DISSERTATION

NUMERICAL PREDICTION OF TURBULENT FLOW AND AERODYNAMIC LOADING ON BLUFF BODIES

Submitted by

Jaeyong Chung

Department of Civil Engineering

In partial fulfillment of the requirements

for the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2005

COLORADO STATE UNIVERSITY

February 14, 2005

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY JAEYONG CHUNG ENTITLED NUMERICAL PREDICTION OF TURBULENT FLOW AND AERODYNAMIC LOADING ON BLUFF BODIES BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Committee on Graduate Work

Adviser

Department Head

ABSTRACT OF DISSERTATION

NUMERICAL PREDICTION OF TURBULENT FLOW AND

AERODYNAMIC LOADING ON BLUFF BODIES

The effort described in this dissertation led to the development and implementation in a computer code of a hybrid Two-Layer/Large Eddy Simulation (hybrid TL/LES) turbulence model suitable for prediction of flow past and the aerodynamic loading on bluff bodies. The hybrid TL/LES model is based on the Reynolds Averaged Navier-Stokes equation (RANS) in the near wall region and on the Large Eddy Simulation (LES) in the outer region. In the near-wall region, a two-layer model was adopted to allow for a reduction in the number of computational grid points (in the direction normal to the wall) without scarifying the accuracy of the computational results. One-equation model was used to solve for the turbulent kinetic energy. To combine the two-layer model with the LES model, the position of the switching line was automatically established during computations. The hybrid TL/LES model was employed in the unsteady three-dimensional calculations of turbulent flow past a square cylinder, a

surface-mounted cube, and the Texas Tech University test building. Overall, a good agreement was found between the hybrid TL/LES predictions and the corresponding experimental data. A comparison of the hybrid TL/LES and the LES results showed that the same level of accuracy could be accomplished with a significantly smaller number of computational (normal to the wall) grid points when the LES model was replaced by the hybrid TL/LES model. Further studies are needed to fully explore potential of the hybrid TL/LES model in practical applications involving separated turbulent flows.

Jaeyong Chung
Department of Civil Engineering
Colorado State University
Fort Collins, CO, 80523
Spring 2005

ACKNOWLEDGMENTS

I wish to express my sincere gratitude to my advisor, Dr. Bogusz Bienkiewicz, for his guidance during my study and preparation of my dissertation. His advice and encouragement helped me in my study and in my research, and it will be an invaluable asset in my future endeavors.

I also would like to express my appreciation of the advice and encouragement I received from the members of my Graduate Committee: Dr. Robert N. Meroney, Dr. David E. Neff, Dr. Martin Gelfand, and Dr. Peter P. Sullivan. Special thanks must be given to Dr. Sullivan, who provided invaluable input on emerging trends in numerical simulations of turbulent flows.

I also would like to thank my colleagues, Juntack Lim and Munehito Endo, for their kindness and support during my graduate study at Colorado State University.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Roman Alphabet

A_P, A_E, A_W, A_N, A_S, A_T, A_B	Coefficients in a Poisson equation
C_k, C_ε	Constants in SGS turbulence kinetic transport equation
C_p	Pressure coefficient
C_s	Smagorinsky constant in LES model
$C_{\varepsilon 1}$, $C_{\varepsilon 2}$	Constants in the $k-\varepsilon$ turbulence model
$\tilde{d}$	Length scale in DES model
F	Flux
k	Turbulent kinetic energy
k_{SGS}	Sub-grid turbulent kinetic energy
l_s	Smagorinsky length scale
l_μ, l_ε	Length scale of two-layer model

$\dot{m}_\phi$	Mass flux
$\vec{n}$	Unit vector normal to a surface
p'	Pressure correction
P_k	Production term
Re_z	Local Reynolds number
S	Magnitude of the vorticity in the Spalart-Allmaras model
$\bar{S}_{ij}$	Large scale strain-rate tensor
S_ϕ	Area of the cell face ϕ
t	Simulation time
u_*	Friction velocity
u_i	Instantaneous velocity in i direction
u_i'	Fluctuation velocity in i direction
$\bar{U}_i$	Mean velocity in i direction
V_{ijk}	Volume of cell (i, j, k)
x, y, z	Cartesian coordinate in physical domain
z^+	Non-dimensional distance
z_0	Aerodynamic roughness length

Greek Alphabet

δ_{ij}	Kronecker delta (1 when $i=j$)
Δ	Filter width
Δt	Time step
$\widehat{\Delta}$	Test filter
ε	Dissipation rate of turbulent kinetic energy
κ	von Kármán constant
μ	Molecular viscosity
μ_t	Eddy viscosity
ρ	Mass density
τ_{ij}^{sgs}	Sub-grid scale stress tensor
τ_w	Wall shear stress

Super- and Subscripts

conv	Convection
diff	Diffusion

i, j, k	Cartesian coordinate index indicating x_i or x_j direction
n	Time level
ref	Reference
ϕ	East, West, North, South, Top, and Bottom faces of cell

Special Symbols

$\overline{f}$	Filtered variable
$G(x,x')$	Filter function
$G(x)$	Top-hat filter
I(z)	Turbulence intensity

Abbreviations

CFL	Courant-Friedrich-Levy number
CWE	Computational Wind Engineering
DES	Detached Eddy Simulation
FVM	Finite Volume Method

LES	Large Eddy Simulation
NSE	Navier-Stokes Equations
RANS	Reynolds-Averaged Navier-Stokes equation
S-A	Spalart-Allmaras model
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
TL	Two-Layer model

CHAPTER 1. INTRODUCTION

1.1. OVERVIEW OF COMPUTATIONAL WIND ENGINEERING

Computational Wind Engineering (CWE) is a relatively new discipline that employs techniques of computational fluid dynamics to overcome the limitations of physical modeling of wind engineering problems. It emerged in 1980's.

Currently, a majority of issues of interest in wind engineering community – e.g. determination the wind velocities, wind-induced pressures and overall wind forces on buildings and structures, and dispersion of pollutants in the urban environment - are addressed through wind tunnel testing. There are no doubts that the wind tunnel tests are very useful and in many cases the only practical techniques to analyze these problems. However, physical modeling has its own limitations, e.g. inability to model wind environments where all (or most) properties of atmospheric boundary layer flows would be correctly duplicated at the laboratory scales. The CWE has the potential to overcome these and other physical limitations by employing numerical methods. It appears that as advances in the CWE are achieved, computational modeling might become a primary

tool to be used in wind engineering analyses of wind effects on built and natural environments.

Numerical studies of turbulent flows around various building shapes have been reported by a number of researchers, Yeung and Kot (1985), Paterson and Apelt (1986), Mathews (1987), Murakami and Mochida (1988), and others. In these studies various turbulence models were employed to account for wind turbulence. The most commonly applied model has been the standard k-ε model. It was applied in analysis of many flow phenomena and it led to reasonable predictions of the mean quantities. Although this model has been widely used in the CWE, it has significant shortcomings that have resulted in limitations in prediction of some of crucial flow characteristics. For example, the use of the standard k-ε model leads to the over-production of turbulent kinetic energy in areas near the frontal corners of flows past bluff bodies. This over-production is caused by the assumption of the (isotropic) eddy viscosity concept which is violated in flow regions dominated by flow impingement or other flow features associated (for example) with flow separation and reattachment.

Many attempts have been made to improve the standard k-ε model through introduction of a number of modifications of this model. In addition, more sophisticated models, such as Reynolds Averaged Navier-Stokes (RANS) models, have been proposed

(Kato and Launder 1993, Murakami *et al.* 1997, and others). Introduction of these models led to significant improvements in flow predictions, e.g. it resulted in significant reduction in the over-production of turbulent kinetic energy around the frontal corner exhibited by the standard k-ε model.

In parallel to these developments, early attempts to apply the Large Eddy Simulation (LES) in wind engineering were reported in late 1980ties, Murakami *et al.* (1987). Recently the LES has been mostly used as a research tool (in academic environments and research laboratories) to study the physical nature of turbulence, through computer simulations. Over the years, a number of wind engineering investigations employing the LES have been reported (CWE 92, CWE 96).

Away from solid surfaces, the LES represents a powerful approach providing a reasonable description of flow at energy-containing large scales, that are typically dependent on geometry (e.g. of obstacles placed in flow) and on boundary conditions. When the LES is applied to boundary layer flows, the computational cost for the whole domain does not differ significantly from that of the Direct Numerical Simulation (DNS), which involves direct integration of the full Navier-Stokes equations. The "large eddies" close to the wall are physically small and in high Reynolds number boundary layers the LES may not sufficiently resolve the near-wall structures. This deficiency will typically

lead to inaccurate modeling of boundary layer growth and separation.

In 1997, Spalart *et al.* proposed a concept of the Detached-Eddy Simulation (DES), a hybrid method which has the RANS behavior near the wall and becomes the LES in regions away from solid surfaces. In short, the DES is a non-zonal technique that is computationally feasible for high Reynolds number prediction, and it also resolves time-dependent, three-dimensional turbulent motions as the LES does. Modeling of flow past a circular cylinder using the DES was reported by Travin *et al.* (1999), while applications of the DES to simulate channel flows were presented by Nikitin *et al.* (2000). Numerical predictions using the DES for a full aircraft were reported by Squires *et al.* (2001). Davidson (2001) proposed a hybrid RANS/LES model which consisted of the k-ω model in the near wall regions and Yoshizawa's one-equation sub-grid scale model in the remaining flow regions. Krajnović and Davidson (2002) combined a one-equation sub-grid model with dynamic LES model for computing flow around a surface-mounted cube. Dejoan and Schiestel (2002) suggested new empirical constants incorporated in the DES model, for the use in the hybrid RANS/LES model. Moreover, they employed the Hassid and Poreh model (1975) to close the sub-grid scale turbulent kinetic energy transport equation and to adjust some constants incorporated in the hybrid RANS/LES model. For compressible flows, Hamed *et al.* (2003) used the DES for modeling of the

unsteady three-dimensional supersonic flow over a cavity. Breuer *et al.* (2003) investigated the results of the DES, the RANS, and the LES calculations for the separated flow around a flat plate at high incidence. Mani (2004) tested a hybrid RANS/LES model based on the length scale of Menter's k-ω model (1994). Table 1.1 provides a partial listing of reported applications of the DES in various computational studies.

In spite of its potential, literature review reveals that the DES has not been widely applied in modeling of problems of bluff body aerodynamics. Limited testing of this technique for flow past three-dimensional wall-mounted cube was recently reported by Krajnović and Davidson (2002). Unfortunately, no references were found on application of the DES in the field of wind engineering.

Table 1.1 Computational Studies Employing DES

Year	**Authors**	**Hybrid Methods**	**Applications**
1997	Spalart *et al.*	LES Spalart-Allmaras Model	Airplane Wing
1999	Travin *et al.*	LES Spalart-Allmaras Model	Flow Past Circular Cylinder
2000	Nikitin *et al.*	LES Spalart-Allmaras Model	Open Channel

2001	Strelets	LES Mentor's SST Model	NACA 0012 Airfoil
2002	Krajnović Davidson	LES 1-Equation Sub-Grid Model	Surface-Mounted Cube
2002	Morton *et al.*	LES Mentor's SST Model	Delta Wing
2002	Dejoan Schiestel	LES 1-Equation Sub-Grid Model	Open Channel
2002	Langhe *et al.*	LES RNG Model	Backward-Facing Step
2002	Battern *et al.*	LES Standard k-ε Model	Backward-Facing Step
2003	Georagiadis *et al.*	LES Cebeci-Smith Algebraic Model	Supersonic Flow
2003	Breuer *et al.*	LES Spalart-Allmaras Model	Flat Plate at High Incidence
2003	Xiao *et al.*	LES k-ξ Model	Flat Plate
2004	Mani	LES Spalart-Allmaras Model	Transonic Jet Flow

1.2. OBJECTIVES AND SCOPE OF STUDY

The objectives of this dissertation were (1) to propose a hybrid Two-Layer/Large Eddy Simulation (TL/LES) model and (2) to demonstrate the effectiveness of the hybrid TL/LES model in the field of wind engineering. The hybrid TL/LES model developed in this study is a combination of the two-layer model proposed by Rodi (1988) and the LES model. The original DES (Spalart, *et al.*, 1997) modeling of the separated flow leads to

some problems. In part it is due to the fact that the Spalart-Allmaras model (S-A model) (Spalart and Allmaras, 1994) was originally developed for the RANS only, and for low Reynolds number flows. Usually the DES works well in purely the RANS mode, if the flow does not separate (Shur *et al.*, 1999). However, it cannot be relied on to predict the decay of homogeneous, isotropic turbulence and in other applications. The S-A model has been also criticized for its inability to rapidly accommodate changes in length scale. In contrast with the S-A model, the two-layer model has been reported to give more promising results than other low Reynolds number models or wall function approaches (Hwang and Jaw, 1998). In addition, the two-layer model is simpler than any other two-equation turbulence model listed in Table 1.1. In view of the shortcomings of the S-A model, the aim of this study was to propose a hybrid TL/LES computational model that would comprise of the two-layer model in the near solid surface regions and of the LES away from the wall. To test this model, a computer code was developed and it was employed in a number of bluff body test cases, to validate the application of the hybrid TL/LES model in the field of computational wind engineering.

The content of this dissertation is as follows. Chapter 2 presents the theoretical background on turbulence models. This chapter reviews the general theory behind the most popular turbulence models. It includes discussion of the standard k-ε model, two-

layer model, the LES, the sub-grid one equation model, and the DES. Pertinent background information on numerical techniques employed in computational analysis of fluid flows is synthesized in Chapter 3. This chapter presents details on the methods of numerical discretization, including methods employed in dealing with each term in the governing equations, using the finite volume method. A three time level method, as the time integration technique, is also reviewed in this chapter. Chapter 4 discusses boundary conditions implemented in numerical simulations of steady and unsteady flows. Chapter 5 introduces the governing equations of a hybrid TL/LES model. It includes a discussion of the procedure for establishing the matching line between the two-layer model and the LES model. The developed computer code implementing the hybrid TL/LES model is described in this chapter. A full listing of this code is provided in Appendix. Chapter 6 demonstrates the performance of the hybrid TL/LES model, for a number of bluff body flow fields. The presented numerical results are compared with computational simulations by other researchers and with experimental results. Finally, Chapter 7 summarizes the conclusions resulting from this study and presents recommendations for future investigations.

CHAPTER 2. TURBULENCE MODELING

Most of problems of wind engineering involve high Reynolds number and, as a result, the wind and wind-induced loads are turbulent. Various turbulence models have been proposed and adapted to solve the problems of computational wind engineering. The most commonly used model employed in such applications is the standard k-ε model. It is a simple model that exhibits a good performance in time-independent implementations. Its performance, however, deteriorates in flows with large strains, in rotating flows, and in time-dependent (unsteady) cases. These limitations led to development of improved turbulence models aimed towards overcoming these shortcomings. The time-dependent turbulence models are useful in prediction of instantaneous values that are critical in the wind-resistant design of buildings and in analysis of the behavior of flows around buildings and structures. This chapter presents an overview of turbulence models that have been employed in wind engineering applications and in other disciplines involving turbulent flows.

2.1. THE REYNOLDS AVERAGED NAVIER-STOKES EQUATION

The ensemble or time-averaged Navier-Stokes equation (NSE) is called the Reynolds Averaged Navier-Stokes equation (RANS). This equation is used to calculate the steady-state velocity and pressure. It accounts for the velocity and pressure fluctuations through additional modeled variables. The modeled variables are dealt with in a number of ways, ranging from the application of the eddy viscosity concept through the second moment closure models which represent the effects of each component of the Reynolds stress tensor on the mean flow.

Numerical modeling of any fluid flow problem requires the solution of the governing equations of fluid motion: the Navier-Stokes equations and the continuity equation. When the flow is turbulent, the instantaneous variables, for example velocity components u_i can be decomposed into the mean $\overline{U}_i$ and fluctuation $u_i^{'}$,

$$u_i = \overline{U}_i + u_i^{'} \qquad (i = 1,\ 2,\ 3) \tag{2.1a}$$

A similar decomposition can be introduced for pressure

$$p = \overline{P} + p^{'} \tag{2.1b}$$

The general form of the three dimensional incompressible continuity and Navier-Stokes equations expressed in Cartesian coordinate system is as follows:

$$\frac{\partial(\rho u_i)}{\partial x_i} = 0$$
$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] \tag{2.2}$$

where, ρ is the mass density and μ is the molecular viscosity.

Inserting Eq. (2.1) into Eq. (2.2), combined with temporal (Reynolds) averaging, leads to the time-averaged continuity equation and Navier-Stokes equations

$$\frac{\partial(\rho \overline{U}_i)}{\partial x_i} = 0$$
$$\frac{\partial(\rho \overline{U}_i)}{\partial t} + \frac{\partial(\rho \overline{U}_i \overline{U}_j)}{\partial x_j} = -\frac{\partial \overline{P}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial \overline{U}_i}{\partial x_j} + \frac{\partial \overline{U}_j}{\partial x_i}\right) - \overline{\rho u_i' u_j'}\right] \tag{2.3}$$

A new term, $-\overline{\rho u_i' u_j'}$, that appears on the right-hand side of Eq. (2.3) is the Reynolds stress tensor, which is symmetric and it represents correlations between velocity fluctuations. This new, turbulence-induced, term requires modeling that is needed to

close Eq. (2.3).

Boussinesq (1877) postulated that the Reynolds stresses should be proportional to the mean strain rates.

$$-\rho\overline{u_i'u_j'} = \mu_t\left(\frac{\partial \overline{U}_i}{\partial x_j}+\frac{\partial \overline{U}_j}{\partial x_i}\right) \tag{2.4}$$

where, μ_t is the eddy viscosity, due to the flow turbulence. This concept assumes that both the viscous stresses and the Reynolds stresses act on the mean flow in a similar manner. Substitution of Eq. (2.4) in Eq. (2.3) results in the equation involving mean (time-averaged) variables and enhanced flow viscosity through the term involving the eddy viscosity μ_t. Using this approach, the modeling process can be completed provided that the eddy viscosity is expressed in terms of other variables.

2.2. THE k-ε MODEL

The k-ε model (Launder and Spalding, 1990) comprises of two model transport equations. One is for the flow turbulent kinetic energy k, the other is for the dissipation rate ε of turbulent kinetic energy. These variables are used to define the velocity scale, υ,

and the length scale, l, of the flow field, at any location and time instant. The velocity scale is defined as

$$\upsilon = k^{1/2} \tag{2.5}$$

The turbulent energy (of the large scales) is postulated to be dissipated over a time period proportional to l/υ, which gives

$$\varepsilon = Const.\left(\frac{\upsilon^2}{l/\upsilon}\right) = Const.\left(\frac{\upsilon^3}{l}\right) \tag{2.6}$$

Equations (2.5-6) imply the length scale

$$l = Const.\frac{k^{3/2}}{\varepsilon} \tag{2.7}$$

The eddy viscosity is computed using Eq. (2.6) and Eq. (2.7)

$$\mu_t = \rho C_\mu l\upsilon = \rho C_\mu k^2/\varepsilon \tag{2.8}$$

where, C_μ is an empirical constant. The turbulent kinetic energy k and its dissipation rate ε are determined from their transport equations. From the Navier-Stokes equations we can derive the exact equation for k

$$\underbrace{\frac{\partial(\rho k)}{\partial t}}_{I}+\underbrace{\overline{U}_i\frac{\partial(\rho k)}{\partial x_j}}_{II}=\underbrace{-\overline{\rho u_i' u_j'}\frac{\partial \overline{U}_i}{\partial x_j}}_{III}-\underbrace{\frac{\partial}{\partial x_j}\left[\overline{u_j'\left(p+\frac{\rho u_i' u_i'}{2}\right)}\right]}_{IV}+\underbrace{\mu\frac{\partial}{\partial x_j}\left(\frac{\partial k}{\partial x_j}\right)}_{V}$$

$$\underbrace{-\mu\overline{\frac{\partial u_i'}{\partial x_j}\frac{\partial u_i'}{\partial x_j}}}_{VI} \qquad (2.9)$$

The terms in Eq. (2.9) have the following meaning:

(*I*) Transient term

(*II*) Convection term

(*III*) Production of kinetic energy

(*IV*) Turbulent diffusion by pressure-velocity fluctuations and by velocity fluctuations

(*V*) Viscous diffusion term

(*VI*) Viscous dissipation term

A number of the above terms are unknown; the production term (*III*), the turbulent diffusion term (*IV*), the viscous diffusion term (*V*), and the viscous dissipation term (*VI*). They need to be modeled using additional equations. The production term can be expressed as follows:

$$P_k = -\overline{\rho u_i' u_j'} \frac{\partial \overline{U}_i}{\partial x_j} = \mu_t \left(\frac{\partial \overline{U}_i}{\partial x_j} + \frac{\partial \overline{U}_j}{\partial x_i} \right) \frac{\partial \overline{U}_i}{\partial x_j} - \frac{2}{3} \rho k \frac{\partial \overline{U}_i}{\partial x_i} \delta_{ij} \qquad (i, j = 1,\ 2,\ 3) \qquad (2.10)$$

where, δ_{ij} is the Kronecker delta (1, when $i = j$). In order to model the turbulent diffusion term, it is assumed that k is diffused down the gradient from regions of high k to regions of low k

$$\overline{u_j' \left(p + \frac{\rho u_i' u_i'}{2} \right)} = -\frac{\mu_t}{\sigma_k} \frac{\partial k}{\partial x_j} \qquad (2.11)$$

where σ_k is an empirical constant. The pressure diffusion term is typically small and it is commonly neglected. After Eq. (2.10) and Eq. (2.11) are substituted into Eq. (2.9), the modeled equation for k is obtained

$$\frac{\partial(\rho k)}{\partial t}+\overline{U}_i\frac{\partial(\rho k)}{\partial x_j}=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]+P_k-\rho\varepsilon \tag{2.12}$$

An exact equation for ε can be derived from the Navier-Stokes equation, but it is complicated and an alternative approach, based on physical reasoning, is taken. Because ε is viewed as the energy-flow rate in the cascade, it is determined by the large-scale motions, independent of the viscosity at high Reynolds number. In the exact equation for ε the production term includes, turbulent quantities and velocity gradients, as in the k equation.

$$\frac{\partial(\rho\varepsilon)}{\partial t}+\overline{U}_i\frac{\partial(\rho\varepsilon)}{\partial x_j}=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial\varepsilon}{\partial x_j}\right]+C_{\varepsilon 1}\frac{\varepsilon}{k}P_\varepsilon-C_{\varepsilon 2}\rho\frac{\varepsilon^2}{k} \tag{2.13}$$

where, P_ε is the production term, $C_{\varepsilon 2}\rho\varepsilon^2/k$ is the dissipation term, σ_ε, $C_{\varepsilon 1}$, and $C_{\varepsilon 2}$ are empirical constants. Equations (2.12) and (2.13) constitute the standard k-ε model. They involve five constants determined from experimental studies carried out for simple flows. The following (standard) values of the model constants were proposed by Launder and Sharma (1974): $C_\mu=0.09$, $C_{\varepsilon 1}=1.44$, $C_{\varepsilon 2}=1.92$, $\sigma_k=1.0$, and $\sigma_\varepsilon=1.3$.

2.3. TWO-LAYER MODEL

For a two-equation turbulence model, many substitutions for the wall functions approach have been made. Some researchers considered a low Reynolds number model by incorporating either a wall damping effect or a direct effect of molecular viscosity on the empirical constants and functions in the turbulence transport equations, which were devised originally for high Reynolds number, fully turbulent flows remote from the wall. Others applied the idea of a two-layer model by resolving the viscosity-affected near-wall layer with a simpler and numerically more stable algebraic model (Jaw and Chen, 1998).

