	-0.16755	0.026902	-0.08411	0.013035	0.092264
0.039617	-0.17119	0.026909	-0.08516	0.013046	0.09427
0.039625	-0.17443	0.026916	-0.08612	0.013057	0.096326
0.039633	-0.17756	0.026924	-0.087	0.01307	0.098314
0.039641	-0.18051	0.026932	-0.08776	0.013083	0.100345
0.03965	-0.18321	0.026941	-0.08846	0.013096	0.102347
0.039659	-0.18582	0.02695	-0.08909	0.01311	0.10435
0.039669	-0.18821	0.02696	-0.08962	0.013125	0.1064
0.039679	-0.19039	0.02697	-0.09008	0.013142	0.10846
0.03969	-0.19247	0.026981	-0.09048	0.013159	0.110542
0.039701	-0.19438	0.026993	-0.09081	0.013176	0.112628
0.039714	-0.19621	0.027005	-0.09106	0.013196	0.114745
0.039726	-0.19785	0.027017	-0.09125	0.013216	0.116924
0.03974	-0.19933	0.027031	-0.09138	0.013237	0.119097
0.039754	-0.20079	0.027045	-0.09143	0.01326	0.121321
0.039769	-0.20207	0.02706	-0.09143	0.013284	0.12359
0.039786	-0.20324	0.027076	-0.09137	0.013309	0.125901
0.039803	-0.20428	0.027093	-0.09124	0.013336	0.128261
0.039821	-0.20523	0.027111	-0.09105	0.013364	0.130679
0.03984	-0.20607	0.02713	-0.0908	0.013394	0.133149
0.03986	-0.20681	0.027151	-0.09049	0.013426	0.135669
0.039881	-0.20746	0.027172	-0.09011	0.01346	0.138269
0.039904	-0.20802	0.027194	-0.08967	0.013496	0.140919
0.039928	-0.20842	0.027218	-0.08916	0.013534	0.143638
0.039953	-0.20875	0.027243	-0.08859	0.013574	0.14642
0.03998	-0.20901	0.02727	-0.08795	0.013617	0.149281
0.040009	-0.20913	0.027299	-0.08723	0.013662	0.15224
0.040039	-0.20917	0.027329	-0.08645	0.01371	0.155264
0.040071	-0.20912	0.027361	-0.0856	0.013761	0.158384
0.040105	-0.20895	0.027394	-0.08466	0.013815	0.161578
0.040141	-0.20867	0.02743	-0.08365	0.013872	0.16488
0.040179	-0.2083	0.027468	-0.08255	0.013932	0.168273
0.040219	-0.20782	0.027508	-0.08138	0.013996	0.171767
0.040262	-0.20724	0.027551	-0.08011	0.014064	0.175381
0.040307	-0.20653	0.027596	-0.07875	0.014136	0.179104
0.040355	-0.20573	0.027644	-0.0773	0.014212	0.182952
0.040406	-0.20482	0.027694	-0.07575	0.014293	0.186913
0.04046	-0.20379	0.027748	-0.07409	0.014379	0.191008
0.040517	-0.20265	0.027805	-0.07233	0.01447	0.195229
0.040577	-0.20139	0.027865	-0.07046	0.014566	0.199585