The two-layer model involves a length scale model, replacing the inconsistent and improperly behaved ε equation in the near-wall layer. The two-layer model has been suggested to save grid points in the near wall (typically, 60-100 grid points across boundary layers are required for proper numerical resolution) and to save computer time. In the two-layer model, the flow is separated into two regions: the main flow region where the standard k-ε model is used, and the near-wall region which is analyzed by a simpler model that accurately resolves the boundary layer.

Rodi (1988) adopted the length-scale of Norris and Reynolds (1975). This model was found to perform well in boundary layers with an adverse pressure gradient and

transpiration. The k-equation shown below is solved with a new dissipation term based on the eddy-viscosity assumption

$$\frac{\partial(\rho k)}{\partial t}+\overline{U}_i\frac{\partial(\rho k)}{\partial x_j}=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]+P_k-\rho\frac{k^{2/3}}{l_\varepsilon} \tag{2.14}$$

where the eddy viscosity is specified as

$$\mu_t=C_\mu\rho\sqrt{k}\ell_\mu \tag{2.15}$$

Norris-Reynolds model employs the following relation for the length-scale, ℓ_μ:

$$\ell_\mu=C_\ell z\left[1.0-\exp\left(-\frac{Re_z}{A_\mu}\right)\right] \tag{2.16}$$

where, the constant C_ℓ is chosen as $C_\ell=\kappa C_\mu^{-3/4}$, κ is the von Kármán constant, A_μ is the model constant of 50.5, and $Re_z\,(=\sqrt{k}z/\nu)$ is the local Reynolds number of turbulence. Re_z varies relatively slowly along lines parallel to the wall. It does not vanish at separation and it remains well defined in regions of flows reversal. For the

length-scale ℓ_ε, Norris and Reynolds proposed the following damping function:

$$\ell_\varepsilon = \frac{C_\ell z}{1.0 + \frac{5.3}{Re_z}} \tag{2.17}$$

The damping function above is employed to compute the length scales of the hybrid TL/LES model.

2.4. LARGE EDDY SIMULATION

Large eddy simulation (LES) directly resolves large eddies of turbulent flows and models small-scale eddies. The LES is an intermediate technique between direct numerical solution of the Navier-Stokes Equations (NSE) – the Direct Numerical Simulation (DNS) - and the RANS. The LES is based on observation that, while large turbulent eddies are flow-dependent, small scales tend to be more universal and asymptotically isotropic. Thus, the modeling of these sub-grid (small) scales eddies appears to be significantly simpler than that of the large scales. Significant contribution to the development and early implementation of the LES were made by Smagorinsky (1963), Lilly (1967), and Deardorff (1974). Detailed reviews of the LES were presented by Pope (2000), Orlandi (2001), Piomeli *et al.* (2001), and Caughey and Jothiprasad (2002).

The LES decomposition of turbulent flow into large and small scales, summarized herein, is based on the filtering operation proposed by Leonard (1974) and defined by

$$\overline{f}(x) = \int G(x,x')f(x')dx' \tag{2.18}$$

where $\overline{f}(x)$ is the filtered variable $f(x')$ and $G(x,x')$ is the filter function. Filter functions applied in the LES implementations included a Gaussian filter, a box filter, and a cutoff filter which eliminates all Fourier coefficients belonging to wave numbers above the cutoff value. The most commonly used filter function is the top-hat filter, defined as

$$G(x) = \begin{cases} 1/\Delta, & \text{if } |x| \le \Delta/2 \\ 0 \quad , & \text{otherwise} \end{cases} \tag{2.19}$$

where Δ is the filter width. The eddies larger in size than Δ are directly computed while eddies smaller than Δ are modeled.

Applying a filter operation (Eq. 2.18) to the continuity and the Navier-Stokes equations (Eq. 2.2) yields the following filtered equations

$$\frac{\partial\left(\rho\overline{U}_i\right)}{\partial x_i}=0$$
$$\frac{\partial\left(\rho\overline{U}_i\right)}{\partial t}+\frac{\partial\left(\rho\overline{U_iU_j}\right)}{\partial x_j}=-\frac{\partial\overline{P}}{\partial x_i}+\frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial\overline{U}_i}{\partial x_j}+\frac{\partial\overline{U}_j}{\partial x_i}\right)\right] \tag{2.20}$$

The filtered NSE differs from the original equations because the stress in the convection term $\overline{U_iU_j}$ is different from the product of the filtered velocities. The difference is the sub-grid scale stress tensor τ_{ij}^{sgs} which is defined by

$$\tau_{ij}^{sgs}=\overline{U_iU_j}-\overline{U}_i\overline{U}_j \tag{2.21}$$

Substitution of Eq. (2.21) into Eq. (2.20) leads to

$$\frac{\partial\left(\rho\overline{U}_i\right)}{\partial x_i}=0$$
$$\frac{\partial\left(\rho\overline{U}_i\right)}{\partial t}+\frac{\partial\left(\rho\overline{U}_i\overline{U}_j\right)}{\partial x_j}=-\frac{\partial\overline{P}}{\partial x_i}+\frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial\overline{U}_i}{\partial x_j}+\frac{\partial\overline{U}_j}{\partial x_i}\right)\right]-\frac{\partial\tau_{ij}^{sgs}}{\partial x_i} \tag{2.22}$$

Equation (2.22) is the final form of the filtered NSE in the LES. The sub-grid scale tensor τ_{ij}^{sgs} is formally similar to the Reynolds stress tensor of the RANS. This sub-grid scale tensor will vanish when the filter width is reduced to zero, $\Delta\to 0$. It must be modeled to

achieve closure of the system of equations.

2.4.1. Smagorinsky Model

The most commonly used Sub-Grid Scale model (SGS model) was proposed by Smagorinsky (1963). It is an eddy viscosity model which is based on the observation that the SGS stress tensor leads to an increase in momentum transport and dissipation. Using the concept of the eddy viscosity model, the sub-grid scale stress tensor can be written as

$$\tau_{ij}^{sgs} - \frac{1}{3}\tau_{kk}^{sgs}\delta_{ij} = \mu_t\left(\frac{\partial \overline{U}_i}{\partial x_j} + \frac{\partial \overline{U}_j}{\partial x_i}\right) = 2\mu_t \overline{S}_{ij} \tag{2.23}$$

where, μ_t is the eddy viscosity and $\overline{S}_{ij} = \left(\partial \overline{U}_i / \partial x_j + \partial \overline{U}_j / \partial x_i\right)/2$ is the large-scale strain-rate tensor. The form of the sub-grid scale eddy viscosity (SGS viscosity) is expressed as follows:

$$\mu_t = \rho l_S^2 \left|\overline{S}\right| = \rho\left(C_s\Delta\right)^2 \left|\overline{S}\right|$$
$$\left|\overline{S}\right| = \left(\overline{S}_{ij}\overline{S}_{ij}\right)^{1/2} \tag{2.24}$$

where, C_s is Smagorinsky constant, l_S is Smagorinsky length scale, and $|\bar{S}|$ is the characteristic filtered rate of strain. The theoretical value of C_s for isotropic turbulence is $C_s \approx 0.17$ (Lilly, 1967). In practical applications, C_s takes values between 0.065 and 0.2. This constant depends on the flow problem Reynolds number and strictly speaking it is not a constant. The length-scale Δ is usually determined by the size of the control volume of the grid $(\Delta x, \Delta y, \Delta z)$. It was found through numerical experimentation that in order to improve accuracy/convergence of the LES calculations C_s must be decreased in the presence of shear, near solid boundaries or in transitional flows. Moin and Kim (1982) suggested new l_S, using a Van Driest damping function

$$l_S = C_s \Delta \left[1.0 - \exp(z^+ / A^+)\right] \tag{2.25}$$

where, $z^+ (= \rho u_\tau z / \mu)$ is a non-dimensional distance from the wall and A^+ is a constant equal to 25.0.

2.4.2. Dynamic Model

Many advanced sub-grid models have been proposed to overcome shortcomings of the Smagorinsky model. One example of such improvements is the dynamic eddy

viscosity model proposed by Germano *et al.* (1991). It is based on the introduction of two filters; one is the grid filter which defines the resolved and the sub-grid scales, the other is a test filter, whose width $\hat{\Delta}$ is larger than that of the grid filter Δ. This test filter is applied to the filtered Navier-Stokes equation. It defines a new sub-grid scale stress tensor, T_{ij}, which is similar to τ_{ij}^{sgs}. The relation between T_{ij} and τ_{ij}^{sgs} is given by Germano's identity, (Germano *et al.* 1990).

$$L_{ij} = T_{ij} - \tau_{ij}^{sgs} \tag{2.26}$$

Equation (2.26) can be expressed by means of the Smagorinsky model. If $|\bar{S}| = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}}$, it follows that

$$L_{ij} = -\left(2C_s\hat{\Delta}^2|\hat{\bar{S}}|\hat{\bar{S}}_{ij} - 2C_s\Delta^2|\bar{S}|\bar{S}_{ij}\right) = -2C_s\Delta^2 M_{ij} \tag{2.27}$$

In this expression, $M_{ij} = \left(\hat{\Delta}^2/\Delta^2|\hat{\bar{S}}|\hat{\bar{S}}_{ij} - |\bar{S}|\bar{S}_{ij}\right)$. Both L_{ij} and M_{ij} (which are functions of x and t) are known in terms of $\bar{U}(x,t)$. This information can be used to determine the value of the Smagorinsky coefficient. Lilly (1992) stated that the mean-square error is minimized when C_S is specified as

$$C_S(x,t) = -\frac{1}{2\Delta^2}\frac{L_{ij}M_{ij}}{M_{ij}M_{ij}} \tag{2.28}$$

Equation (2.28) is not mathematically self-consistent since it requires that a spatially-dependent coefficient be extracted from a filtering operation (Cabot and Moin, 1993). To overcome this problem, spatial averaging (in the three directions) was proposed

$$C_s(x,t) = -\frac{1}{2\Delta^2}\left\langle\frac{L_{ij}M_{ij}}{M_{ij}M_{ij}}\right\rangle \tag{2.29}$$

With this implementation, the dynamic model leads to good modeling of transitional and fully turbulent flows, provided that the employed grid is fine enough to resolve the near wall energy-containing motions.

2.5. THE SUB-GRID SCALE ONE EQUATION MODEL

All of the LES models for the residual-stress tensor are considered. By adding a transport equation for the sub-grid scale kinetic energy, it is reasoned that the accuracy of the LES sub-grid scale stress model will be improved. The model equation is of the same form as a Reynolds-stress closure. The instantaneous rate of dissipation ε of the SGS

kinetic energy is related to k_{SGS} and the cutoff filter width Δ by

$$\varepsilon_{SGS} = \frac{C_\varepsilon k_{SGS}^{3/2}}{\Delta} \tag{2.30}$$

with the eddy viscosity given by

$$\mu_{t,SGS} = \rho C_k \Delta k_{SGS}^{1/2} \tag{2.31}$$

Yoshizawa (1993) proposed one-equation model for the SGS turbulent kinetic energy as follows

$$\frac{\partial(\rho k_{SGS})}{\partial t} + \overline{U}_i \frac{\partial(\rho k_{SGS})}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\mu + \mu_{t,SGS})\frac{\partial k_{SGS}}{\partial x_j}\right] + 2\mu_{t,SGS}\overline{S}_{ij}\overline{S}_{ij} - \rho C_\varepsilon \frac{k_{SGS}^{1/2}}{\Delta} \tag{2.32}$$

where, $\Delta = (\Delta x \Delta y \Delta z)^{1/3}$, C_k and C_ε correspond to 0.1 and 1.0, respectively.

2.6. DETACHED EDDY SIMULATION

The Detached Eddy Simulation (DES) is a turbulence model proposed by Spalart

et al. (1997). It can be classified as a hybrid of the RANS and the LES model. This formulation is well suited for simulation of complex flows, including flow separation regions and the remainder of the flow region. Many researchers focused their efforts on application of the LES and the DNS as alternative prediction methods for the complex unsteady flows. It is well known that these methods are computationally very demanding. Due to the presence of near-wall turbulent boundary layers with small eddies (whose size is much less than the boundary layer thickness) the computational demands of the LES exceed the available computing power by a few orders of magnitude. As a result, there is currently no real prospect of using the LES in complex engineering computations and the DES is regarded as a powerful alternative to the LES.

The original DES formulation is based on the Spalart-Allmaras model (S-A model) (Spalart and Allmaras, 1994) in the near solid surfaces and the LES away from the wall. The S-A model is an additional transport equation to determine the eddy viscosity

$$\frac{D\tilde{\mu}}{Dt}+u_i\frac{\partial\tilde{\mu}}{\partial x_i}=c_{b1}\tilde{S}\tilde{\mu}+\frac{1}{\sigma}\left[\nabla\cdot((\mu+\tilde{\mu})\nabla\tilde{\mu})+c_{b2}(\nabla\tilde{\mu})^2\right]-(c_{w1}f_w)\left[\frac{\tilde{\mu}}{d}\right]^2 \qquad (2.33)$$

where $\tilde{\mu}$ is the working variable. The last term of the right-hand side of Eq. (2.33) represents the dissipation of the turbulent kinetic energy in the near wall region. Owing to its physical nature, the dissipation term in the RANS model is directly related to the reciprocal of the minimal wall distance *d*. The eddy viscosity is obtained from

$$\mu_t = \tilde{\mu} f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi \equiv \frac{\tilde{\mu}}{\mu} \tag{2.34}$$

where μ is the molecular viscosity. The production term is expressed as

$$\tilde{S} \equiv S + \frac{\tilde{\mu}}{\kappa^2 d^2} f_{v2} \tag{2.35}$$

where S is the magnitude of the vorticity. The remaining terms in Eqs. (2.33) through (2.35) are

$$f_{v1} = \left(1 + \frac{\chi}{c_{v2}}\right)^{-3}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad f_w = g\left[\frac{1 + c_{w3}^6}{g + c_{w3}^6}\right]^{1/6} \tag{2.36}$$

$$g = r + c_{w2}(r^6 - r), \quad r \equiv \frac{\tilde{\mu}}{\tilde{S}\kappa^2 d^2} \tag{2.37}$$

The constants appearing in the above equations are: $\kappa = 0.41$, $\sigma = 2/3$, $c_{b1} = 0.1355$, $c_{b2} = 0.622$, $c_{v1} = 7.1$, $c_{w1} = c_{b1}/\kappa^2 + (1 + c_{b2})/\sigma$, $c_{w2} = 0.3$, and $c_{w3} = 2$. This is the standard RANS formulation of the S-A model. Under the condition of local equilibrium, the production term in Eq. (2.33) is balanced by the dissipation term. The local balance leads to the relation $\tilde{\mu} \sim \tilde{S}d^2$, which is similar to the relation given by the Smagorinsky sub-grid model used for the LES if the wall distance d is replaced by the filter width Δ. In this formulation the length parameter, the distance to the nearest wall, d, by $\tilde{d}$ is defined as,

$$\tilde{d} \equiv \min(d, C_{DES}\Delta) \tag{2.38}$$

Equation (2.38) leads to a uniform model for the RANS and the LES model of the DES approach. The length scale $\tilde{d} = C_{DES}\Delta$ in the LES region yields a Smagorinsky eddy viscosity $\tilde{\nu} \propto S\Delta^2$. This makes the pseudo-Kolmogorov length scale, based on the eddy viscosity, proportional to the grid. Shur *et al.* (1999) recommended $C_{DES} = 0.65$ in homogeneous turbulence. The main advantages of the DES are lower demands on computational resources when compared with the LES, and more reliable to prediction of

separated flows, when compared with the RANS (Breuer *et al.*, 2003).

Many researchers have applied the concept of the DES to various flow cases. These attempts are desirable to predict flows where the S-A model can potentially fail to predict separation accurately enough. Although the DES formulation is built on the basis of the S-A model, the LES/S-A link is not fundamental and other models can be incorporated into the DES (Strelets, 2001).

CHAPTER 3. NUMERICAL METHODS

In this chapter, the numerical methods employed in this study are described. A steady state SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm (Patankar, 1980) is summarized and it is transformed into the unsteady state algorithm. The three-time level implicit method based on the SIMPLE is employed for time integration. The governing equations are discretized using Finite Volume Method (FVM).

3.1. GOVERNING EQUATIONS

The governing equations of incompressible flow are the conservation of mass, momentum and turbulent kinetic energy. The continuity equation is considered in a conservation form

$$\int_S \left(\rho \vec{U}\right) \cdot \vec{n} dS = 0 \tag{3.1}$$

where, ρ is mass density, and $\vec{U} = u\vec{i} + v\vec{j} + w\vec{k}$ is the flow velocity vector. Integration

is taken over the control volume and all directions. Figure 3.1 shows a control volume and the notation used, for a Cartesian 3D grid.

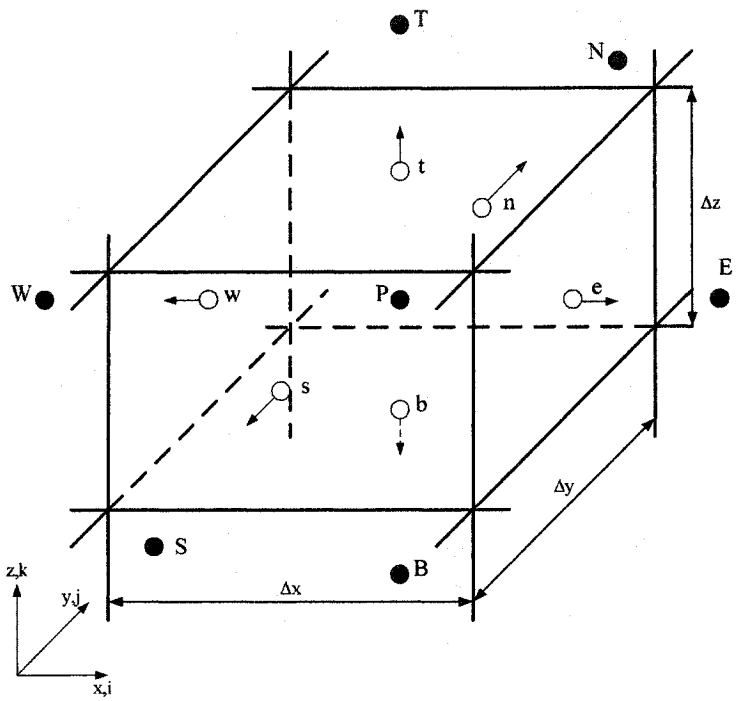


Figure 3.1 Control Volume and Notation, Cartesian 3D Grid

The momentum equation in the integral form is written as

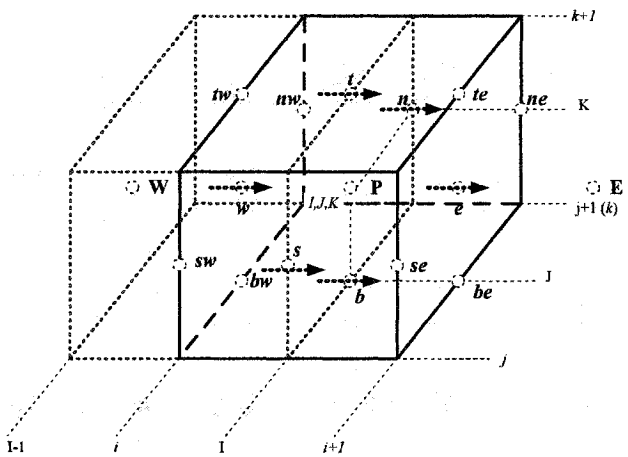
$$\frac{\partial}{\partial t}\int_{CV}(\rho u_i)dV + \int_S (\rho u_i)\vec{U}\cdot\vec{n}dS = -\int_S p\vec{i}_i\cdot\vec{n}dS + \int_S \tau_{ij}\vec{i}_j\cdot\vec{n}dS \qquad (3.2)$$

where $\vec{n} = n_x\vec{i} + n_y\vec{j} + n_z\vec{k}$ is a unit vector of the control volume face and $\vec{i}_i$ is a unit normal vector in i-th direction. The subscript i varies from 1 through 3, depending on the

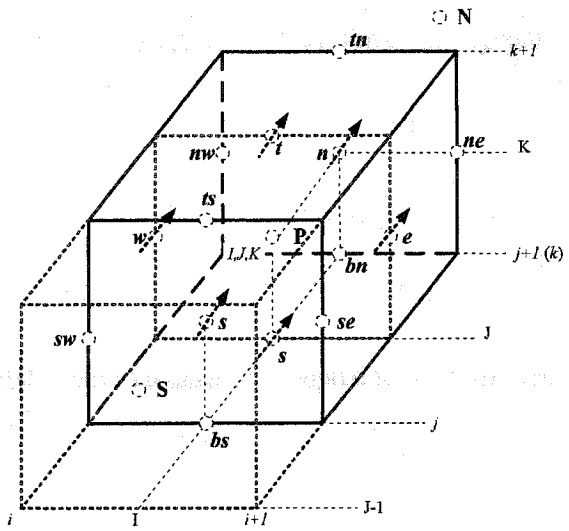
velocity directions - x, y, z coordinates. Body forces are ignored in this study. The discretization of Eqs. (3.1) and (3.2) is discussed in the next section.

3.2. DISCRETIZATION OF GOVERNING EQUATIONS

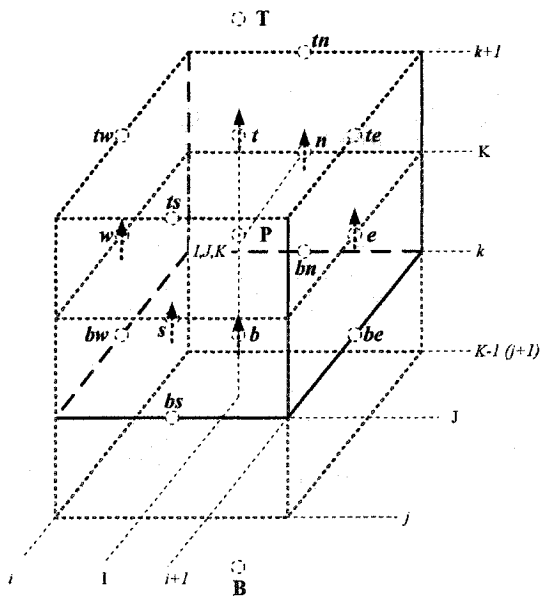
The staggered control volumes are adopted to discretize the governing equation using the finite volume method. Figure 3.2 shows the staggered grids of x, y and z momentum.



(a) Staggered Grid for x-Momentum



(b) Staggered Grid for y-Momentum



(c) Staggered Grid for z-Momentum

Figure 3.2 Control Volume of Staggered Momentum Grids

After rearranging, Eq (3.2) becomes

$$\frac{\partial}{\partial t}\int_{CV}(\rho u_i)\,dV = -\int_S p\vec{i}_i\cdot\vec{n}dS - \int_S(\rho u_i)\vec{U}\cdot\vec{n}dS + \int_S \tau_{ij}\vec{i}_j\cdot\vec{n}dS \tag{3.3}$$

The generalized momentum equations are discretized in *x*-, *y*-, and *z*-directions. Hereafter, the discretization in the x-direction is outlined. The procedure and expressions for the remaining (y, z) directions are similar. An implicit Euler method is used to solve Eq. (3.3).