0.040641	-0.20002	0.027929	-0.06847	0.014668	0.2041
0.040709	-0.19853	0.027997	-0.06637	0.014777	0.208761
0.040781	-0.19692	0.028069	-0.06415	0.014891	0.213567
0.040858	-0.19521	0.028145	-0.0618	0.015013	0.218533
0.040939	-0.19338	0.028226	-0.05933	0.015142	0.223668
0.041025	-0.19143	0.028311	-0.05672	0.015278	0.228968
0.041115	-0.18937	0.028402	-0.05399	0.015423	0.234435
0.041212	-0.1872	0.028498	-0.0511	0.015576	0.240065
0.041314	-0.18493	0.028599	-0.04809	0.015739	0.245851
0.041422	-0.18255	0.028707	-0.04492	0.015911	0.251786
0.041537	-0.18006	0.028822	-0.04162	0.016094	0.257882
0.041659	-0.17748	0.028943	-0.03816	0.016288	0.264113
0.041788	-0.17482	0.029071	-0.03456	0.016448	0.270473
0.041924	-0.17207	0.029208	-0.0308	0.016493	0.276948
0.042069	-0.16923	0.029352	-0.02691	0.01671	0.277561
0.042223	-0.16633	0.029505	-0.02285	0.016941	0.28323
0.042385	-0.16336	0.029667	-0.01865	0.017186	0.289968
0.042558	-0.16034	0.029839	-0.0143	0.017445	0.296766
0.042741	-0.15726	0.030021	-0.00981	0.017719	0.303594
0.042934	-0.15415	0.030214	-0.00519	0.018011	0.310416
0.04314	-0.15102	0.030419	-0.00044	0.018319	0.317208
0.043358	-0.14787	0.030636	0.004417	0.018647	0.323922
0.043588	-0.14472	0.030866	0.0094	0.018993	0.330522
0.043833	-0.14158	0.03111	0.014473	0.019361	0.336952
0.044092	-0.13846	0.031368	0.019644	0.019751	0.343175
0.044367	-0.13537	0.031642	0.024873	0.020164	0.349113
0.044658	-0.13234	0.031932	0.030169	0.020601	0.354728
0.044967	-0.12938	0.03224	0.035479	0.021065	0.359939
0.04502	-0.136	0.032566	0.040805	0.021557	0.364682
0.045294	-0.13337	0.032912	0.046117	0.022079	0.368878
0.045641	-0.13035	0.033279	0.051384	0.022632	0.372453
0.046009	-0.12744	0.033667	0.05659	0.023218	0.375338
0.046398	-0.12468	0.034079	0.061676	0.023838	0.37744
0.046811	-0.12206	0.034515	0.066644	0.024497	0.378703
0.047249	-0.11965	0.034978	0.071425	0.025195	0.379006
0.047713	-0.11746	0.035469	0.075985	0.025935	0.378278
0.048205	-0.11553	0.035988	0.080267	0.026719	0.376459
0.048726	-0.11392	0.036539	0.08423	0.02755	0.373441
0.049279	-0.11265	0.037123	0.087818	0.028431	0.369145
0.049864	-0.11178	0.037742	0.090982	0.028916	0.36351
0.050485	-0.11136	0.038022	0.09226	0.029365	0.358234
0.051143	-0.11144	0.038398	0.093764	0.030355	0.355319
0.051841	-0.1121	0.039094	0.095941	0.031405	0.346515
0.05258	-0.11339	0.039831	0.097515	0.032518	0.336082
0.053364	-0.11538	0.040613	0.098404	0.033697	0.323947

0.054194	-0.11812	0.041441	0.098533	0.034947	0.31016
0.055075	-0.12167	0.042319	0.097826	0.036273	0.29471
0.056008	-0.12611	0.04325	0.096187	0.037678	0.277481
0.056997	-0.13147	0.044236	0.093548	0.039167	0.258364
0.058046	-0.13783	0.045282	0.089791	0.040746	0.237399
0.059157	-0.14523	0.04639	0.084868	0.041646	0.214554
0.060335	-0.15373	0.047565	0.078638	0.04242	0.198738
0.061584	-0.16334	0.048811	0.07105	0.044194	0.188309
0.062908	-0.17409	0.050131	0.061996	0.046075	0.161601
0.064311	-0.18598	0.051531	0.051416	0.04807	0.133136
0.065798	-0.19899	0.053014	0.039241	0.050184	0.102992
0.066398	-0.20486	0.053513	0.034248	0.052425	0.071243
0.067374	-0.21361	0.054586	0.02491	0.054675	0.038021
0.069045	-0.22867	0.056253	0.009381	0.054801	0.002778
0.070816	-0.24455	0.05802	-0.0078	0.05732	0.001989
0.072693	-0.26104	0.059893	-0.02662	0.059991	-0.03388
0.074682	-0.27786	0.061878	-0.04696	0.062822	-0.07055
0.076791	-0.29465	0.063983	-0.0687	0.065823	-0.10766
0.079027	-0.3109	0.066214	-0.09163	0.068013	-0.1447
0.081397	-0.32596	0.068579	-0.1154	0.069006	-0.17186
0.083909	-0.33903	0.069068	-0.12104	0.07238	-0.18241
0.086571	-0.34909	0.071085	-0.14041	0.075957	-0.21711
0.087701	-0.35158	0.073743	-0.16447	0.07975	-0.24892
0.089394	-0.35496	0.07656	-0.18764	0.081707	-0.2763
0.092385	-0.35458	0.079546	-0.20893	0.083771	-0.28759
0.095556	-0.34639	0.082711	-0.22711	0.088035	-0.29795
0.098917	-0.32827	0.084687	-0.23547	0.092555	-0.30965
		0.086067	-0.24097	0.095782	-0.30921
		0.089624	-0.24755	0.097349	-0.2975
		0.093396	-0.24507	0.012901	-0.2924
		0.097393	-0.23088		

Coefficient of Pressure for LRR-IP Turbulence Model

Position along x axis	Pressure
0.999498	101255
0.99849	101262
0.99747	101266
0.996439	101270
0.995396	101274
0.994341	101277
0.993273	101280
0.992193	101283
0.991101	101285
0.989996	101288

0.988877	101290
0.987747	101293
0.986602	101295
0.985443	101297
0.984271	101299
0.983085	101301
0.981885	101303
0.98067	101305
0.97944	101307
0.978196	101308
0.976936	101310
0.975661	101312
0.974369	101314
0.973062	101315
0.971738	101317
0.970398	101319
0.96904	101320
0.967665	101322
0.966272	101324
0.964862	101325
0.963435	101327
0.961987	101328
0.960522	101330
0.959037	101331
0.957532	101333
0.956008	101334
0.954464	101335
0.952898	101337
0.951312	101338
0.949703	101340
0.948074	101341
0.946422	101342
0.944747	101344
0.943049	101345
0.941327	101346
0.939582	101348
0.937811	101349
0.936016	101351
0.934195	101352
0.932348	101353
0.930474	101354
0.928574	101356
0.926646	101357
0.924689	101358
0.922703	101360

0.920687	101361
0.918642	101362
0.916566	101364
0.914458	101365
0.912318	101366
0.910146	101367
0.90794	101369
0.905699	101370
0.903423	101371
0.901112	101372
0.898764	101374
0.896378	101375
0.893953	101376
0.89149	101378
0.888986	101379
0.886441	101380
0.883854	101381
0.881224	101383
0.878549	101384
0.87583	101385
0.873064	101387
0.870251	101388
0.867389	101389
0.864476	101390
0.861512	101392
0.858497	101393
0.855427	101394
0.852301	101396
0.84912	101397
0.845879	101398
0.84258	101400
0.839218	101401
0.835794	101402
0.832305	101404
0.828748	101405
0.825124	101406
0.82143	101408
0.817662	101409
0.813821	101410
0.809902	101412
0.805905	101413
0.801826	101415
0.797664	101416
0.793416	101417
0.789078	101419

0.784649	101420
0.780125	101422
0.775505	101423
0.770783	101425
0.765958	101426
0.761026	101428
0.755984	101429
0.750826	101431
0.745551	101432
0.740153	101434
0.734629	101435
0.728973	101437
0.723183	101439
0.717253	101440
0.711177	101442
0.704951	101444
0.698569	101445
0.692024	101447
0.685312	101449
0.678425	101451
0.671357	101452
0.6641	101454
0.656647	101456
0.648991	101458
0.641122	101460
0.633032	101462
0.624712	101464
0.616151	101466
0.607341	101468
0.598269	101471
0.588923	101474
0.579292	101475
0.569361	101477
0.559119	101481
0.548548	101482
0.537634	101484
0.52636	101487
0.514708	101490
0.502658	101493
0.49019	101496
0.477282	101500
0.463912	101503
0.450051	101507
0.435676	101512
0.420755	101517