After discretization for a node (i, j, k), Eq. (3.3) becomes

$$\left(\frac{\rho u^{n+1} - \rho u^n}{\Delta t}\right)_{ijk} Vol_{ijk} = -(pn_i)_\phi S_\phi - \sum_{\phi=1}^{faces}\left(F_\phi^{conv}\right)^{n+1} S_\phi + \sum_{\phi=1}^{faces}\left(F_\phi^{diff}\right)^{n+1} S_\phi \tag{3.4}$$

where, $n+1$ is the current time level, n is the previous time level, and Δt is the time step. Also, Vol_{ijk} is the volume of cell (i, j, k), and S_ϕ is the area of each cell face ϕ (ϕ= East, West, North, South, Top, and Bottom faces, for the three dimensional case). The convective flux in the *x*-direction at each cell face ϕ is

$$F_\phi^{conv} = F_\phi^{conv} S_\phi = \left(\rho u_\phi \vec{U}_\phi\cdot\vec{n}_\phi\right)S_\phi = \dot{m}_\phi u_\phi \tag{3.5}$$

Here, $\dot{m}_\phi$ is mass flux through the face ϕ. The diffusive flux for the u-momentum at the face 'e' becomes

$$F^{diff}_{i+\frac{1}{2},j,k} = \frac{\mu}{\Delta x} S_\phi (u_{i+1,j,k} - u_{ijk}) \tag{3.6}$$

where, Δx is the distance between the nodes (i, j, k) and $(i+1, j, k)$. In the implicit solution, the flux is linearized by using the hybrid scheme as

$$F^{n+1}_{i+\frac{1}{2},j,k}(u_{ijk}, u_{i+1,jk}, p_{i+\frac{1}{2},jk}) = F^n_{i+\frac{1}{2},jk} + \frac{\partial F^n_{i+\frac{1}{2},jk}}{\partial t}\Delta t = F^n_{i+\frac{1}{2},jk} +$$
$$\frac{\partial F^n_{i+\frac{1}{2},jk}}{\partial u_{ijk}}\Delta u_{ijk} + \frac{\partial F^n_{i+\frac{1}{2},jk}}{\partial u_{i+1,jk}}\Delta u_{i+1,jk} + \frac{\partial F^n_{i+\frac{1}{2},jk}}{\partial p_{i+\frac{1}{2},jk}}\Delta p_{i+\frac{1}{2},jk} =$$
$$F^{n+1}_{i+\frac{1}{2},j,k}(u_{ijk}, u_{i+1,jk}, p_{i+\frac{1}{2},jk}) + \Delta p_{i+\frac{1}{2},jk} n_{x_{i+\frac{1}{2},jk}} S_{i+\frac{1}{2},jk}$$
$$\left(\max\left[\dot{m}_{i+\frac{1}{2},jk}, \left(\mu\frac{u_{i+1,jk}}{\Delta x} + \frac{\dot{m}_{i+\frac{1}{2},jk}}{2}\right), 0\right]\right)\Delta u_{ijk} -$$
$$\left(\max\left[-\dot{m}_{i+\frac{1}{2},jk}, \left(\mu\frac{u_{i+1,jk}}{\Delta x} - \frac{\dot{m}_{i+\frac{1}{2},jk}}{2}\right), 0\right]\right)\Delta u_{i+1,jk} \tag{3.7}$$

where, the index $(i+1/2, j, k)$ refers to the wall between nodes (i, j, k) and $(i+1, j, k)$. The hybrid differencing scheme of Spalding (1972) is employed to control

accuracy of differencing scheme. If Peclet number (the ratio of convection to diffusion) is small, the central differencing scheme is employed. The upwind scheme is employed for large Peclet number. The mass flow $\dot{m}_{i+\frac{1}{2},jk}$ is computed as an average between the nodes and it is kept constant in the linearization. After integration at every cell face, Eq. (3.4) becomes

$$\left(\frac{\rho V_{ijk}}{\Delta t}\right)_{ijk} \Delta u_{ijk} = F^{n+1}_{i-\frac{1}{2},jk} + F^{n+1}_{i+\frac{1}{2},jk} + F^{n+1}_{i,j-\frac{1}{2},k} + F^{n+1}_{i,j+\frac{1}{2},k} + F^{n+1}_{ij,k-\frac{1}{2}} + F^{n+1}_{ij,k+\frac{1}{2}} + Source \tag{3.8}$$

where, the source term includes the pressure term. Linearized terms of Eq. (3.7) are inserted into Eq. (3.8). Afterwards, the diagonal coefficient becomes

$$\begin{aligned} A_{P,ijk} = {} & \frac{\rho V}{\Delta t} - (A_E + A_W + A_N + A_S + A_T + A_B)_{ijk} \\ & - \left[\left(\dot{m}_{i+\frac{1}{2},jk} - \dot{m}_{i-\frac{1}{2},jk} \right) + \left(\dot{m}_{i,j+\frac{1}{2},k} - \dot{m}_{i,j-\frac{1}{2},k} \right) + \left(\dot{m}_{ij,k+\frac{1}{2}} - \dot{m}_{ij,k-\frac{1}{2}} \right) \right] \end{aligned} \tag{3.9}$$

The mass residual, the last term in Eq. (3.9), is ignored. The remaining coefficients in Eq. (3.9) are as follows:

$$
\begin{aligned}
A_E &= \left(\max\left[-\dot{m}_{i+\frac{1}{2},jk}, \left(\mu \frac{u_{i+\frac{1}{2},jk}}{\Delta x} - \frac{\dot{m}_{i+\frac{1}{2},jk}}{2} \right), 0 \right] \right) \\
A_W &= \left(\max\left[\dot{m}_{i-\frac{1}{2},jk}, \left(\mu \frac{u_{i+\frac{1}{2},jk}}{\Delta x} + \frac{\dot{m}_{i-\frac{1}{2},jk}}{2} \right), 0 \right] \right) \\
A_N &= \left(\max\left[-\dot{m}_{i,j+\frac{1}{2},k}, \left(\mu \frac{u_{i,j+\frac{1}{2},k}}{\Delta y} - \frac{\dot{m}_{i,j,+\frac{1}{2},k}}{2} \right), 0 \right] \right) \\
A_S &= \left(\max\left[\dot{m}_{i,j-\frac{1}{2},k}, \left(\mu \frac{u_{i,j-\frac{1}{2},k}}{\Delta y} + \frac{\dot{m}_{i-\frac{1}{2},jk}}{2} \right), 0 \right] \right) \\
A_T &= \left(\max\left[-\dot{m}_{ij,k+\frac{1}{2}}, \left(\mu \frac{u_{ij,k+\frac{1}{2}}}{\Delta z} - \frac{\dot{m}_{ij,k+\frac{1}{2}}}{2} \right), 0 \right] \right) \\
A_B &= \left(\max\left[\dot{m}_{ij,k-\frac{1}{2}}, \left(\mu \frac{u_{ij,k-\frac{1}{2}}}{\Delta z} + \frac{\dot{m}_{ij,k-\frac{1}{2}}}{2} \right), 0 \right] \right)
\end{aligned}
\tag{3.10}
$$

Therefore, Eq. (3.3) can now be written as

$$
\begin{aligned}
&A_P \Delta u_{ijk} + A_E \Delta u_{i+1,jk} + A_W \Delta u_{i-1,jk} + A_N \Delta u_{i,j+1,k} + A_S \Delta u_{i,j-1,k} + A_T \Delta u_{ij,k+1} + A_B \Delta u_{ij,k-1} \\
&= -\Delta p_\phi n_{x,\phi} S_\phi + \left(F^n_{i+\frac{1}{2},jk} + F^n_{i-\frac{1}{2},jk} + F^n_{i,j+\frac{1}{2},k} + F^n_{i,j-\frac{1}{2},k} + F^n_{ij,k+\frac{1}{2}} + + F^n_{ij,k-\frac{1}{2}} \right)
\end{aligned}
\tag{3.11}
$$

3.3. PRESSURE-CORRECTION METHOD

The pressure-correction approach is employed to solve the momentum equation,

using the SIMPLE algorithm (Patankar, 1980). When the steady state is reached, the last term of Eq. (3.11) is zero. The linearized increments Δu and Δp are replaced in the following by iterative corrections u' and p'

$$A_P\Delta u'_{ijk} + A_E\Delta u'_{i+1,jk} + A_W\Delta u'_{i-1,jk} + A_N\Delta u'_{i,j+1,k} + A_S\Delta u'_{i,j-1,k} + A_T\Delta u'_{ij,k+1} + A_B\Delta u'_{ij,k-1} = -p'_\phi n_{x,\phi} S_\phi \tag{3.12}$$

The SIMPLE algorithm is applied to drop the non-diagonal terms from Eq. (3.12). The derivation of Eq. (3.12) can be done in a shifted control volume to obtain u' at the cell face

$$A_{P,\phi}\Delta u'_{ijk} = -(p'n_xS)_{i+1,jk} + (p'n_xS)_{ijk} - (p'n_xS)_{i+\frac{1}{2},j+\frac{1}{2},k} - (p'n_xS)_{i+\frac{1}{2},j-\frac{1}{2},k} - (p'n_xS)_{i+\frac{1}{2},j,k+\frac{1}{2}} - (p'n_xS)_{i+\frac{1}{2},j,k-\frac{1}{2}} \tag{3.13}$$

The coefficients $A_{P,\phi}$ is similar for all other components at the same cell face ϕ. Equation (3.13) can be written for v' and w' by replacing the cell face normal n_x with n_y and n_z respectively. The continuity equation states that

$$\sum_{\phi=1}^{faces} \rho(\vec{U}^* + \vec{U}')_\phi \cdot \vec{n}_\phi S_\phi = 0$$

$$\sum_{\phi=1}^{faces} (u' n_x S)_\phi + (v' n_y S)_\phi + (w' n_z S)_\phi = -\sum_{\phi=1}^{faces} (\rho \vec{U}^* \cdot \vec{n} S)_\phi \tag{3.14}$$

where, $\vec{U}^*$ is a velocity field. It does not obey the mass balance and $\vec{U}'$ is obtained as the iterative correction. After substituting u', v' and w' from Eq. (3.13) into Eq. (3.14), $\vec{U}' \cdot \vec{n} S$ at the face $S_{i+\frac{1}{2},jk}$ becomes

$$\frac{\left[-(((Sn_x)^2 + (Sn_y)^2 + (Sn_z)^2)p')_{i+1,jk} + ((Sn_x)^2 + (Sn_y)^2 + (Sn_z)^2)p')_{ijk}\right]}{A_{P,i+\frac{1}{2},jk}} \tag{3.15}$$

After computing terms at all faces and collecting, the following equation is obtained

$$\begin{aligned}
&\frac{1}{A_{P,i+\frac{1}{2},jk}}\left[-(S^2 p')_{i+1,jk} + (S^2 p')_{ijk}\right] + \frac{1}{A_{P,i-\frac{1}{2},jk}}\left[-(S^2 p')_{i-1,jk} + (S^2 p')_{ijk}\right] \\
&+ \frac{1}{A_{P,ij+\frac{1}{2},k}}\left[-(S^2 p')_{i,j+1,k} + (S^2 p')_{ijk}\right] + \frac{1}{A_{P,ij-\frac{1}{2},k}}\left[-(S^2 p')_{i,j-1,k} + (S^2 p')_{ijk}\right] \\
&+ \frac{1}{A_{P,ijk+\frac{1}{2}}}\left[-(S^2 p')_{ij,k+1} + (S^2 p')_{ijk}\right] + \frac{1}{A_{P,ijk-\frac{1}{2}}}\left[-(S^2 p')_{ij,k-1} + (S^2 p')_{ijk}\right] \\
&= -\Delta \dot{m}_{ijk}
\end{aligned} \tag{3.16}$$

The surface areas S_ϕ are approximated as averages and after some regrouping of the

terms, the pressure correction equation is finally obtained

$$B_P p'_{ijk} + B_E p'_{i+1,jk} + B_W p'_{i-1,jk} + B_N p'_{i,j+1,k} + B_S p'_{i,j-1,k} + B_T p'_{ij,k+1} + B_B p'_{ij,k-1} = -\Delta \dot{m}_{ijk} \qquad (3.17)$$

where the coefficients are defined as

$$B_E = \frac{S^2_{i+\frac{1}{2},jk}}{A_{P,i+\frac{1}{2},jk}} \qquad B_W = \frac{S^2_{i-\frac{1}{2},jk}}{A_{P,i-\frac{1}{2},jk}}$$

$$B_N = \frac{S^2_{i,j+\frac{1}{2},k}}{A_{P,i,j+\frac{1}{2},k}} \qquad B_S = \frac{S^2_{i,j-\frac{1}{2},k}}{A_{P,i,j-\frac{1}{2},k}} \qquad (3.18)$$

$$B_T = \frac{S^2_{ij,k+\frac{1}{2}}}{A_{P,ij,k+\frac{1}{2}}} \qquad B_B = \frac{S^2_{ij,k-\frac{1}{2}}}{A_{P,ij,k-\frac{1}{2}}}$$

$$B_P = -(B_E + B_W + B_N + B_S + B_T + B_B)$$

3.4. TIME DISCRETIZATION

A general multi-step time-integration method is defined by the following equation

$$(1+\gamma)(\rho\vec{U})^{n+1}-(1+2\gamma)(\rho\vec{U})^{n}+\gamma(\rho\vec{U})^{n-1} = \frac{\Delta t}{V_{ijk}}\left[(1-\beta)\vec{R}^{n}+\beta\vec{R}^{n+1}\right] \tag{3.19}$$

where, residual $\vec{R}$ includes the convective, the diffusive and the pressure terms. With different choices of parameters β and γ, the following time advance schemes can be obtained: $\beta=\gamma=0.0$ for an explicit Euler scheme, $\beta=1.0,\ \gamma=0.0$ for an implicit Euler scheme, $\beta=0.5,\ \gamma=0.0$ for a Crank-Nicolson scheme, and $\beta=1.0,\ \gamma=0.5$ for a three-time-level implicit scheme. The three-time-level implicit scheme is less prone to producing oscillatory solution with large values of Δt and it is second-order accurate. After applying the three-time-level implicit scheme, Eq. (3.19) is linearized between the solved state n and the state n+1 to be computed

$$(1+\gamma)\left[\rho(\vec{U}^{k}+\vec{U}^{n+1}-\vec{U}^{k})\right]-(1+2\gamma)\rho\vec{U}^{n}+\gamma\rho\vec{U}^{n-1} = \frac{\Delta t}{V_{ijk}}\left\{(1-\beta)\vec{R}^{n}+\beta\left[\vec{R}^{k}+\left(\frac{\partial\vec{R}}{\partial\vec{U}}\right)^{k}(\vec{U}^{n+1}-\vec{U}^{k})\right]\right\} \tag{3.20}$$

Rearrangement of terms in Eq. (3.20) leads to the following equation

$$\left[1+\gamma-\beta\frac{\Delta t}{V_{ijk}}\left(\frac{\partial \vec{R}}{\partial \vec{U}}\right)^{k}\right]\Delta\vec{U}=$$
$$\frac{1}{U^{n+1}}\left\{-(1+\gamma)\vec{U}^{k}+(1+2\gamma)\vec{U}^{n}-\gamma\vec{U}^{n-1}+\Delta t\left[(1-\beta)\vec{R}^{n}+\beta\vec{R}^{k}\right]\right\} \tag{3.21}$$

where, $\Delta\vec{U}=\vec{U}^{n+1}-\vec{U}^{k}$ (the superscript k is an intermediate level between n and n+1 time levels). $\vec{U}^{n+1}$ is iterated and it must be stabilized for arbitrary time steps.

The computation starts with an initial guess value which is also copied at the previous time level. The momentum equations are solved sequentially. The residual $\vec{R}^{n}$ is computed and the time derivative term is added to the modified residual. The mass balance is computed and is used as a source for the pressure correction equation. If the residual of u, v, w, p or mass balance is greater than some pre-defined convergence criteria, the solution is iterated within the time step. As the convergence is reached, the results are written and the iteration of the solution of the new time level is started by updating the variables at the old and present time levels.

3.5. DETERMINATION OF A TIME STEP

An appropriate time step must be selected to ensure convergence and accuracy of a numerical simulation. This determination can be accomplished by considering the Courant-Friedrichs-Lewy (CFL) condition based on the Courant number

$$CFL = \Delta t \max\left[\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z}\right] \tag{3.22}$$

The Courant number is the ratio of time step Δt to the characteristic convection time, $|u_i|/\Delta x_i$, the time required for a disturbance to be convected a distance Δx_i. The Courant number should be smaller than 1.0. This procedure was used in the present study. Before any averaging is performed, a simulation has to be run for a sufficiently long time period, so that the turbulent structures have fully developed (Jacobsen, 1997). The simulations described herein were carried out until this condition was reached and subsequent results were used to calculate statistical characteristics of the generated flow and the wind-induced loading.

CHAPTER 4. BOUNDARY CONDITIONS

The present chapter presents an overview of boundary conditions used in three-dimensional numerical simulations of wind past and wind-induced loading on buildings and structures. All CWE problems require that appropriate boundary conditions be defined. This requirement is an important part of a numerical algorithm and it is mandatory to ensure that correct results are obtained and that computational efficiency is achieved. Overall, four categories of boundary conditions are needed in numerical simulations for wind engineering applications.

4.1. INFLOW BOUNDARY CONDITION

4.1.1. Steady-State Simulation

The distribution of all flow parameters needs to be specified at inflow boundaries. In computational wind engineering, usually the mean velocity profile is obtained from the experimental data or it is determined using empirical models, such as the logarithmic law (Simiu and Scanlan, 1996)

$$\bar{U}(z) = \frac{1}{\kappa} u_* \ln \frac{z}{z_0} \tag{4.1}$$

where, κ is the von Kármán constant, z_0 is the aerodynamic roughness length, and u_* is the friction velocity. Alternatively, the power law is frequently used in wind engineering applications (ASCE 7-98, 1998) to define the mean velocity profile

$$\bar{U}(z) = U_{ref} \left(\frac{z}{z_{ref}} \right)^{\alpha} \tag{4.2}$$

where, the subscript *ref* denotes the reference values (mean wind speed U and elevation z) and α is an exponent index which is dependent on the surface roughness. In the most cases of CWE simulations, the mean wind velocity profile is obtained from experimental data, using Eq. (4.1) or Eq. (4.2). The inflow turbulent kinetic energy, k, is estimated from the turbulence intensity profile, obtained from the experimental data

$$I(z) = \frac{\sqrt{\overline{u'^2}}}{\bar{U}(z)}$$
$$k = \frac{3}{2}\overline{u'^2} \tag{4.3}$$

4.1.2. Unsteady Simulation

Three velocity components, as functions of space and time - are required to simulate the unsteady flows. As a result, the initial velocity fluctuations have to be provided. The stochastic components of the initial velocity can be obtained as a superposition of the mean velocity and random perturbations

$$u_i = \bar{U}_i(1 + TI_i f) \tag{4.4}$$

where, TI is the turbulence intensity and $f(random)$ is the Gaussian random number generator. Selvam (1997) produced the initial velocity fluctuations using a random numbers having the Gaussian distribution and considered a logarithmic mean velocity profile determined from the experimental data. Finally, adjusted random velocity fluctuations were superimposed on the mean velocity profile. More refined methods were reported by other researchers. They included re-using of the LES or the DNS numerical data obtained in simulation of turbulent boundary flows past rough plate (Murakami *et al*, 1992) and computer simulation of divergence-free inflows of prescribed profiles of mean velocity and power spectra of velocity fluctuations (Kondo *et al.* 1997). They generated the inflow turbulence for the LES using the method of Hoshiya (1972) based on Monte

Carlo simulation, considering the power spectral matrix that included the power spectra and the cross-spectra. These advanced approaches require significant computational effort and they lead to significant increase in computational costs, CPU-time and memory requirements.

4.2. WALL BOUNDARY CONDITION

In numerical simulation, the momentum flux through the boundary must be known and the flow conditions at solid boundaries must be specified. The wall non-slip condition allows the determination of the convective part of the $\overline{u_i u_j}$ momentum flux, at the wall. In boundary layer type flows, the viscous wall region is very important, the production, dissipation, kinetic turbulence energy, and Reynolds-stress anisotropy all achieve their peak values at locations very near the wall.

The implementation of the wall boundary conditions in turbulent flows starts with the evaluation of the non-dimensional distance from the wall

$$z^+ = \frac{\rho z u_*}{\mu} \tag{4.5}$$

To accurately represent the structures in the near-wall region, the first grid point must be

located at $z^+ \leq 20$. As $Re \to \infty$, an increasing number of grid points must be used to resolve the near-wall layer of the decreasing thickness. When the grid spacing is not fine enough to resolve the near-wall gradients, the momentum flux near the wall cannot be evaluated directly by numerical differentiation and the wall layer must be modeled by specifying a relationship between the velocity in the outer flow and the stress at the wall.

4.2.1. Wall Function

Numerical simulations of three-dimensional flows with very fine grid spacing in the near-wall regions may lead to excessive demands on computer resources. A commonly used solution in such cases is the application of the law of wall in the near-wall regions. This assumption postulates that the flow near wall behaves like a fully developed turbulent boundary layer. As a consequence, the boundary conditions in these regions take a form denoted as wall functions.

In the fully developed turbulent boundary layer, the turbulence production term and the dissipation term, in the log-law layer ($30 < z^+ < 100$), are much larger than the remaining terms. The log-law mean velocity profile can be written as

$$\frac{\bar{U}}{u_*} = \frac{1}{\kappa} \ln(z^+ E) \tag{4.6}$$

where, E = 9.794. In the log-law region we can write the modeled turbulent kinetic energy equation, the k equation, as follows

$$P_k - \rho\frac{k^{3/2}}{l} = 0 \tag{4.7}$$

In addition, in the log-law region the shear stress is equal to the wall shear stress τ_w. According to the Boussinesq assumption, the shear stress reads

$$\tau_w = -\rho\overline{uw} = \mu_t\frac{\partial\bar{U}}{\partial z} \tag{4.8}$$

Using the definition of the wall shear stress $\tau_w = \rho u_*^2$ and inserting Eq. (4.8), the definition of turbulent viscosity in the one-equation ($\mu_t = \rho C_\mu k^{1/2} l$) into Eq. (4.7), we get

$$C_\mu = \left(\frac{u_*^2}{k}\right)^2 \tag{4.9}$$

From experiments, u_*^2/k is approximately equal to 0.3 in the log-law region of a boundary layer, so that C_μ is equal to 0.09. For the velocity component parallel to the

wall, the wall shear stress is used as a flux boundary condition. When we are using wall functions, k is not solved at the nodes adjacent to the walls. Instead, the turbulent kinetic energy is determined from Eq. (4.9), i.e.

$$k_p = C_\mu^{-1/2} u_*^2 \tag{4.10}$$

where, the friction velocity u_* is obtained iteratively from the log-law. Index p denotes the first interior node adjacent to the wall.

4.3. OUTFLOW BOUNDARY CONDITION

An unsteady convective condition is used at the outflow boundary

$$\frac{\partial U_i}{\partial t} - U_{exit} \frac{\partial U_i}{\partial x} = 0 \tag{4.11}$$

where U_{exit} is the mean exit velocity that is independent of location on the outflow surface and is required to make the outflow mass flux equal to the inflow mass flux. The exit boundary condition must allow pressure waves to smoothly leave the computational domain. Alternative exit boundary conditions were tested by various researchers, e.g. Pauley (1988).