0.405257	101523
0.38915	101529
0.372399	101534
0.354965	101540
0.336804	101545
0.318302	101551
0.300337	101556
0.283353	101562
0.267293	101567
0.252109	101573
0.237751	101579
0.224174	101585
0.211336	101591
0.199196	101598
0.187717	101604
0.176861	101611
0.166596	101618
0.156888	101626
0.147709	101633
0.13903	101641
0.130823	101649
0.123063	101657
0.115726	101665
0.10879	101674
0.102233	101683
0.096034	101692
0.090175	101701
0.084638	101709
0.079404	101718
0.074459	101727
0.069788	101735
0.065374	101744
0.061206	101753
0.057269	101761
0.053553	101769
0.050045	101776
0.046734	101783
0.043611	101790
0.040666	101796
0.037889	101800
0.035272	101804
0.032806	101807
0.030484	101808
0.028299	101808
0.026244	101807

0.024312	101804
0.022497	101799
0.020793	101792
0.019195	101783
0.017697	101770
0.016295	101754
0.014983	101735
0.013758	101712
0.012615	101686
0.011549	101655
0.010557	101620
0.009636	101581
0.008781	101537
0.007988	101488
0.007256	101435
0.006579	101378
0.005956	101317
0.005382	101252
0.004855	101184
0.004373	101114
0.003931	101041
0.003527	100966
0.003159	100890
0.002824	100814
0.002519	100738
0.002242	100662
0.001991	100588
0.001764	100515
0.001558	100444
0.001372	100375
0.001204	100309
0.001053	100244
0.000916	100183
0.000793	100125
0.000682	100069
0.000581	100016
0.000491	99966
0.00041	99918.8
0.000337	99873.3
0.000271	99831.5
0.000212	99790.4
0.000158	99753
0.00011	99717.3
6.64E-05	99682.8
2.72E-05	99651

-8.22E-06	99621.1
-4.02E-05	99591.6
-6.90E-05	99564.4
-9.50E-05	99538.6
-0.00012	99513.7
-0.00014	99490.2
-0.00016	99467.8
-0.00018	99446.6
-0.00019	99426.3
-0.00021	99407.4
-0.00022	99388.4
-0.00023	99370.4
-0.00024	99354.9
-0.00025	99336.6
-0.00026	99322.8
-0.00027	99306.1
-0.00027	99293.1
-0.00028	99277
-0.00028	99265.4
-0.00029	99249.4
-0.00029	99238.3
-0.00029	99223.8
-0.0003	99211.8
-0.0003	99199.2
-0.0003	99185.9
-0.0003	99174.8
-0.0003	99160.7
-0.0003	99150.2
-0.0003	99136.1
-0.0003	99125
-0.00029	99112
-0.00029	99099.3
-0.00028	99087.4
-0.00028	99074.3
-0.00027	99061.2
-0.00026	99048.7
-0.00025	99035.7
-0.00024	99022.5
-0.00023	99009.3
-0.00021	98995.8
-0.00019	98982.8
-0.00017	98968.2
-0.00015	98955.5
-0.00013	98941.5
-9.79E-05	98928.4

-6.62E-05	98914.2
-3.06E-05	98901.3
9.25E-06	98888.1
5.39E-05	98875.1
0.000104	98862.9
0.000159	98849.7
0.000221	98839.3
0.000291	98827.5
0.000367	98817.4
0.000453	98808.4
0.000548	98799.5
0.000653	98792.8
0.000769	98786.6
0.000898	98782.3
0.00104	98779.1
0.001197	98777.9
0.00137	98777.6
0.00156	98780.5
0.001769	98783.6
0.001999	98790.1
0.002251	98804.3
0.002527	98823.3
0.002829	98841.5
0.003159	98859.9
0.003519	98879.5
0.003912	98899.7
0.004338	98920.9
0.004802	98945.8
0.005306	98971
0.005852	98997.6
0.006443	99027.9
0.007082	99058.6
0.007773	99089.7
0.008518	99125.2
0.009321	99160.1
0.010186	99194.3
0.011116	99232
0.012115	99270.4
0.013188	99307.3
0.014338	99345.7
0.015571	99383.9
0.016891	99422.5
0.018304	99461.9
0.019814	99499.3
0.021427	99536.2