4.4. LATERAL BOUNDARY CONDITION

To apply periodic boundary conditions, we need to set the flux of all flow variables leaving the north boundary equal to the flux entering the south boundary. This is achieved by equating the values of each variable at the nodes in the south boundary plane to the nodal values in the north boundary plane.

4.5. CALCULATION OF STATISTICAL PROPERTIES OF VELOCITY COMPONENTS

In order to compute the mean values from the unsteady numerical simulation, it is necessary to time-average the velocity components. The mean and root mean square fluctuation values are computed at each time step as follows

$$\bar{U}_i = \frac{\sum_{tstart}^{t} U_i}{t - t_{start}}$$

$$\overline{u_i'^2} = \sqrt{\frac{\sum_{tstart}^{t} \left(U_i - \bar{U}_i\right)^2}{t - t_{start}}} \qquad (4.12)$$

The time averaging process starts after a designated number of time steps, t_{start}, to allow

the solution to develop, and generally the longer the time averaging process the more accurate the calculated mean and RMS values. Equation (4.12) was also possible to calculate transient and time averaged Reynolds stress data.

CHAPTER 5. HYBRID TL/LES MODEL

5.1. INTRODUCTION

The hybrid TL/LES model - a combination of the two-layer model (Rodi, 1988) and the LES model – is proposed in this section. The RANS kinetic energy equation of the two-layer model is used to close the RANS equations, while the filtered sub-grid kinetic energy (k_{SGS}) is used to close the LES model. The near-wall layer of the viscosity-affected region is resolved by the two-layer model. In the two-layer model, the RANS turbulent kinetic energy k is determined from the modeled k transport equation while its dissipation rate ε is determined from a prescribed length-scale distribution l. The outer region is resolved by the sub-grid one-equation model of Yoshizawa (1993). For the eddy viscosity, the RANS eddy viscosity ($\mu_{t,RANS}$) is solved by the two-layer model in the RANS region and the sub-grid viscosity ($\mu_{t,SGS}$) is solved by Yoshizawa's model in the LES region.

The hybrid TL/LES model involves matching between the two-layer model and the LES model. The matching occurs at some location near the edge of the viscous sub-

layer; i.e., in a region where viscous effects are negligible. The matching line between the RANS and the LES regions is computed from the length-scale distribution.

5.2. THE GOVERNING EQUATIONS

The Navier-Stokes equations and the continuity equation, time-averaged in the near wall regions and filtered in the outer regions, are rewritten as

$$\frac{\partial(\rho\overline{U}_i)}{\partial x_i} = 0$$
$$\frac{\partial(\rho\overline{U}_i)}{\partial t} + \frac{\partial(\rho\overline{U}_i\overline{U}_j)}{\partial x_j} = -\frac{\partial\overline{P}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial\overline{U}_i}{\partial x_j} + \frac{\partial\overline{U}_j}{\partial x_i}\right)\right] - \frac{\partial\tau_{ij}}{\partial x_i} \tag{5.1}$$

and the turbulent kinetic energy transport equation for the RANS and the LES regions are

$$\frac{\partial(\rho k_{RANS})}{\partial t} + \overline{U}_i\frac{\partial(\rho k_{RANS})}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_{t,RANS}}{\sigma_k}\right)\frac{\partial k_{RANS}}{\partial x_j}\right] + P_{k_{RANS}} - \rho\varepsilon_{RANS} \tag{5.2}$$

$$\frac{\partial(\rho k_{SGS})}{\partial t} + \overline{U}_i\frac{\partial(\rho k_{SGS})}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\mu + \mu_{t,SGS}\right)\frac{\partial k_{SGS}}{\partial x_j}\right] + P_{k_{SGS}} - \rho\varepsilon_{SGS} \tag{5.3}$$

Equation (5.2) is the two-layer model and Eq. (5.3) is the one-equation model of

Yoshizawa (1993). These models are based on the eddy viscosity hypothesis and thus the stress tensor τ_{ij} takes the form of

$$\tau_{ij} = -2\rho\mu_t \bar{S}_{ij}, \quad \bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{U}_i}{\partial x_j} + \frac{\partial \bar{U}_j}{\partial x_i}\right) \tag{5.4}$$

The bar ($\bar{}$) over the velocity components and pressure denotes time averaging in the RANS region and filtering in the LES region. $P_{k_{RANS(SGS)}}$ is the production term; $k_{RANS(SGS)}$ is the turbulence RANS (sub-grid) kinetic energy; $\rho\varepsilon_{RANS(SGS)}$ is the dissipation term, $\mu_t = \mu_{RANS}$ and τ_{ij} is the Reynolds stress tensor for $z \le z_{cr}$, otherwise $\mu_t = \mu_{SGS}$, and τ_{ij} is the sub-grid scale stress tensor. The switching distance z_{cr} represents the distance from the wall at which the RANS and the LES are linked. The definition of the RANS eddy viscosity in the two-layer model is following

$$\mu_{RANS} = \rho C_\mu \sqrt{k_{RANS}} \cdot l_\mu \tag{5.5}$$

whereas, the sub-grid eddy viscosity in the Yoshizawa's model is defined as

$$\mu_{SGS} = \rho C_s \sqrt{k_{SGS}} \cdot \Delta_{SGS} \tag{5.6}$$

where C_μ and C_s are the model constants and the length-scale respectively, Δ_{SGS}, is determined from the local grid volume $(\Delta_x \cdot \Delta_y \cdot \Delta_z)^{1/3}$ ($\Delta_x, \Delta_y, \Delta_z$ are grid sizes in the x, y, z directions, respectively) in the outer region. $\rho\varepsilon_{RANS}$ is expressed as

$$\rho\varepsilon_{RANS} = \rho\frac{k_{RANS}^{3/2}}{l_\varepsilon} \tag{5.7}$$

where l_μ and l_ε are the length-scales defined by Norris and Reynolds (1975), while $\rho\varepsilon_{SGS}$ becomes

$$\rho\varepsilon_{SGS} = \rho C_\varepsilon \frac{k_{SGS}^{3/2}}{\Delta_{SGS}} \tag{5.8}$$

5.3. DEFINING OF SWITCHING LINE

In the DES or hybrid method, the switching distance (z_{cr}) is required to blend μ_{RANS} with μ_{SGS}. At this distance where the damping function in the expression for l_μ, in Eq. (2.16), has a value close to 1.0 the viscous effects are small. Rodi (1988) tested the switching line of the two-layer model. In boundary layers, this switching criterion effectively led to a switching between the models at $z^+ = 80$ - 90. Another investigation of the switching criterion was performed by Chen and Patel (1988). They found that the

results were not sensitive to the switching criterion when the minimum Re_z was greater than 200. It should be noted that in normal boundary-layer flow, $Re_z = 250$ is roughly matching $z^+ = 135$. Iacovides and Launder (1987) reported the switching between the models in the z^+ region of 80 - 120. According to prior research, the switching distance z_{cr} of the hybrid TL/LES model should be in the range $80 \le z^+ \le 135$.

Figure 5.1 shows the initial distribution of the grid points of the computational domain for the hybrid TL/LES model. The first grid point is located at $z^+ = 7.0$ and 15 grid points are placed in the viscous sub-layer to resolve the viscous–dominant flow. This grid point distribution has been established based on the recommendation (15 - 20 points) by Lakehal and Rodi (1997). It can be seen that z^+ is equal to 106 when the local turbulence Reynolds number reaches 250. Such a distance appears to be a reasonable location for the switching line and it is comparable with the values used by other researchers. In view of the above, the initial switching criterion for simulations discussed in this dissertation is set to 106. The separation between the near wall region and the main flow region has to be subsequently computed on each wall boundary. During the computation, the boundary layer thickness changes as flow develops. The modeled $k_{RANS(SGS)}$ are allowed to be transported across the matching line by convection and diffusion. Constants and functions used in the hybrid TL/LES model are presented in

Table 5.1.

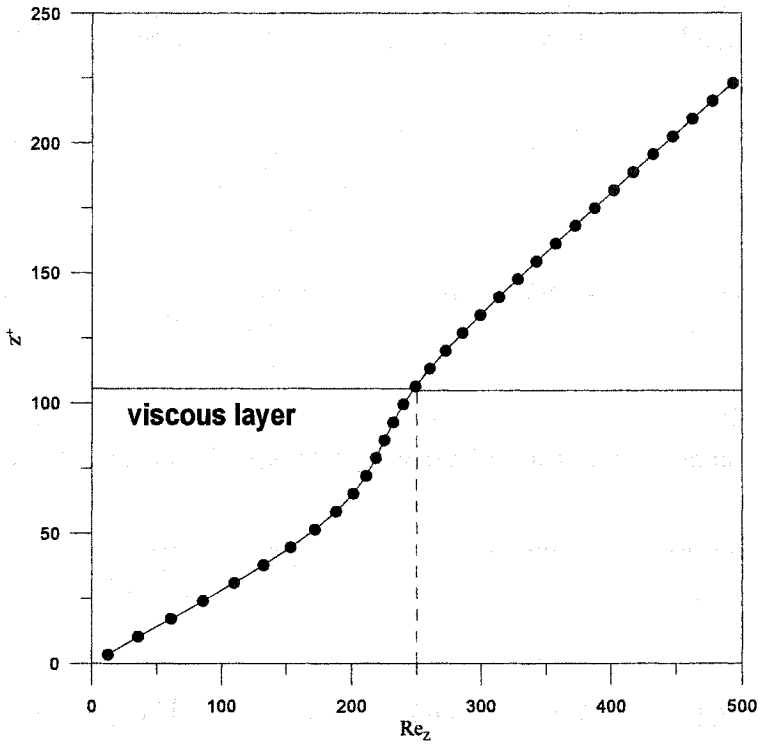


Figure 5.1 Grid Point Locations in Near Wall Region

Table 5.1 Constants and Functions Used in Hybrid TL/LES Model

Constants and Functions	Value and Formula	References and Comments
σ_k	1.0	Launder and Sharma (1974)
C_μ	0.09	Launder and Sharma (1974)
κ	0.41	von Kármán Constant

C_k	0.1	Yoshizawa (1993)
C_ε	1.0	Yoshizawa (1993)
A_μ	50.5	Rodi *et al.* (1988)
l_μ	$\kappa C_\mu^{-3/4} z\left[1.0-\exp\left(-\frac{R_z}{A_\mu}\right)\right]$	Norris and Reynolds (1975)
l_ε	$\frac{\kappa C_\mu z}{1.0+\frac{5.3}{R_z}}$	Norris and Reynolds (1975)
R_z	$\frac{\rho\sqrt{k}z}{\mu}$	Reynolds Number of Turbulence
z_{cr}	106	Switching Criterion

5.4. DESCRIPTION OF COMPUTER CODE

The hybrid TL/LES model has been incorporated in a computer code developed to simulate wind flow past and wind-induced loading on buildings and structures, and other bluff bodies. The code comprises of a main program and a number of blocks (subroutines). The program flow chart is schematically shown in Figure 5.2. A brief description of the main program and each subroutine is presented in Table 5.2. A complete listing of the program is provided in Appendix.

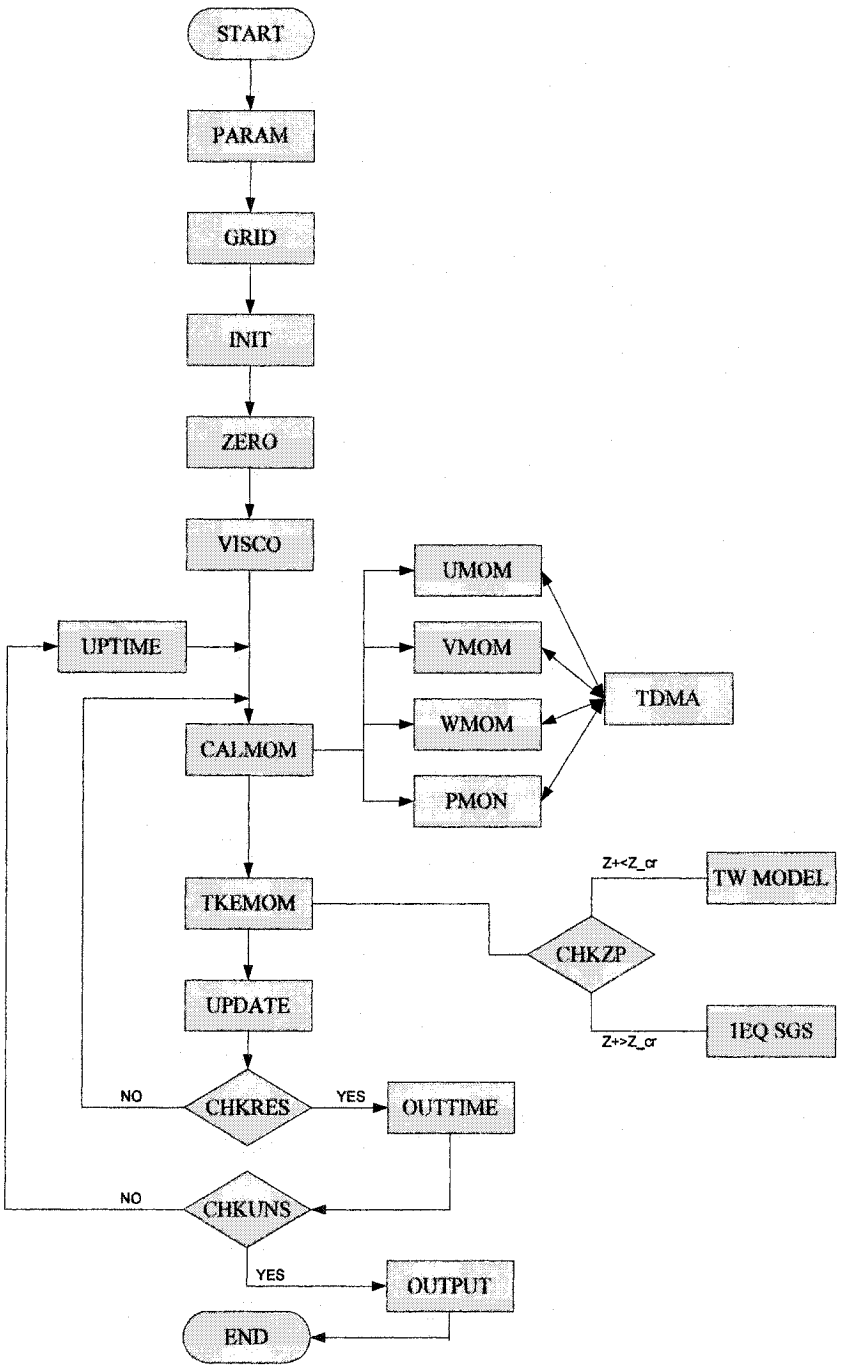


Figure 5.2 Flow Chart of Computer Program

Table 5.2 Description of Program Blocks (Subroutines)

PARAM	This subroutine defines all parameters and the size of array.
GRID	This subroutine computes the size and the number of grids of each direction.
INIT	This subroutine reads input parameters, grid data, the initial conditions, and the values of constants.
ZERO	This subroutine makes the whole computational domain set to zero.
VISCO	This subroutine computes the eddy viscosity in the computational domain and calculates the dimensionless distance from the wall. The eddy viscosity is computed by the hybrid relation of the eddy viscosity as mentioned in Section 5.4.
CALMOM	This subroutine discretizes and solves the momentum equations with three-time level method. In this subroutine the correction values are computed using the SIMPLE method. The current values are stored at time-step n in order to compute the values at advanced time-step n+1. Each subroutine of momentum equation is linked to TDMA subroutine to solve the matrix.
TKEMOM	This subroutine computes the turbulent kinetic energy equation.
CHKZP	This subroutine computes the switching line. Depending on the dimensionless distance, this subroutine is linked between the two-layer model and the SGS one-equation model, automatically.
UPDATE	This subroutine updates the each values of momentum and transport equation, and modifies the eddy viscosity.

CHKRES	This subroutine checks the residual error of each equation. If the residual error satisfies the criteria of error, the processing goes to the next processing. If it does not satisfy the criteria of error, the updated values are used as the initial values to compute the governing equation.
OUTTIME	This subroutine prints out the variables at the each time step.
CHKUNS	This subroutine checks the stabilization of the steady flow fields. This subroutine computes the errors between current time and old time. If the stabilization of the flow satisfies with the criteria of the steady state, the program will be terminated.
OUTPUT	This subroutine prints out the variables at the steady state.

CHAPTER 6. RESULTS OF NUMERICAL SIMULATIONS

This chapter presents the results of a series of numerical simulations of wind and wind-induced loading on bluff bodies, obtained using a developed program incorporating the hybrid TL/LES model described in this dissertation. The numerical experiments discussed herein include 3D calculations carried out for flow past a square cylinder, a surface-mounted cube, and the TTU test building.

First, numerical simulations of flow past a 3D square cylinder and a surface-mounted cube are presented. The detailed experimental data for the flows past a 3D square cylinder and a surface-mounted cube were reported by Durao *et al.* (1988), Lyn and Rodi (1994), Lyn *et al.* (1995), and Castro and Robins (1977). In the numerical simulations, Bosch and Rodi (1998) adopted a 2D ensemble-averaged unsteady Navier-Stokes equation combined with an ad-hoc model to simulate vortex shedding past a square cylinder. Their numerical results agreed well with experimental data. Second, modeling of turbulent flows past 2D square cylinder and 3D surface-mounted cube for different Reynolds numbers using various LES and RANS models was reported by Rodi

(1997). The presented results generally agreed with the experimental data. The application of the LES to turbulent flow around a square cylinder was reported by Murakami (1993), who employed the LES with the Smagorinsky SGS model. Lee and Bienkiewicz (1997) investigated 3D flow past a square cylinder, using the standard LES combined with the finite element method discretization. Johansen *et al.* (2004) carried out numerical experiments on a flow past a square cylinder using the filter-based unsteady k-ε model. These results exhibited a better performance of the filter-based k-ε model, when compared with the standard k-ε model. Application of the DES technique to flow past a square cylinder and a wall-mounted cube was reported by Schmidt and Thiele (2002). Their results showed that the DES was able to capture the most dominant flow patterns (similar to those reproduced using the LES), while the application of the RANS led to a poor representation of the unsteady flow phenomena.

The third section of this chapter presents the results of a numerical experiment carried out for the Texas Tech University test building. Wind engineering research at the Texas Tech University field site has stimulated various research groups to focus their research efforts on laboratory and numerical simulations of wind at the TTU site, and wind-induced loading on the TTU test building. Laboratory simulations were carried out by Surry (1991), Tieleman *et al.* (1996), Ham and Bienkiewicz (1998), Okada and Ha

(1992), and others. Numerical simulations included calculations of wind past and wind-induced pressures on the TTU test building, Paterson and Holmes (1992), Qasim *et al.* (1992), Mochida *et al.* (1993), Selvam (1997), and others. Most of the numerical simulations were based on steady-state calculations incorporating the standard k-ε model to account for the flow turbulence. Mochida *et al.* (1993) and Selvam (1997) performed the numerical simulation using the LES. The numerical results presented in this dissertation for the TTU test building were compared with the laboratory data obtained at the Wind Engineering and Fluid Laboratory (WEFL), Colorado State University.

6.1. FLOW PAST SQUARE CYLINDER

Figure 6.1 shows the geometry of the experimental set up of Lyn and Rodi (1994), which was selected for the first numerical experiment discussed in this dissertation. A uniform velocity was imposed as the inflow condition and the no-slip condition was enforced on the surface of the square cylinder. A periodic condition was imposed on both lateral sides. At the outflow boundary, the convective condition was imposed for the velocity components. The symmetry condition was imposed on the upper and lower sides. The center of square cylinder was 5H downstream of the inflow boundary and 20H upstream of the outflow boundary. The upper, lower, and both lateral

boundaries were at a distance of 7H from the cylinder center.

At the inflow boundary, the mean velocity was uniform and horizontal. The Reynolds number based on the mean approach velocity and the cylinder side was 22000. The inflow turbulent intensity was 2%, for all flow components. The kinetic energy of turbulence $k = 0$ was imposed on the surface of the square. The computational domain comprised of 70 × 20 × 50 grid points, respectively in the (x, y, z) directions. The dimensionless distance (z^+) from the cylinder surface to the nearest grid point was 7.0.

A summary of the parameters used in the presented numerical testing is provided in Table 6.1. Similar values of the simulation parameters were employed in numerical calculations reported during the Workshop on LES (1995).

Table 6.1 Summary of Simulation Parameters

Side of Square Cylinder (H)	0.04 m
Mean Velocity (3H Upstream)	0.535 m/s
Reynolds Number	22000
Turbulence Intensity	2 %
Grid Size (Horizontal × Lateral×Vertical)	70 × 20 × 50

Computational Domain	25H × 14H × 14H
Location of First Grid Point (Viscous Distance from Surface of Cylinder)	$z^+ = 7.0$

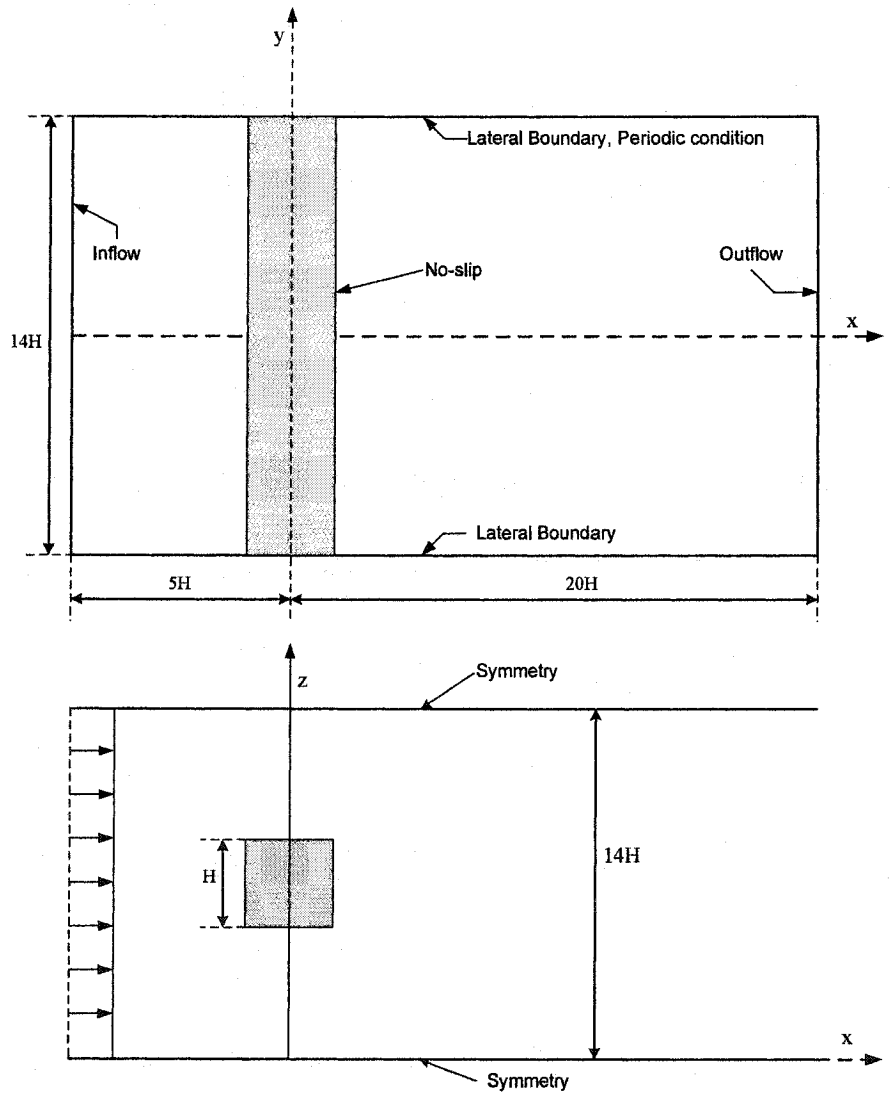


Figure 6.1 Geometry and Boundary Conditions

6.1.1. Velocity Distribution

The profiles of the time-averaged velocities at a location 1.0H downstream of the leeward surface of the cylinder are shown in Figures 6.2 and 6.3. A good agreement between the results was obtained using the hybrid TL/LES model, the LES, and the experimental data. Figure 6.3 shows a lower level of agreement between the hybrid TL/LES and the experimental data, in the wake region.