0.02315	99574.6
0.024989	99611.3
0.026949	99646
0.029039	99683.2
0.031265	99721.1
0.033635	99759.4
0.036159	99792.9
0.038843	99822.7
0.041698	99856.1
0.044733	99888.1
0.047958	99918.1
0.051385	99947.6
0.055024	99981.1
0.058888	100014
0.062989	100040
0.067341	100066
0.071959	100093
0.076857	100120
0.082051	100148
0.087558	100174
0.093395	100198
0.099582	100225
0.106138	100249
0.113084	100269
0.120443	100301
0.128236	100335
0.136488	100354
0.145227	100372
0.154477	100396
0.164269	100419
0.174631	100443
0.185596	100467
0.197196	100491
0.209468	100516
0.222445	100541
0.236169	100566
0.250678	100592
0.266014	100619
0.282222	100647
0.299346	100676
0.317436	100705
0.336539	100736
0.356709	100768
0.377427	100801
0.397618	100833

0.416794	100866
0.435026	100894
0.452385	100921
0.468928	100947
0.484714	100966
0.499793	100985
0.514209	101002
0.528007	101019
0.541225	101034
0.553899	101048
0.566061	101061
0.577743	101072
0.588972	101083
0.599773	101093
0.610171	101101
0.620189	101109
0.629847	101117
0.639163	101124
0.648157	101131
0.656843	101137
0.665239	101143
0.673358	101148
0.681214	101153
0.68882	101157
0.696186	101162
0.703326	101165
0.710249	101169
0.716964	101172
0.723481	101176
0.72981	101178
0.735957	101181
0.741932	101184
0.747741	101187
0.75339	101189
0.758887	101191
0.764238	101193
0.769448	101195
0.774523	101197
0.779468	101199
0.784288	101201
0.788989	101202
0.793574	101204
0.798048	101205
0.802415	101207
0.806678	101208

0.810843	101210
0.814911	101211
0.818886	101212
0.822771	101213
0.82657	101214
0.830286	101215
0.83392	101216
0.837477	101217
0.840959	101218
0.844367	101219
0.847704	101220
0.850972	101221
0.854174	101221
0.857312	101222
0.860388	101223
0.863403	101224
0.86636	101224
0.869258	101225
0.872103	101226
0.874893	101226
0.877631	101227
0.880319	101228
0.882958	101228
0.885549	101229
0.888094	101229
0.890592	101230
0.893048	101230
0.895461	101231
0.897832	101231
0.900162	101232
0.902453	101232
0.904706	101233
0.906921	101233
0.909099	101233
0.911243	101234
0.913352	101234
0.915427	101235
0.917469	101235
0.919479	101235
0.921458	101236
0.923405	101236
0.925324	101236
0.927213	101237
0.929074	101237
0.930907	101237

0.932713	101238
0.934493	101238
0.936246	101238
0.937975	101239
0.939678	101239
0.941358	101239
0.943014	101240
0.944647	101240
0.946257	101240
0.947845	101240
0.949412	101241
0.950958	101241
0.952483	101241
0.953987	101241
0.955473	101242
0.956938	101242
0.958384	101242
0.959812	101242
0.961222	101243
0.962614	101243
0.963987	101243
0.965345	101243
0.966685	101244
0.968009	101244
0.969317	101244
0.970609	101244
0.971885	101245
0.973147	101245
0.974392	101245
0.975625	101245
0.976842	101246
0.978045	101246
0.979234	101246
0.98041	101246
0.981573	101246
0.982722	101247
0.983859	101247
0.984983	101247
0.986095	101247
0.987194	101248
0.988282	101248
0.989358	101248
0.990422	101248
0.991476	101249
0.992518	101249

0.993549	101249
0.994569	101249
0.99558	101250
0.996579	101250
0.997568	101250
0.998548	101250
0.999518	101251