The profiles of RMS of the velocities are shown in Figures 6.4 and 6.5. Figure 6.4 shows the location of the maximum of RMS of the horizontal velocity, at approximately $z = 0.6H$, in agreement with the Lyn's experimental data. However, the hybrid TL/LES profile of RMS of the horizontal velocity under-predicts the experimental data at the vertical ordinate $z < 0.6H$. The profile of RMS of the vertical velocity is in a good agreement with the Lyn's experimental data in Figure 6.5. Overall, as is seen in Figures 6.2 through 6.5, the hybrid TL/LES model produced acceptable results, when compared with the experimental and LES data.

Figure 6.6 shows the time-averaged horizontal velocities along the centerline, downstream of the cylinder. It is noted that there are discrepancies between the two experimental data sets, which may have resulted from different experimental conditions, e.g. the Reynolds number of Lyn's experiment was 21,400, while the Reynolds number

of Durao's experiment was 14000. The standard k-ε model is shown to over-predict the reattachment length. The hybrid TL/LES model results are in a very close agreement with the LES data reported by Lee (1997). A comparison of the recirculation length (at the centerline downstream of the cylinder) is presented in Table 6.2.

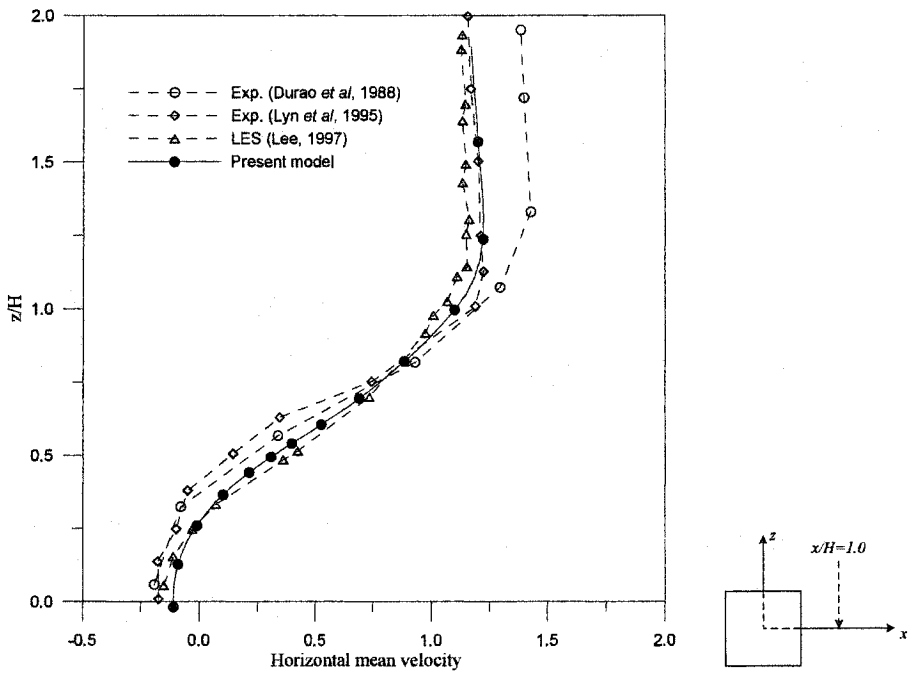


Figure 6.2 Profile of Time-Averaged Horizontal Velocity at x/H = 1.0

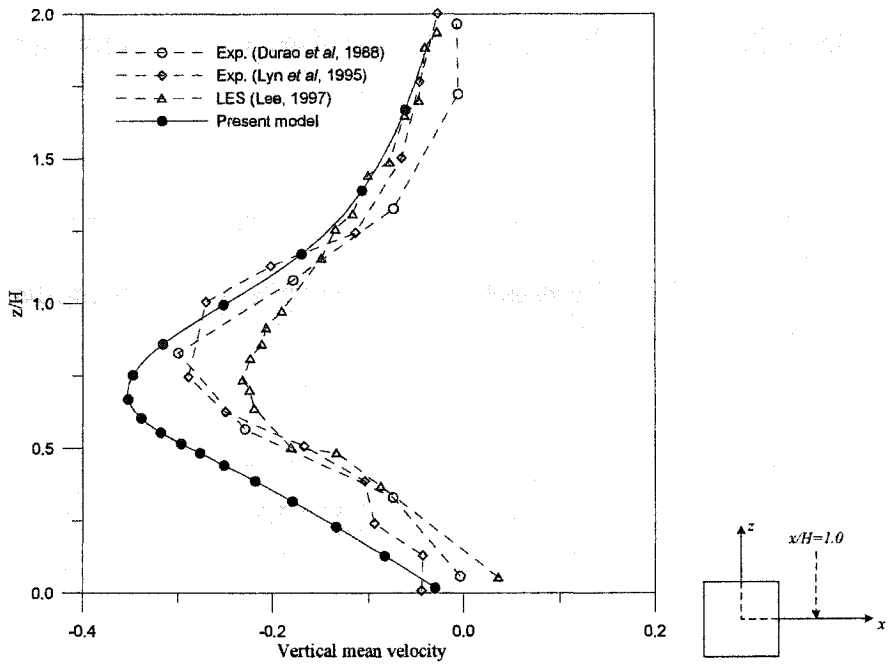


Figure 6.3 Profile of Time-Averaged Vertical Velocity at x/H = 1.0

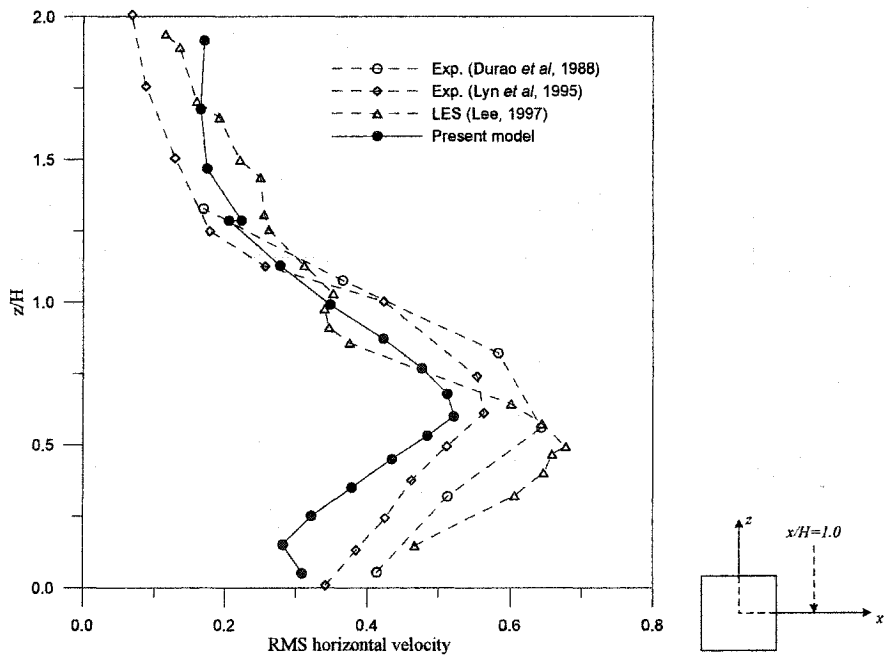


Figure 6.4 Profile of RMS of Horizontal Velocity at x/H = 1.0

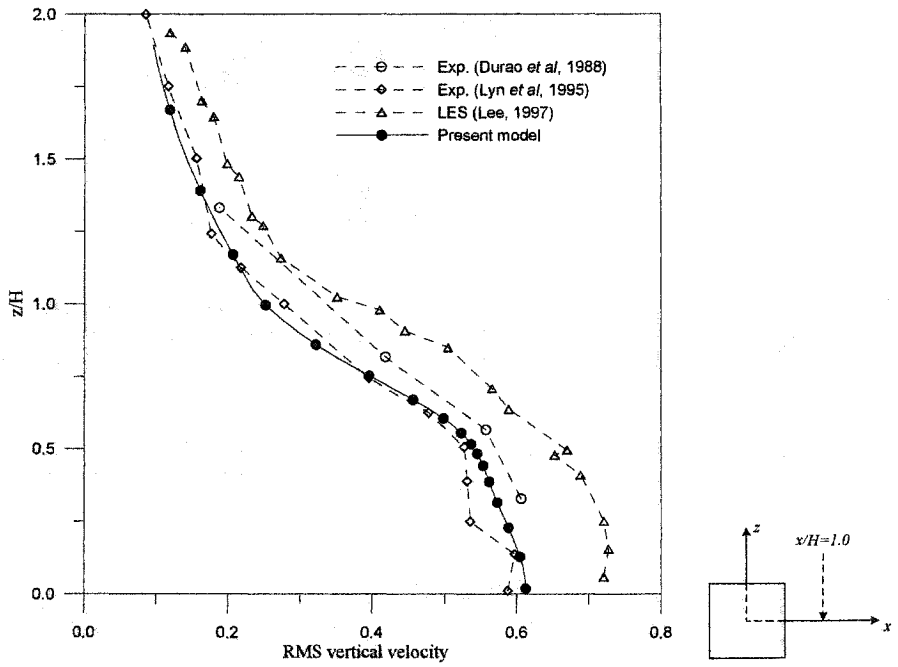


Figure 6.5 Profile of RMS of Vertical Velocity at x/H = 1.0

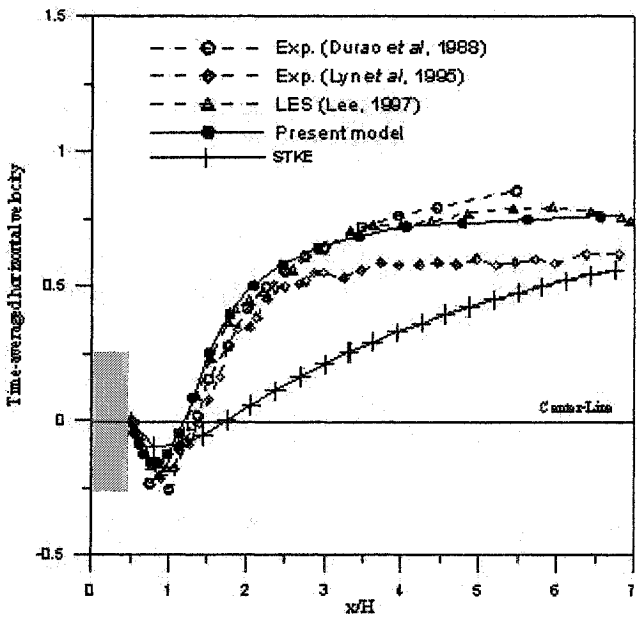


Figure 6.6 Time-Averaged Horizontal Velocity along Center-Line

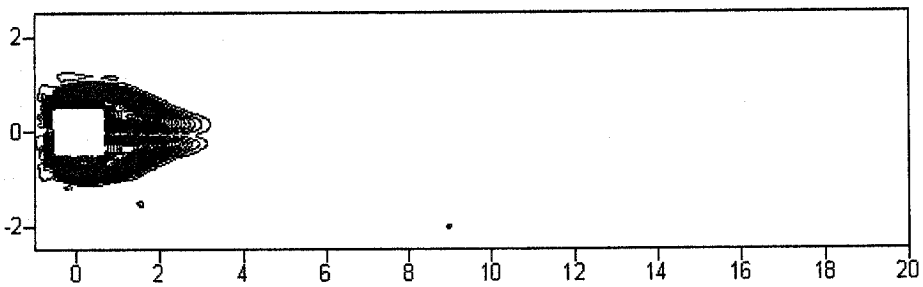
Table 6.2 Comparison of Recirculation Length and Grid Size

Model	Recirculation Length (L/H)	Grid (Horizontal×Vertical×Lateral)
Experiment, Durao *et al.* (1988)	1.37	
Experiment, Lyn *et al.* (1995)	1.38	
Kato and Launder k-ε (1993), Bosh (1995)	2.04	100 × 76 (2D)
LES, Rodi (1997)	1.21	140 × 81 × 13
LES, Lee (1997)	1.36	
Hybrid TL/LES Model	1.21	70 × 50 × 20
k-ε Model	1.95	70 × 50 × 20

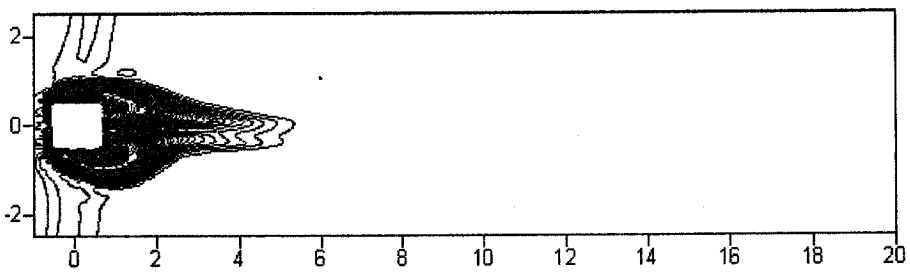
6.1.2. Vorticity Distribution

Figure 6.7 shows the instantaneous span-wise vorticity contours plotted for representative non-dimensional time instants. The computational time step is made dimensionless with the time scale T = H/U, where H is the cylinder height while U is the inflow velocity. It can be seen that the initially symmetric (wake) vortices become non-symmetric as time progresses and that the coherence of the vortices exists only in the near wake (x < 10). The far wake exhibits complex interactions, such as paring and shredding of vortices. Breakdown of the primary vortices into smaller multiple vortices

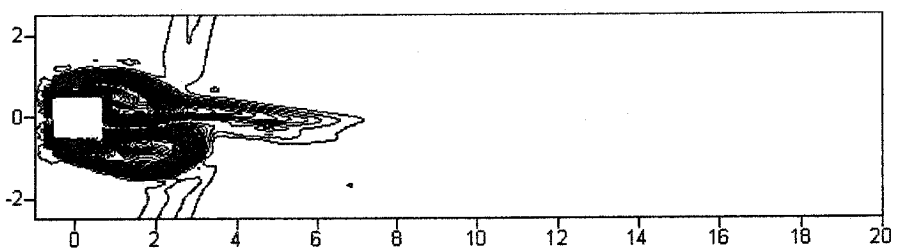
gives rise to a faster decay of the span-wise vorticity.



t = 5T

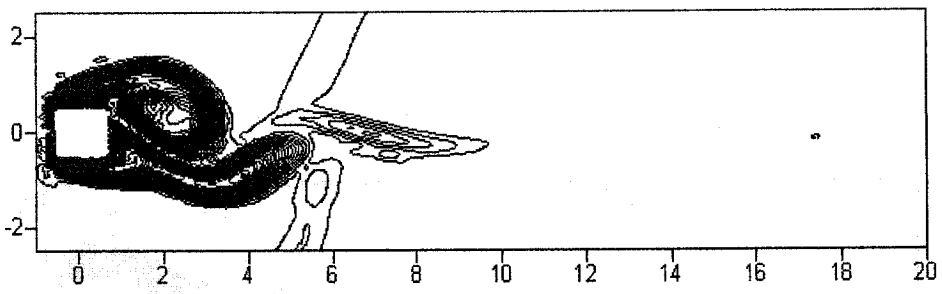


t = 10T

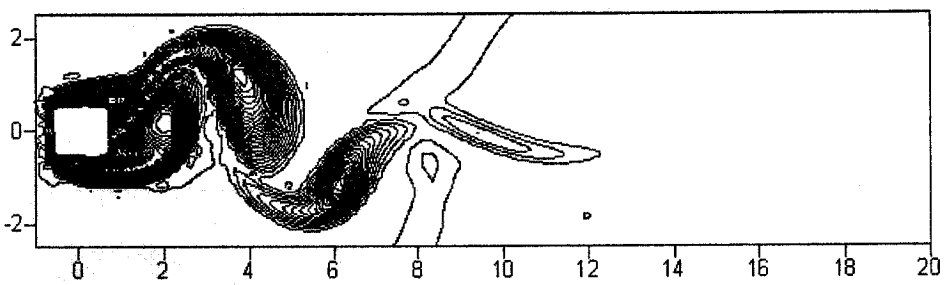


t = 15T

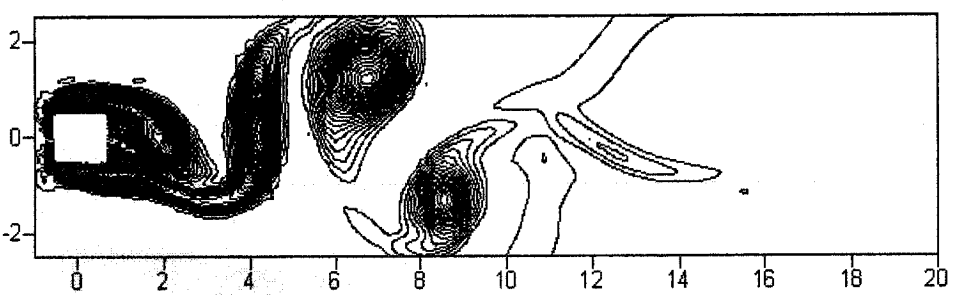
Figure 6.7 Instantaneous Span-Wise Vorticity Contours



t = 20T

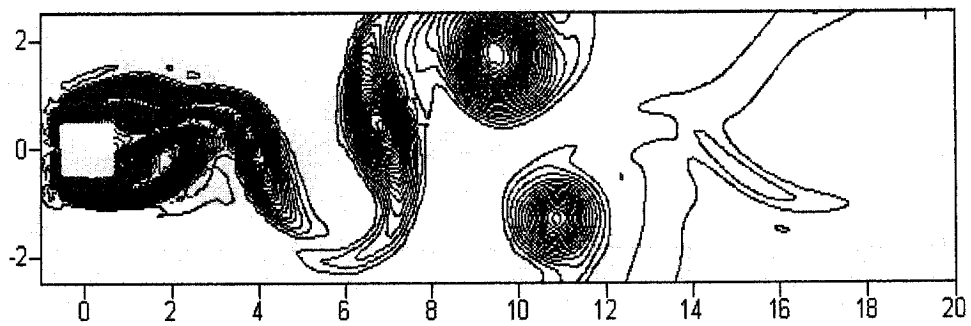


t = 25T

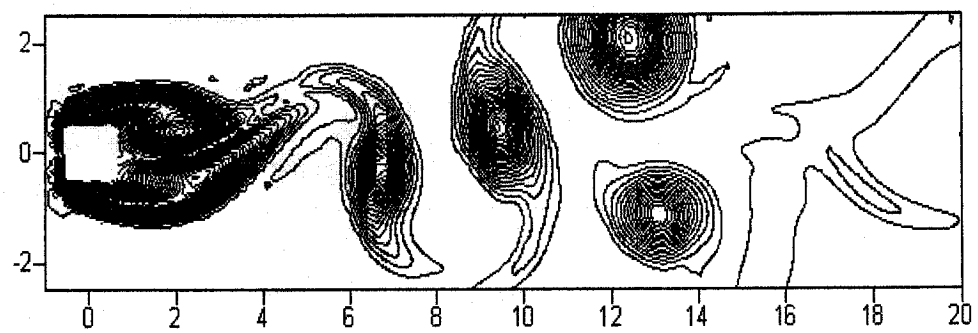


t = 30T

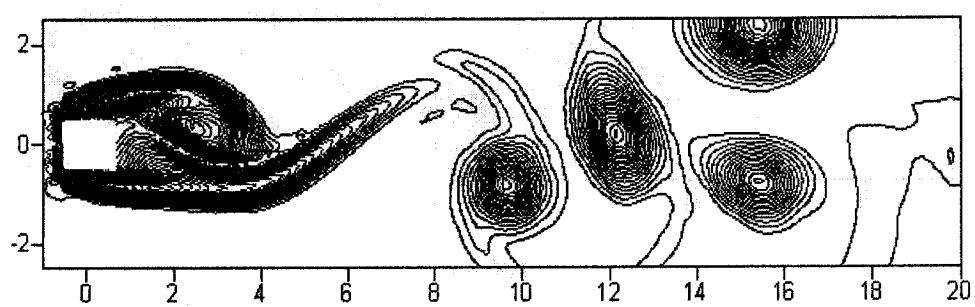
Figure 6.7 (Continued)



t = 35T

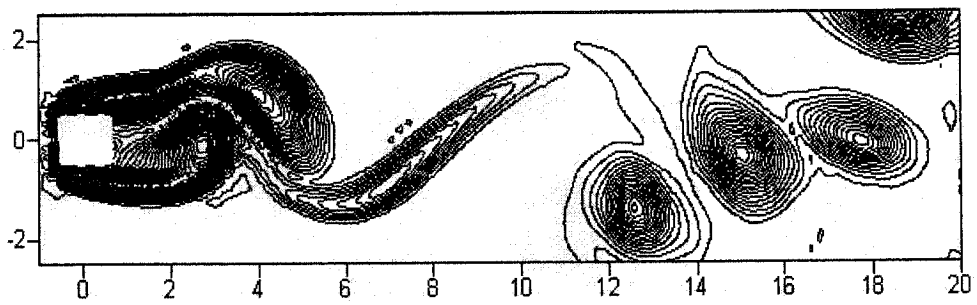


t = 40T

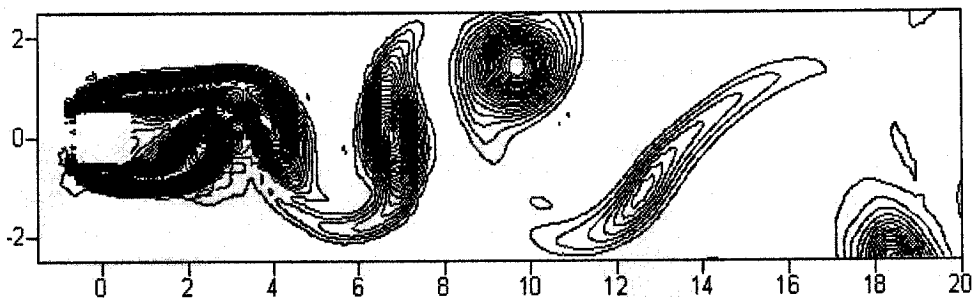


t = 45T

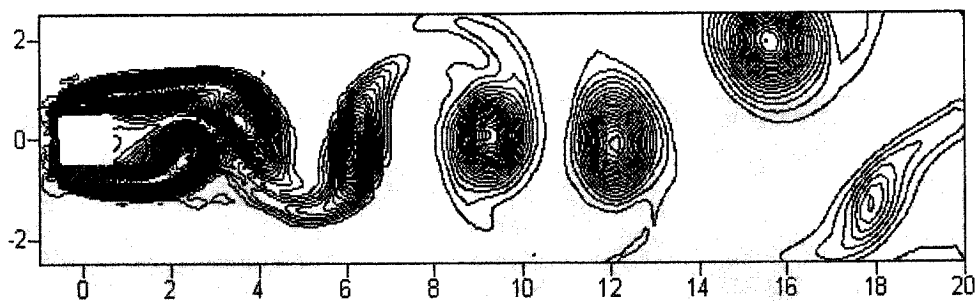
Figure 6.7 (Continued)



t = 50T

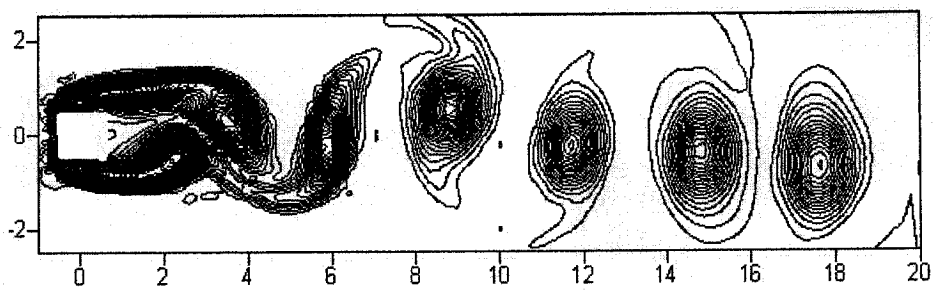


t = 60T

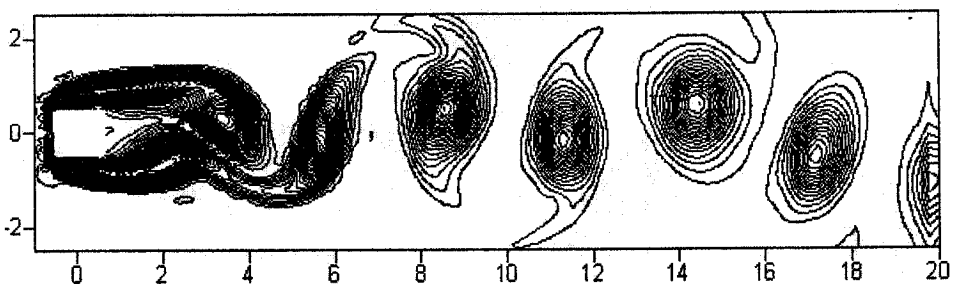


t = 70T

Figure 6.7 (Continued)



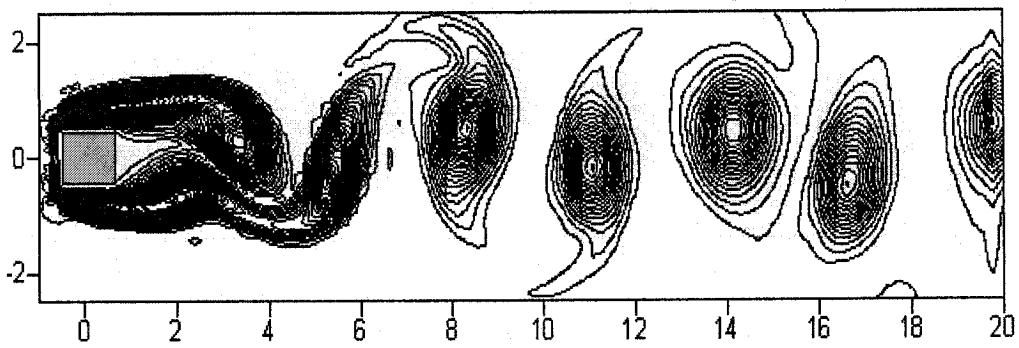
t = 80T



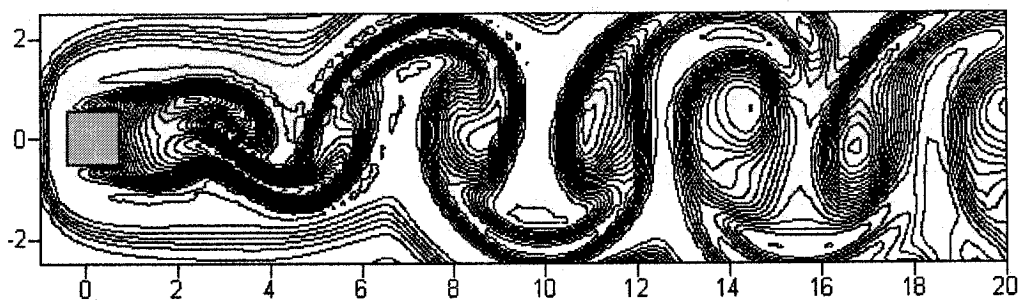
t = 90T

Figure 6.7 (Continued)

The instantaneous span-wise vorticity and the turbulent viscosity contours are compared, for a representative time instant, in Figure 6.8. It can be seen that smaller vortices decay rapidly owing to the viscous action of the fluid.



(a) Span-Wise Vorticity



(b) Eddy Viscosity

Figure 6.8 Contours of the Instantaneous Span-Wise Vorticity and Eddy Viscosity

6.1.3. Turbulent Kinetic Energy

Figure 6.9 shows the time-averaged kinetic energy along the centerline downstream of the cylinder. The results of the hybrid TL/LES model appear to be in a reasonable agreement with the experimental data (Durao *et al.* 1988 and Lyn 1989). However, overall, the time-averaged kinetic energy of wake region predicted by the

hybrid TL/LES model is substantially lower than that inferred from the experimental data. The gradient in the along-flow direction of turbulent kinetic energy of all the compared profiles is relatively small in far wake of the cylinder.

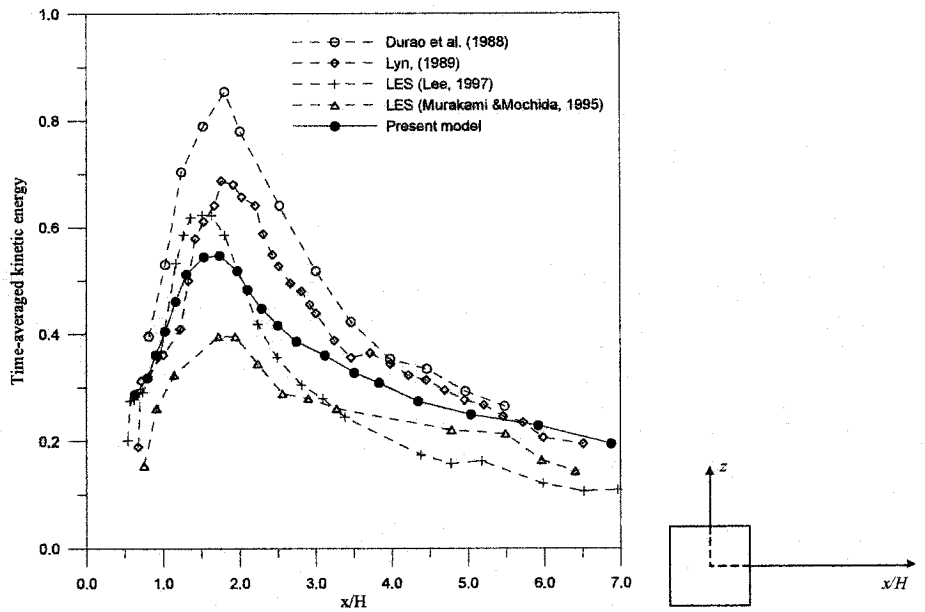
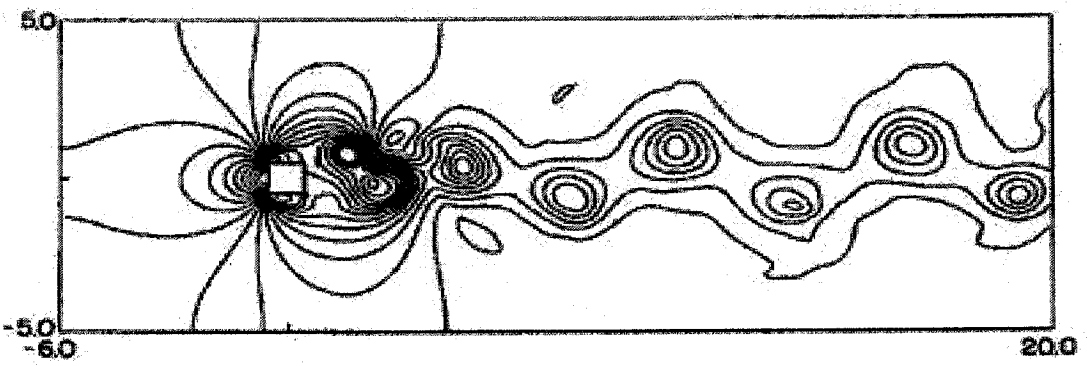


Figure 6.9 Time-Averaged Kinetic Energy at the Center Line

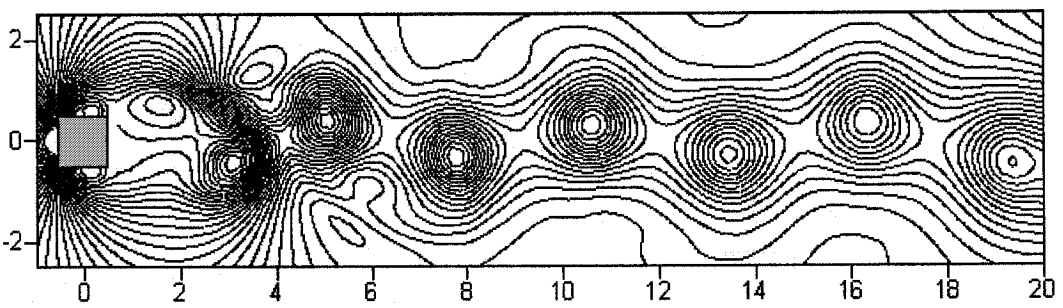
6.1.4 Pressure Distribution

The instantaneous pressure contours obtained using the hybrid TL/LES are compared in Figure 6.10 with the results reported by Saha *et al.* (2003). It can be seen that the compared plots exhibit similar patterns of the pressure. Further examination of Figures 6.8 and 6.10 reveals direct correspondence between the contours of low pressure

and the contours of high levels of vorticity.



(a) Saha *et al.* (2003)



(b) Hybrid TL/LES Model

Figure 6.10 Instantaneous Pressure Contours

The surface pressure coefficient in a plane normal to the cylinder edge is shown in Figure 6.11. The result of the hybrid TL/LES (present model) is compared with the experimental data reported by Bearman and Obasaju (1982) and with the LES results reported by Murakami *et al.* (1996) and by Lee (1997). The pressure coefficient is

defined herein as

$$\bar{C}_P = \frac{\bar{p} - \bar{p}_{ref}}{\frac{1}{2}\rho \bar{U}_{ref}^2} \tag{6.1}$$

where $\bar{p}_{ref}$ and $\bar{U}_{ref}$ are respectively the time-averaged pressure and horizontal velocity at the reference position. The reference position is located at the elevation of the center of cylinder, in the inflow boundary. It can be seen that the pressure coefficient by the hybrid TL/LES is in a very good agreement with the experimental data.

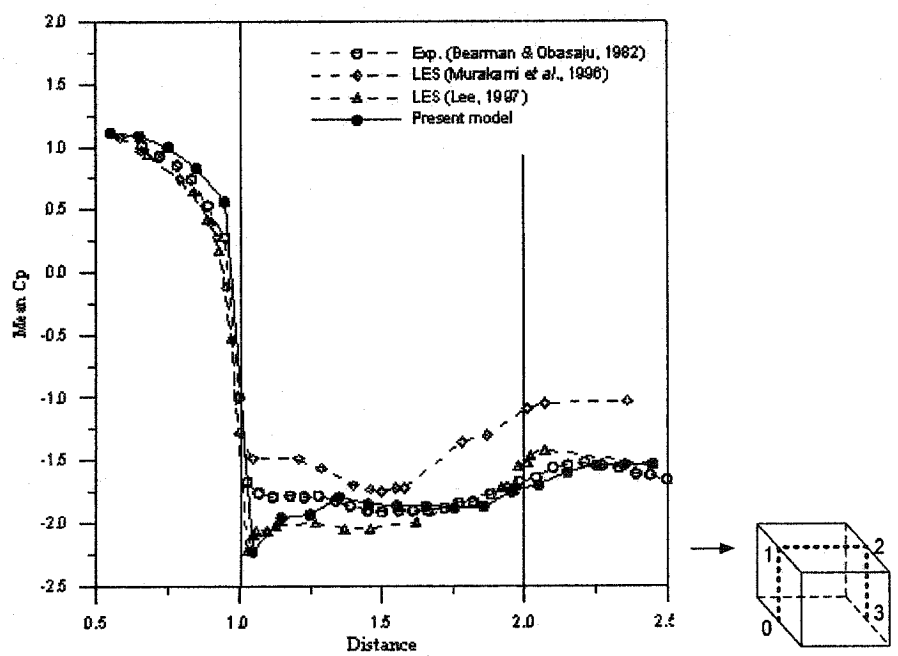


Figure 6.11 Time-Averaged Surface Pressure Coefficient

The distribution of the standard deviation of the surface pressure coefficient obtained by the present model (hybrid TL/LES) is compared in Figure 6.12 with the experimental data of Bearman and Obasaju (1982) and with the result of a numerical simulation reported by Murakami and Mochida (1995). It can be seen that the hybrid TL/LES result shows overall good agreement with the experimental and remaining numerical data on the windward and the leeward faces. However, the hybrid TL/LES model under-predicts the experimental pressure in the middle region of the horizontal (roof) surface.

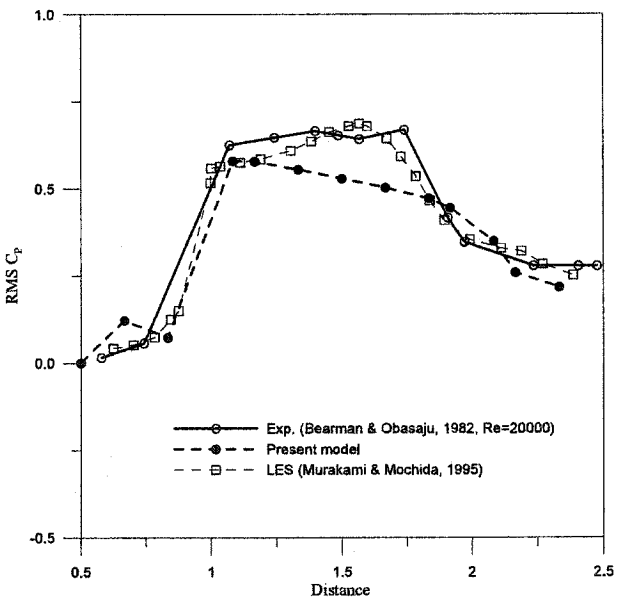


Figure 6.12 RMS of Surface Pressure Coefficient

6.1.5. Summary

The present model was used to simulate (in a three-dimensional domain) the unsteady flow past and the surface pressure on a square cylinder, placed normal to the mean flow. A comparison of the results obtained using the hybrid TL/LES (present model) with the experimental data and numerical results reported by other researchers showed a good performance of the proposed model. Specific observations drawn from this comparison can be summarized as follows:

(1) The profiles of the time-averaged velocities obtained using hybrid TL/LES model were in a good agreement with the experimental and LES data.

(2) The RMS of the horizontal velocity obtained using the hybrid TL/LES model was slightly underestimated (when compared with the experimental data) near the behind the square cylinder. Overall, the RMS prediction using the hybrid TL/LES model was good.

(3) The recirculation length obtained using the hybrid TL/LES model, with a relatively small number of the near-surface grid points, was in a good agreement with the experimental and LES data, while the standard k-ε model failed to provide a correct prediction.

(4) The hybrid TL/LES model captured the instantaneous vortices and complex vertical

interactions in the wake, downstream of the square cylinder.

(5) The kinetic energy of turbulence predicted by the hybrid TL/LES model at the centerline downstream of the square cylinder was smaller than the experimental data.

(6) The predicted instantaneous pressure contours were very similar to the corresponding results reported by other researchers.

(7) The time-averaged pressure coefficient predicted using the hybrid TL/LES was in a very good agreement with the experimental data, while the RMS of pressure coefficient exhibited discrepancies with the experimental data on the horizontal (roof) side of the cylinder.

6.2. FLOW PAST SURFACE-MOUNTED CUBE

Figure 6.13 schematically shows the computational domain and the boundary conditions employed in the three-dimensional numerical simulation of flow past a surface-mounted cube representing a generic building. The building was placed 4.5H (where H is the cube height) downstream of the inlet to the computational domain. The total along-flow length of the domain was 15H and the height was 5H, while its width (in the cross-flow direction) was 9H. The blockage effect is one of the critical factors to be considered in setting up the computational domain. The blockage ratio of the present

study was 2.2 % and it was lower than the upper bound of 3% suggested by Baetke *et al.* (1990). A nominal computational grid was 80 × 33 × 40 in the *x*-, *y*-, and *z*-direction, respectively. It comprised of regions of non-uniformly spaced grid points, with a more refined spatial resolution in the near-wall regions. The location of the closest to the wall grid point was determined using the near-wall treatment criteria discussed earlier. The number of grid points placed in the sub-layer was typically 15-20, similar to that employed by Lakehal and Rodi, (1997). The grid distribution in the viscous sub-layer depicted in Figure 5.1 was employed. For the lower boundary of the computational domain, the no-slip condition and $k = 0$ for the turbulent kinetic energy equation were imposed.

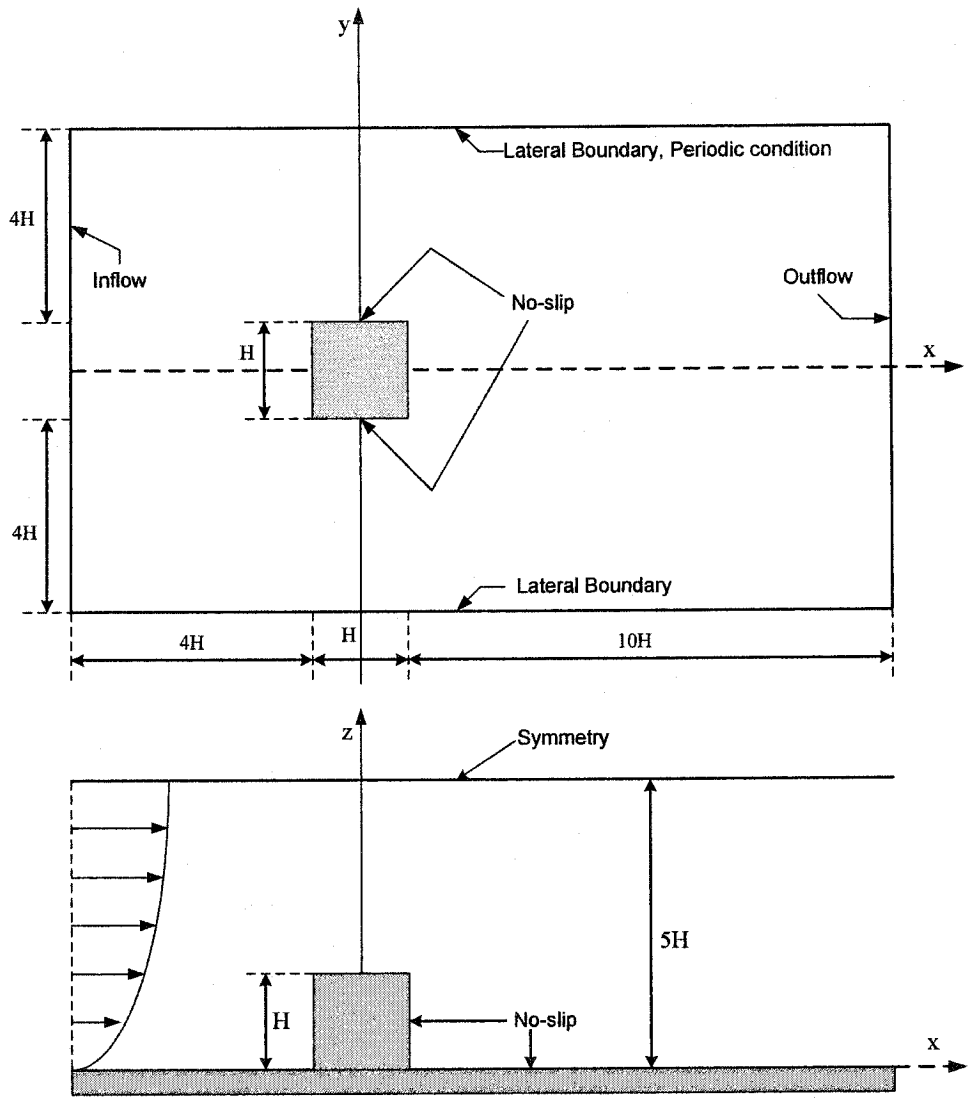


Figure 6.13 Computational Domain and Boundary Conditions

The symmetry condition was enforced on the top of the domain. The periodic condition was employed at the lateral boundaries. A convective boundary condition was applied at the outflow boundary. An approaching velocity profile matched the

experimental profiles reported by Castro and Robins (1977). The fully developed flow based on the experimental conditions was simulated first and it was then used as an inflow condition.

6.2.1. Velocity Distribution

The time-averaged horizontal velocity profiles at two center plane locations downstream of the leeward surface of the cube (x/H = 1.0, and x/H = 2.0) are presented in Figures 6.14 and 6.15. They are compared with the experimental data of Castro and Robins (1977), and with the results of simulations employing the standard k-ε model. In the near wake region behind the cube, the differences between the mean velocity profiles can be attributed to the position and the state of the development of the separated shear layers. As can be seen in Figure 6.14, the predicted mean velocity is in a good agreement with the experimental data, at a close downstream distance from the cube, x/H = 1. However this agreement deteriorates as the downstream distance from the cube is increased, x/H = 2, see Figure 6.15. In the near wake recirculation region (negative velocity region), the hybrid TL/LES results are in a good agreement with the experimental data.

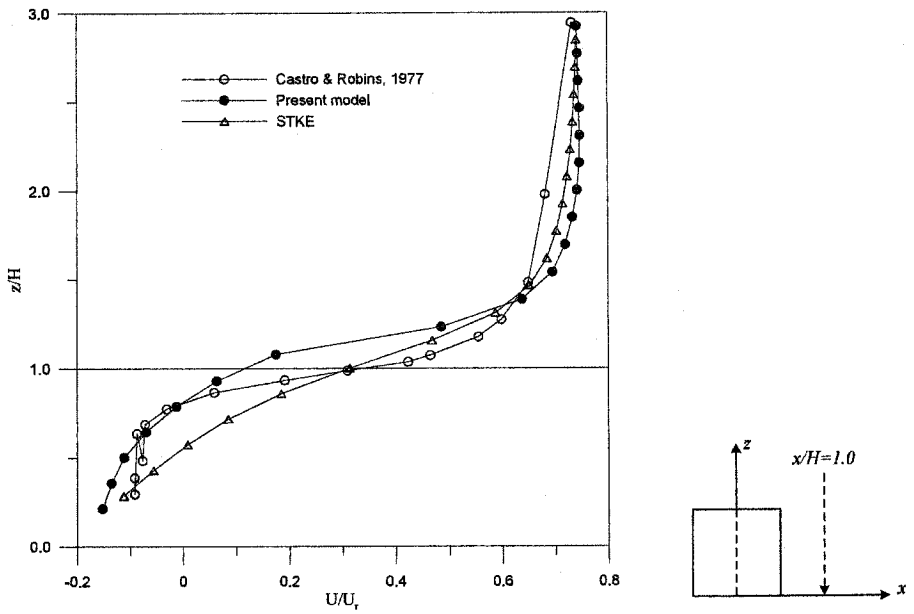


Figure 6.14 Time-Averaged Horizontal Velocity Profiles at $x/H = 1.0$

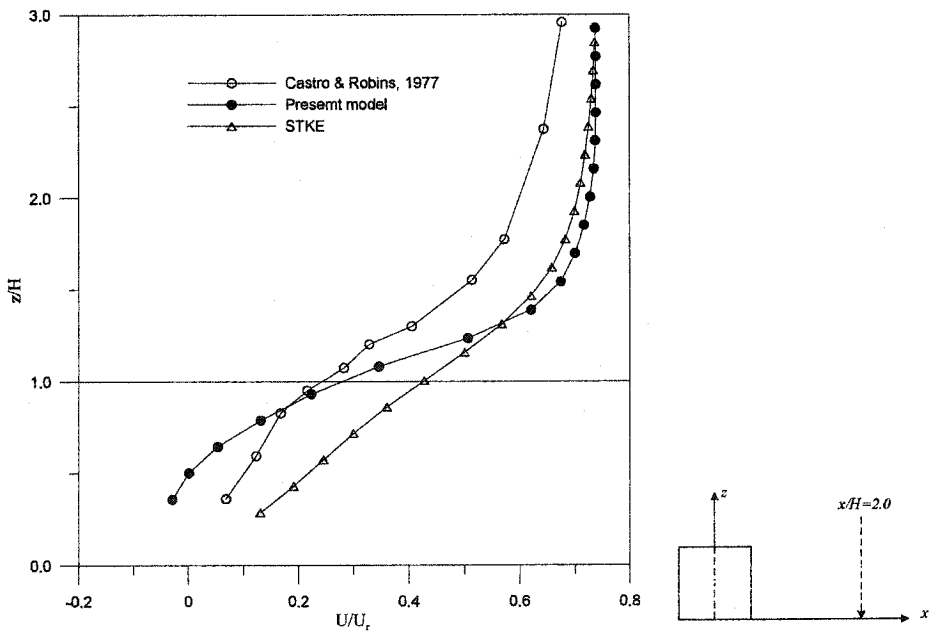


Figure 6.15 Time-Averaged Horizontal Velocity Profiles at $x/H = 2.0$

Figure 6.16 presents a velocity vector plot, in the along-flow plane of symmetry. The location of the primary upstream separation is marked with X_f. This separation is caused by the strong adverse pressure gradient imposed by the presence of the bluff body. The location of the downstream recirculation region of flow is marked with X_R. The numerical values of X_f and X_R obtained in the present study and reported by other researchers are compared in Table 6.3. It can be seen that the values obtained in the present (hybrid TL/LES) study are in a very good agreement with the experimental data of Martinuzzi and Tropea (1993). It also should be noted that the hybrid TL/LES model captured well the separation flow region on the roof of the cube, see Figure 6.16. Overall, the hybrid TL/LES results are in a good agreement with the experimental data even though the employed number of grid points in the vertical direction was 50% smaller than that used in the compared LES cases.

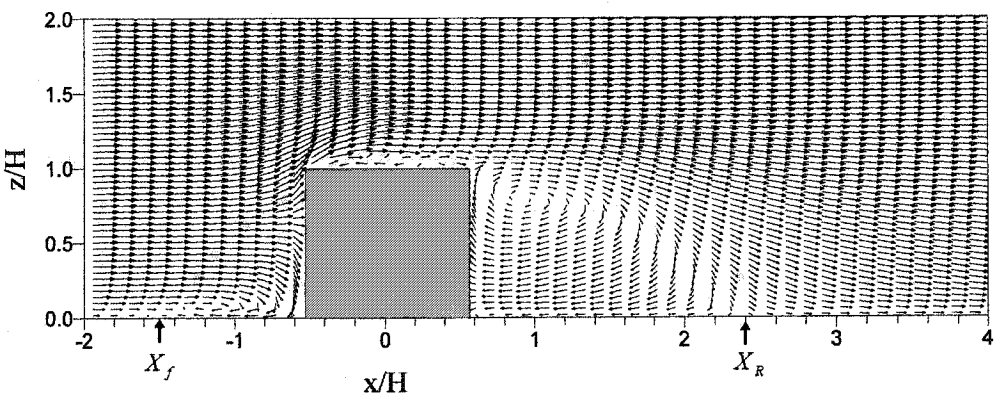


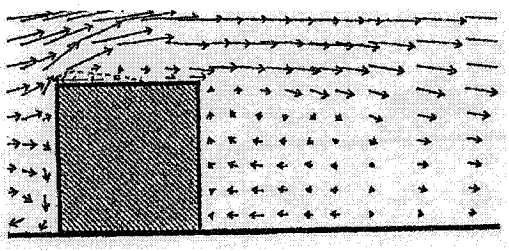
Figure 6.16 Plot of Velocity Vector in Plane of Symmetry

Table 6.3 Comparison of Separation Length and Grid Size

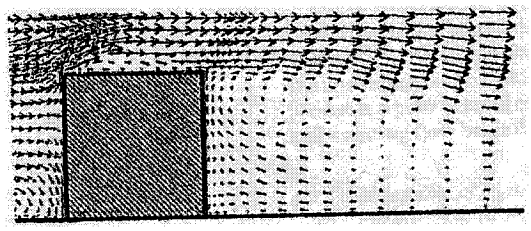
Model	X_f	X_R	Grid (Horizontal ×Lateral×Vertical)
Martinuzzi and Tropea (1993)	1.04	1.61	Experiment
LES-Smagorinsky Breuer *et al.* (1996)	1.287	1.696	165 × 65 × 94
LES-Dynamic Breuer *et al.* (1996)	0.998	1.432	165 × 65 × 94
k-ε Standard Model Lakehal and Rodi (1997)	0.651	2.182	110 × 32 × 32
Two-Layer Model Lakehal and Rodi (1997)	0.95	2.68	142 × 84 × 64
Hybrid TL/LES Model	1.11	1.90	80 × 33 × 40

Figure 6.17 compares the time-averaged velocity vector field obtained in the present study with the results reported by other researchers. They include wind tunnel experimental and numerical simulations (using the LES and the k-ε model) performed by Murakami *et al.* (1990). It can be seen that the hybrid TL/LES results agree fairly well with the experimental and LES data. The reverse flow on the roof and the velocity distribution in the recirculation region behind the model are reproduced. The reverse

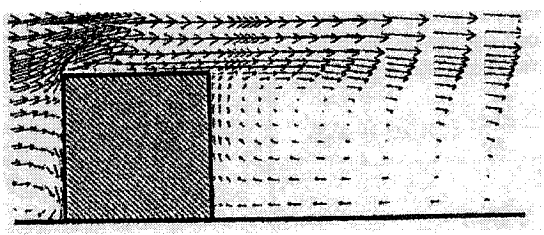
flow on the roof is not reproduced by the k-ε model.



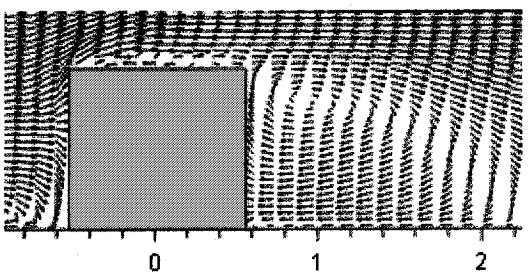
(a) Experiment



(b) LES



(c) k-ε Model



(d) Hybrid TL/LES Model

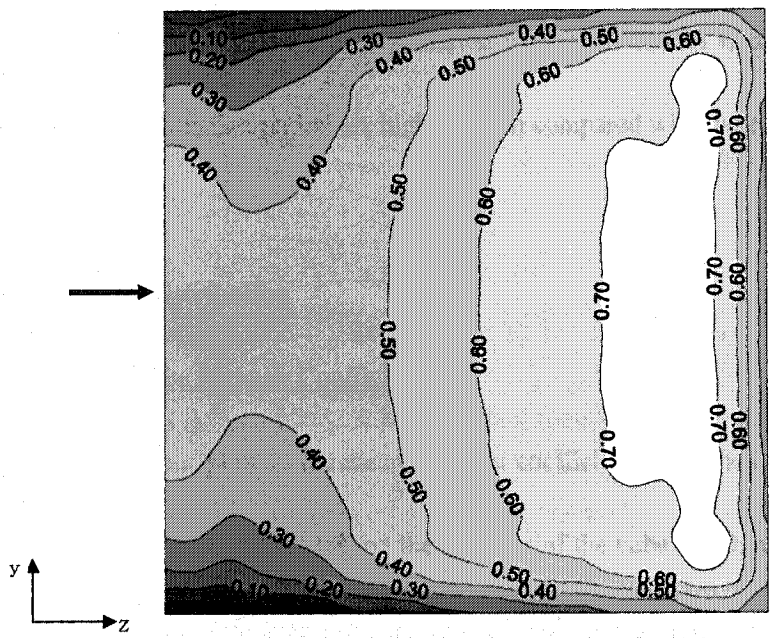
Figure 6.17 Mean Velocity Vector

In addition, the recirculation region predicted by the k-ε model is larger and velocity values in the recirculation region are higher, when compared with the experimental data.

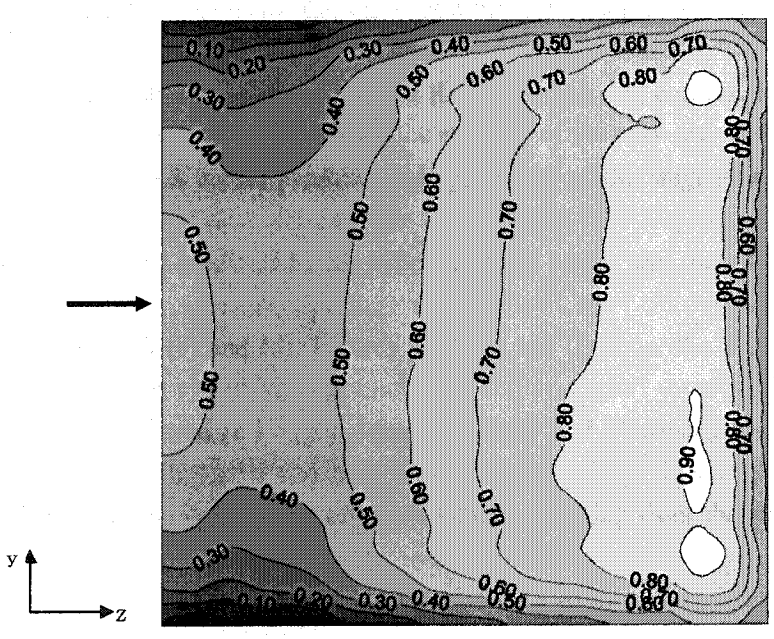
6.2.2. Pressure Distribution

Contours of Surface Pressure Coefficient

The contour plots of the mean pressure coefficients, obtained from the standard k-ε model and the hybrid TL/LES, on the surfaces of the cube are shown in Figure 6.18. The position for the reference pressure and dynamic pressure was at the height of the cube, at the inflow boundary. On the windward wall, see Figures 6.18 (a) and (b), large pressure gradients can be observed near the wall edges and corners. It can be seen that the hybrid TL/LES model predicted a larger value of the stagnation pressure. Higher suctions were also predicted by the hybrid TL/LES on the roof, near the windward edge, see Figures 6.18 (c) and (d). Figures 6.18 (e) and (f) show the pressure distributions predicted by the standard k-ε model and the hybrid TL/LES model, respectively. Overall, the differences between the pressure predictions resulting from the standard k-ε model and the hybrid TL/LES model are attributed to the capacity of the hybrid TL/LES model to predict three-dimensional flow characteristics, especially those associated with flow separation and reattachment.

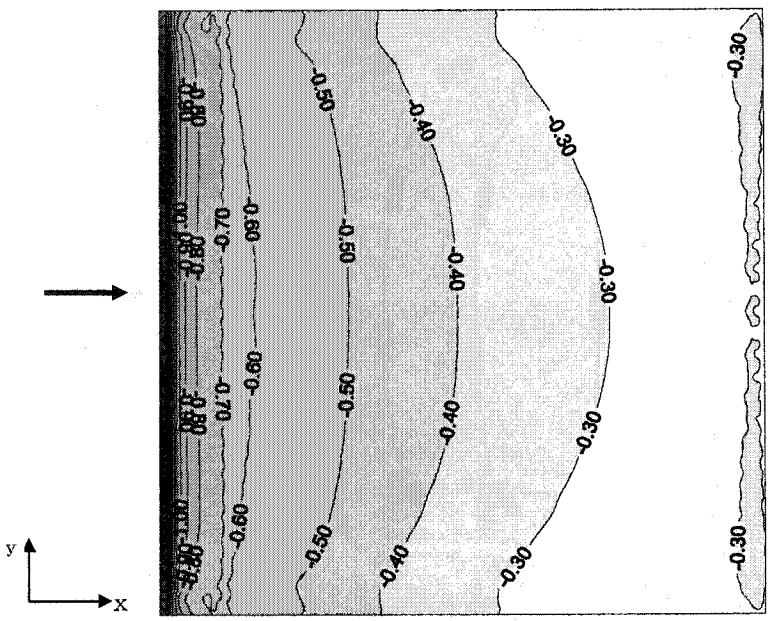


(a) Windward Surface, Standard k-ε Model

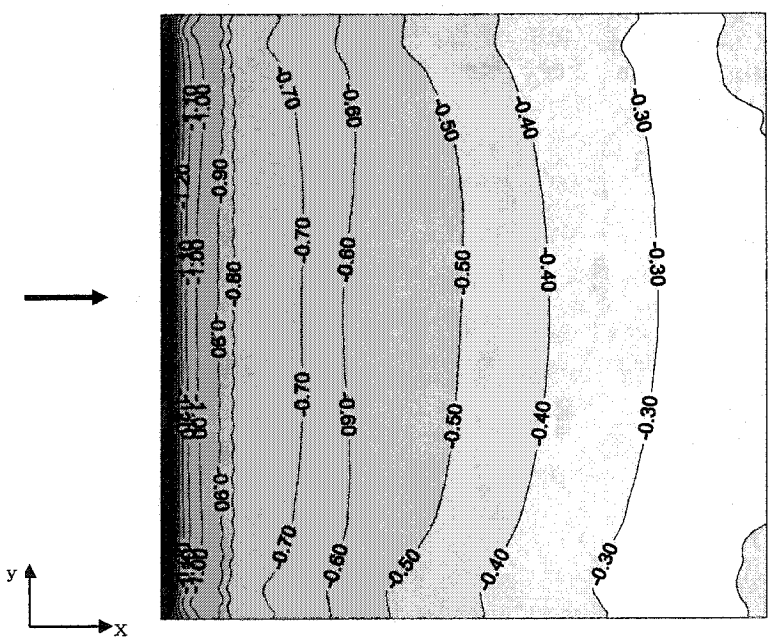


(b) Windward Surface, Hybrid TL/LES Model

Figure 6.18 Contours of Surface Mean Pressure

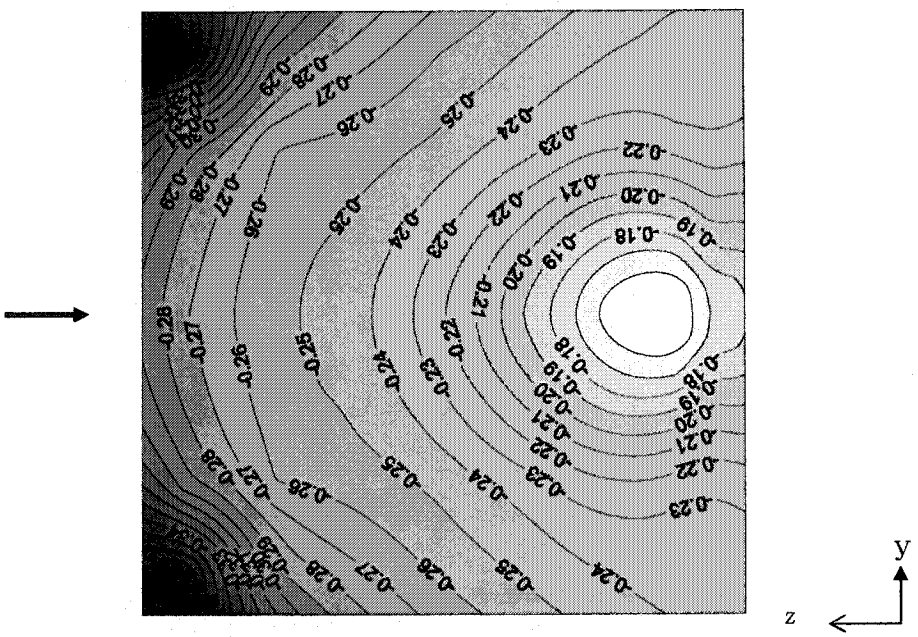


(c) Roof Surface, Standard k-ε Model

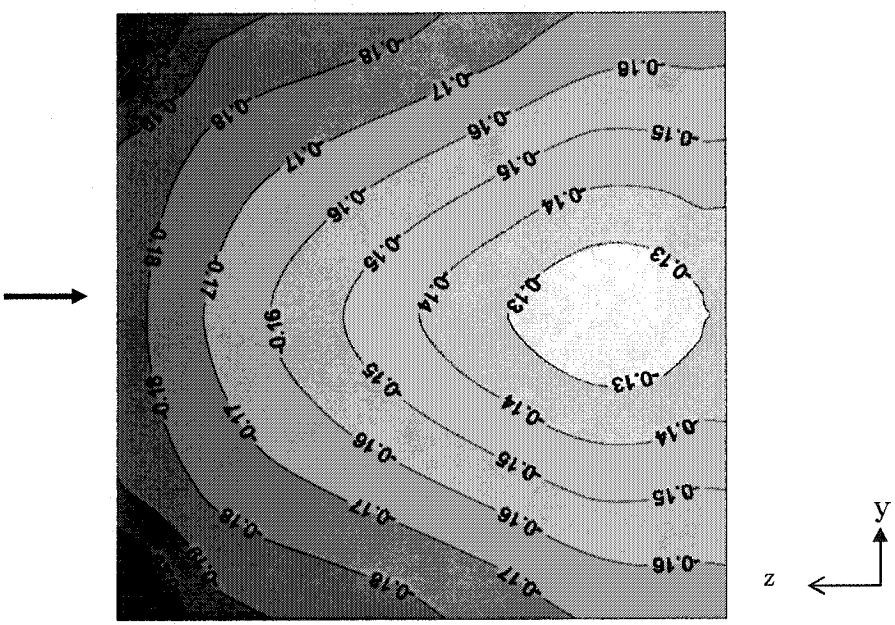


(c) Roof Surface, Hybrid TL/LES Model

Figure 6.18 (Continued) Contours of Surface Mean Pressure



(e) Leeward Surface, Standard k-ε Model



(f) Leeward Surface, Hybrid TL/LES Model

Figure 6.18 (Continued) Contours of Surface Mean Pressure

Distributions of Pressure Coefficient on Surfaces of a Cube

The time-averaged pressure distributions along the center line of the cube surfaces are shown in Figures 6.19 and 6.20. The hybrid TL/LES (present model), the k-ε model and the LES (reported by Murakami, 1992) are compared with the experimental data of Castro and Robins (1977). It can be seen that on the windward surface, the results of the hybrid TL/LES and LES are in a close agreement with the experimental data, whereas the standard k-ε results are under-predicted. The discrepancy in the numerical result of the standard k-ε model also appears in the windward portion of the roof. It is attributed by the inability of the standard k-ε model to account for a loss of isotropy of the turbulent flow. The result of the hybrid TL/LES model is in a good agreement with the experimental data, also on the roof surface. It is evident that the present result is in a closer agreement with the experimental data than the remaining numerical data considered in this comparison. The distribution of the time-averaged surface pressure coefficient in a horizontal plane, see Figure 6.19, also shows that on the windward surface, the present result is in a better agreement with the experimental data than the result of the standard k-ε model. Both the numerical results exhibit discrepancies with the experimental data in the middle portion of the side surface. On the leeward surface, the present result is in a slightly better agreement with the experimental data.

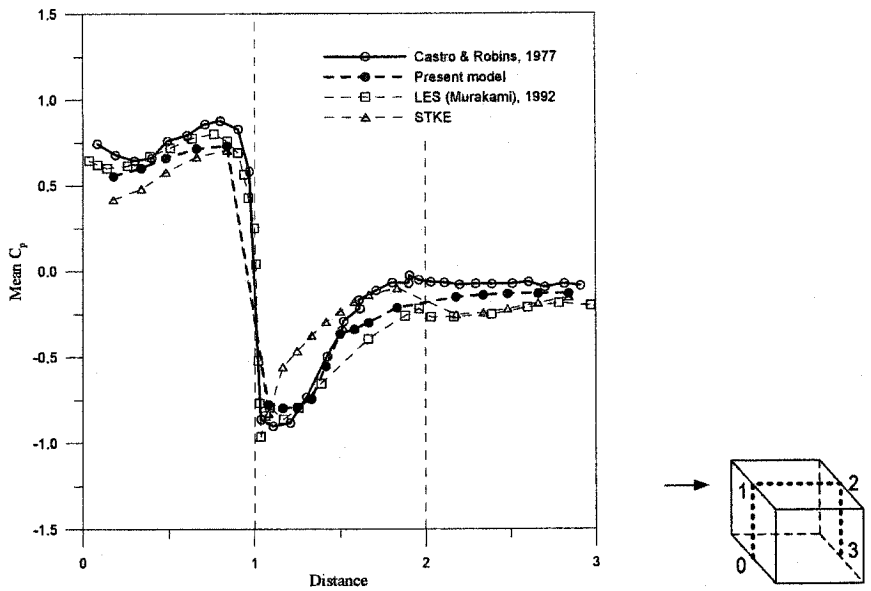


Figure 6.19. Time-Averaged Pressure Coefficient in Center Plane

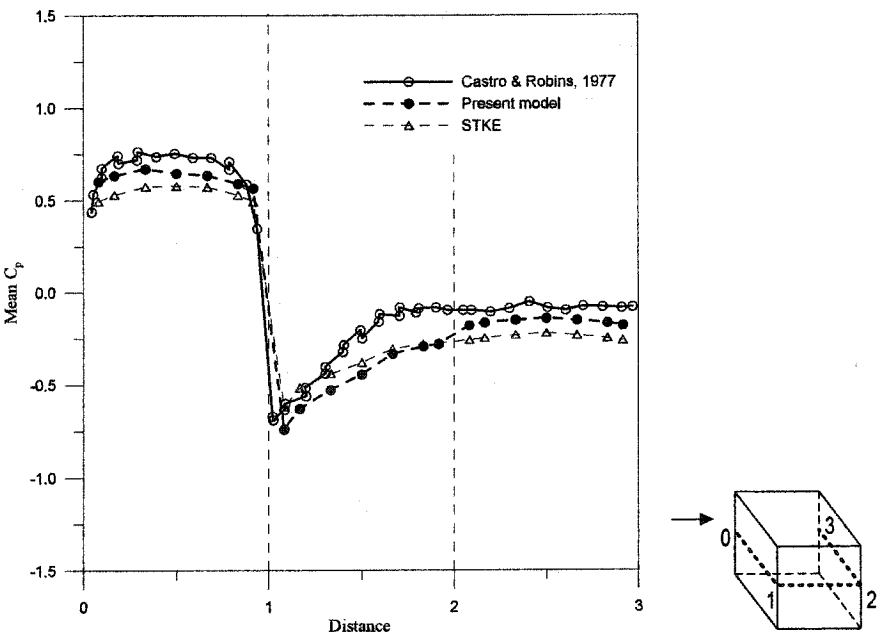
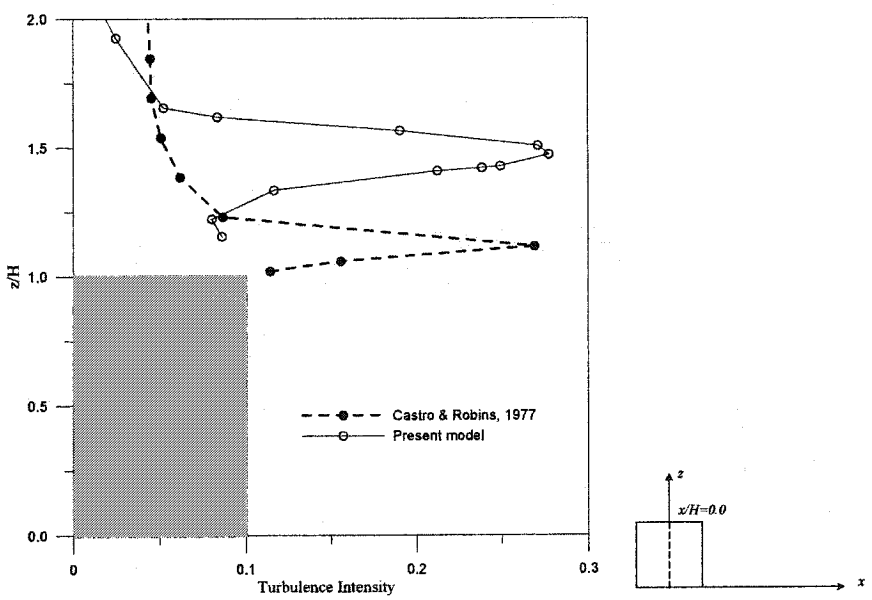


Figure 6.20 Time-Averaged Pressure Coefficient in Horizontal Plane

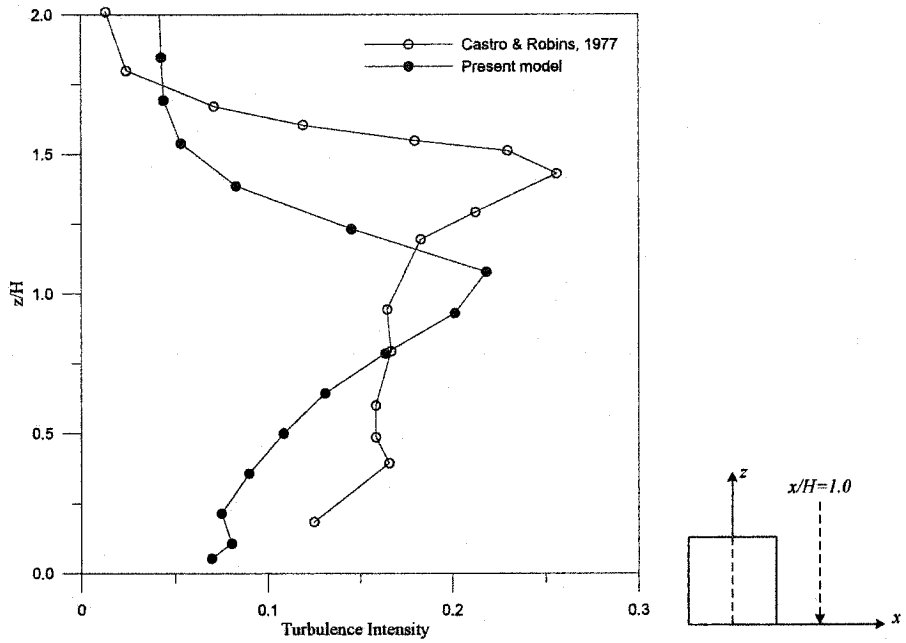
6.2.3 Turbulence Intensity

Figure 6.21 shows the turbulence intensity profiles at three measurement locations. The numerical results obtained using the hybrid TL/LES model are compared with the experimental results. It can be seen in Figure 6.21 that the turbulence intensity profile of the hybrid TL/LES model is in a worse agreement with the experimental data at x/H = 0.0 than at the remaining downstream position. At this location, the peak value from the hybrid TL/LES model is at a significantly lower position than that of the experimental data. At the downstream location (x/H = 1.0, x/H = 2.0), the numerical profile of the turbulence intensity is fairly close to the experimental data.

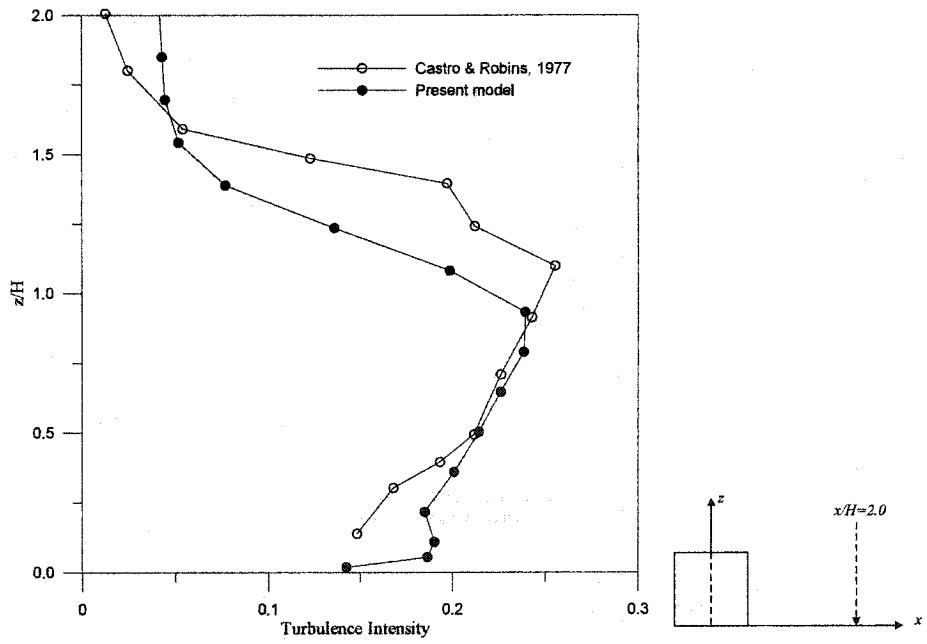


(a) x/H = 0.0

Figure 6.21. Vertical Profile of Turbulence Intensity



(b) x/H = 1.0



(c) x/H = 2.0

Figure 6.21 (Continued) Vertical Profiles of Turbulence Intensity

6.2.4 Summary

The hybrid TL/LES model was used to predict the 3D flow around a surface-mounted cubic model. Findings from the comparison of the hybrid TL/LES model and the experimental results can be summarized as follows:

(1) The time-averaged velocity profiles calculated using the hybrid TL/LES model at downstream positions (x/H = 1.0, x/H = 2.0) were in a better agreement with the experimental data than those of the standard k-ε model simulations.

(2) The velocity vector obtained from the hybrid TL/LES model agreed fairly well with the experimental and LES data.

(3) The hybrid TL/LES model captured the main (upstream) separation and recirculation (downstream) regions, even though a relatively low small grid size was used in computations.

(4) The hybrid TL/LES model provided better and more detailed prediction of the surface pressure distributions than the standard k-ε model.

(5) The time-averaged surface pressure coefficient resulting from the hybrid TL/LES model was in a good agreement with the experimental data.

(6) A significant mismatch between the hybrid TL/LES model and the experimental profiles of turbulence intensity was observed at the center of the roof. At the other

locations (x/H = 1.0, x/H = 2.0) behind the cube, the hybrid TL/LES and experimental profiles were similar.

6.3. FLOW PAST TEXAS TECH UNIVERSITY BUILDING

6.3.1. The Boundary Conditions

The Texas Tech University test building (TTU building) is 9.1 m by 13.7 m in plan and 4.0 m in height. A 1:50 geometrical scale model of this building was tested in a boundary-layer wind tunnel at the Wind Engineering and Fluids Laboratory (WEFL), Colorado State University, Ham (1998). The same model conditions were assumed in numerical simulations described in this section. Figure 6.22 shows that the vertical profile of the mean (horizontal) wind velocity used in the laboratory study reported by Ham (1998).

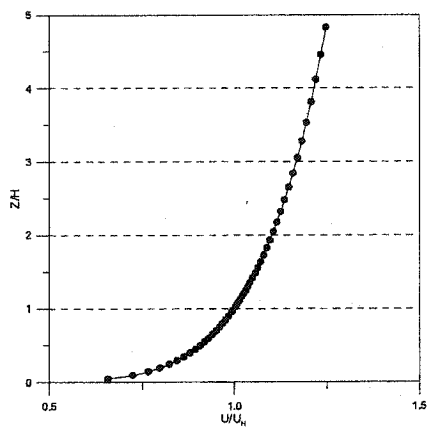


Figure 6.22. Horizontal Mean Wind Velocity Profile

In the numerical simulation, the mean wind velocity of 10.6 m/s, at the roof height of the model (of the TTU building) was assumed. Regarding the boundary conditions employed in computations, the symmetry condition was enforced on the top of the computational domain. At the outflow boundary, the convection condition was imposed for all variables. The no-slip condition was applied at the lower boundary and on the surfaces of the building. The length of the computational domain extended 10H (H denotes the building height) upwind of the windward surface of the building and 20H downwind of the leeward surface of the building. The vertical size of the domain was 5H. The lateral size of the domain was 10H. The building was centrally located between the side boundaries of the domain. Instantaneous velocity profiles, generated using the technique described earlier in this dissertation, were used as inflow velocity conditions. The representative profiles are shown in Figure 6.23. They are compared with the experimental mean velocity profile acquired during an experimental study at WEFL. A power-law fit of the horizontal mean velocity profile was used to calculate the initial friction velocity and the aerodynamic roughness height. Figure 6.24 shows the best fit of the experimental profile was obtained for $z_0 = 0.032$ cm and $u_* = 0.79$ m/s.

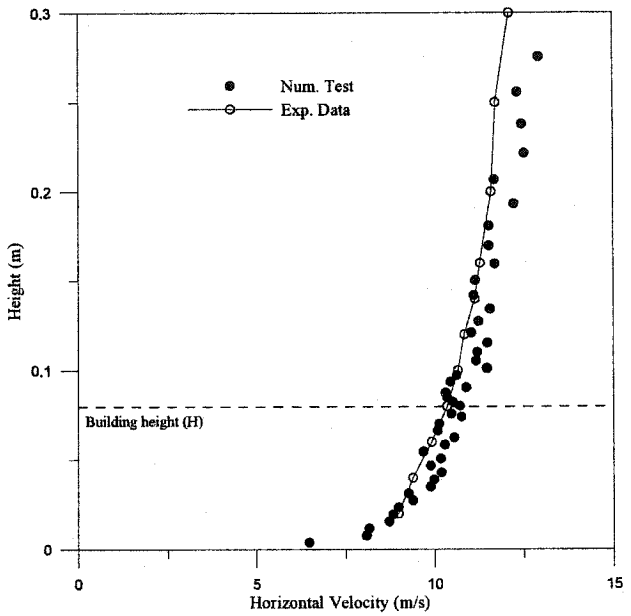


Figure 6.23 Instantaneous Wind Velocity Profiles

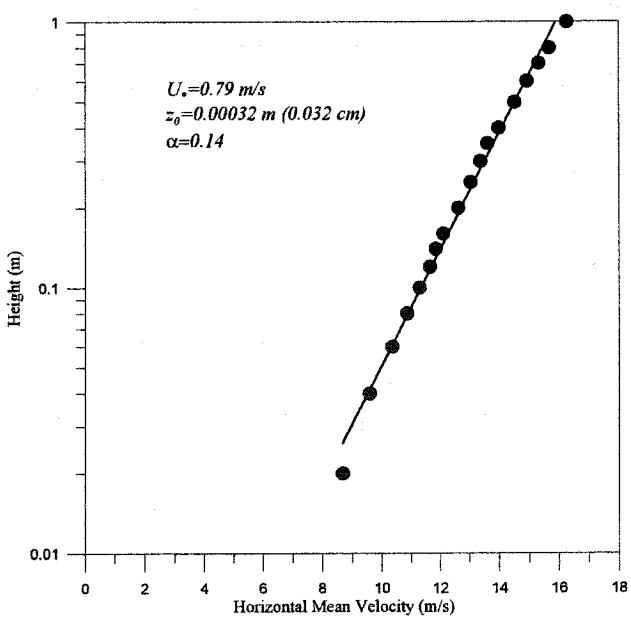


Figure 6.24 Power- and Log-Law Fit of Horizontal Mean Velocity

6.3.2. Pressure Distribution

Unfortunately, the pressure coefficient is the only data available for comparison with the results of the hybrid TL/LES model simulations. The experimental pressure data, the wind-induced external point pressure along the mid-plane of the TTU building, was acquired during wind tunnel testing carried out at the WEFL (Ham, 1998). The mean and standard deviation of wind pressure coefficient induced on the TTU building compiled in Figures 6.25 and 26. Figure 6.25 compares the hybrid TL/LES model (present model) results with the experimental data (Ham, 1998), and numerical simulations obtained using the k-ε model (Paterson & Holmes, 1992 and Mochida *et al.*, 1993) and the LES model (Mochida *et al.*, 1993). It can be seen that the hybrid TL/LES model provides accurate predictions of the mean pressure.

Figure 6.26 compares the distribution of the standard deviation of wind pressure, obtained from various numerical simulations, with the experimental data. It can be seen that the hybrid TL/LES model overestimates the RMS pressure on the roof and on the leeward wall, while it underestimates the pressure on the windward wall. This discrepancy is attributed to insufficient number of grid points near the frontal corner on the roof.

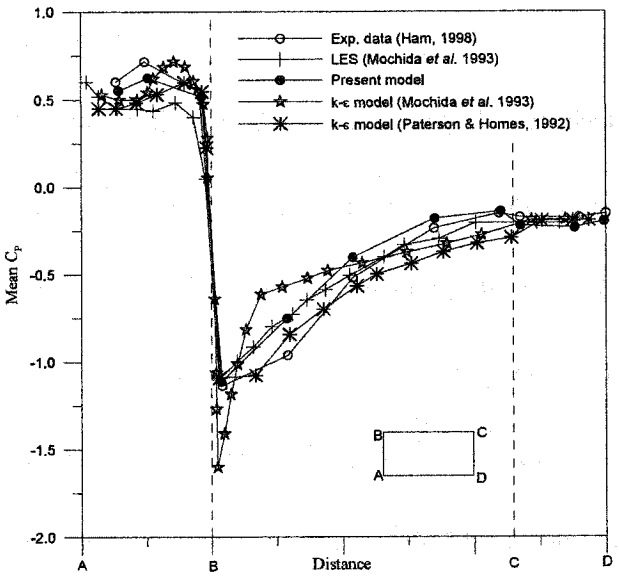


Figure 6.25 Comparison of Mean Pressure Coefficient on TTU Building

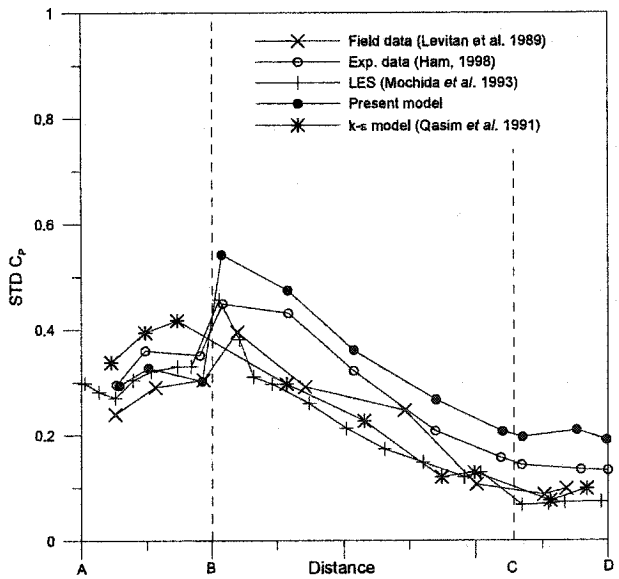


Figure 6.26 Comparison of Standard Deviation of Pressure Coefficient on TTU Building

6.3.3. Summary

The hybrid TL/LES model was employed to predict the 3D flow past TTU building model. Findings for this case can be summarized as follows:

(1) The mean pressure coefficient, obtained using the hybrid TL/LES model was a very good agreement with the experimental and LES data.

(2) The RMS of pressure coefficient was overestimated at the frontal corner on the roof.

(3) The discrepancy in the modeled RMS of pressure coefficient is attributed to insufficient resolution of the computational grid used in numerical simulation.

CHAPTER 7. CONCLUSIONS AND RECOMMENDATIONS

The effort described in this dissertation led to the development and implementation in a computer code of the hybrid TL/LES turbulence model suitable for prediction of flow past and the aerodynamic loading on bluff bodies. The developed computer code has been validated and the representative results are included and discussed in the dissertation.

The hybrid TL/LES model is based on the RANS in the near wall region and on the LES in the outer region. In the near-wall region, a two-layer model is adopted to allow for a reduction in the number of computational grid points (in the direction normal to the wall) without scarifying the accuracy in the computational results. One-equation model was used to resolve the turbulent kinetic energy. To combine the two-layer model with the LES model, the position of the switching line was established automatically during computations. The hybrid TL/LES model was employed in the unsteady three-dimensional calculations of flow past a square cylinder, a surface-mounted cube, and the Texas Tech University test building. The main findings from this research can be

summarized as follows:

(1) The main advantage of the hybrid TL/LES model is a reduction in computational resources required to obtain the results of simulations of flow past bluff bodies with an accuracy of comparable LES predictions.

(2) Comparisons of the hybrid TL/LES and LES simulation results show that the number of computational grid points (in the direction normal to the wall) can be reduced by a factor of two when the standard LES model is replaced by the hybrid TL/LES model, without affecting the accuracy of the computational results.

(3) For given (limited) computational resources, the hybrid TL/LES model provides more reliable predictions of separated flows than other turbulence models.

(4) The results of various numerical tests show a reasonable agreement between the computational predictions involving the hybrid TL/LES model and the experimental data.

(5) The developed computer code was sufficiently accurate to capture the growth of instabilities in the wake behind the bluff body.

(6) The proposed hybrid TL/LES model can be implemented in a computer code suitable for calculations to be carried out on a personal computer.

There are several issues which were not addressed in this dissertation. Topics recommended for future investigations include, but are not limited to:

(1) More detailed studies of the flow in the near-wall region, to determine the number and optimize the distribution of the computational grid points in this region.

(2) Investigation of the effects of various boundary conditions.

(3) A study combining two-layer model with the dynamic SGS model, to examine the numerical simulations in the flow separation and reattachment regions, such as near the windward edges of bluff bodies, e.g. windward roof edges of buildings.

(4) Further investigations are needed to test the performance and limitations of the proposed hybrid TL/LES model. The mathematical consistency of the model should be examined.

(5) Application of higher order methods/unstructured grids should be tested to retain accurate simulations with reduced number of computational grid points.

(6) Implementation of calculations using the multi-processor computational architecture could be included in future work to speed-up calculations and develop a numerical simulation tool suitable for routine application in wind engineering analyses.

(7) The results discussed in this dissertation were obtained from computations carried out using a standard desktop personal computer. Further investigations utilizing enhanced

computational resources (processor and hard disk speed, and RAM memory) are needed to carry out thorough testing of convergence and accuracy of the proposed model.

REFERENCES

ASCE 7-98 (1998), *Minimum design loads for buildings and other structures*, American Society of Civil Engineers, Reston, VA, USA.

Baetke, F., Werner, H., and Hans, W. (1990), "Numerical simulation of turbulent flow over surface-mounted obstacles with sharp edges and corners", *J. Wind Eng. & Ind. Aerodyn.*, vol. 35, pp. 129-147.

Battern, P., Goldberg, U., and Chakravarthy S. (2002), "LNS - An approach towards embedded LES", *AIAA Paper*, no. 2002-0427, pp. 1-10.

Bearman, P.W. and Obasaju, E.D. (1982), "An experimental study of pressure fluctuations on fixed and oscillating square-section cylinders", *J. Fluid Mech.*, vol. 119, pp. 297-321.

Bosch, G. (1995), *Experimentelle und theoretische undersuchung der instationaren stromung um/ylindrische strukturen*, Dissertation, University of Karlsruhe, Germany.

Bosch, G. and Rodi, W. (1998), "Simulation of vortex shedding past a square cylinder with different turbulence models", *Int. J. for Num. Meth. Fluids*, vol. 38, pp. 601-616.

Boussinesq, J. (1877), "Essai sur la théorie des eaux courantes", *Mémoires á l'Académie des Sciences*, vol. 23, no. 1, pp. 601-680.

Breuer, M., Jovicic, N., and Mazaev, K. (2003), "Comparison of DES, RANS and LES for the separated flow around a flat plate at high incidence", *Int. J. for Num. Meth. Fluids*, vol. 41, pp. 357-388.

Breuer, M., Lakehal, D., and Rodi, W. (1996), "Flow around a surface mounted cubical obstacle: Comparison of LES and RANS-results", *Num. Fluid Mech.*, vol. 53, pp. 22-30.

Cabot, W.H. and Moin, P. (1993), "Large eddy simulations of scalar transport with the dynamics subgrid-scale model", *Large eddy simulation of complex engineering and geophysical flows*, B. Galperin, and S.A. Orszag editors, Cambridge University Press., pp. 3-36.

Castro, I.P. and Robins, A.G. (1977), "The flow around a surface-mounted cube in uniform and turbulent streams", *J. Fluid Mech.*, vol. 79, no. 2, pp. 307-335.

Caughey, D.A. and Jothiprasad, G. (2002), "Tools for large-eddy simulation", *Center for Turbulence Research Proceedings of the Summer Program*, pp. 117-127.

Chen, H.C. and Patel, V.C. (1988), "Near-wall turbulence models for complex flows including separation", *AIAA Journal*, vol. 26, no. 6, pp. 641-648.

CWE '92 (1992), S. Murakami editor, *Computational Wind Engineering 1, Proc. of the 1st Int. Symp. Computational Wind Engineering*, Tokyo, Japan.

CWE '96 (1996), R.N. Meroney and B. Bienkiewicz editors, *Computational Wind Engineering 2, Proc. of the 2nd Int. Symp. Computational Wind Engineering*, Fort Collins,

CO, USA.

Davidson, L. (2001), "Hybrid LES-RANS: A combination of a one equation SGS model and a k-ω model for prediction of recirculation flows", *ECCOMAS Computational Fluid Dynamics 2001 Conference*, Swansea, UK.

Deardorff, J.W. (1974), "Three-dimensional numerical study of the height and mean structure of a heated planetary boundary layer", *Boundary-Layer Meteorol.*, vol. 7, pp. 81-106.

Dejoan, A. and Schiestel, R. (2002), "LES of unsteady turbulence via a one-equation subgrid-scale transport model", *Int. J. of Heat and Fluid Flow*, vol. 23, pp. 398-412.

Durao, D.F.G., Heitor, M.V., and Pereira, J.C.F. (1988), "Measurements of turbulent and periodic flows around a square cross-section cylinder", *Experiments in Fluids*, vol. 6, pp. 298-304.

Georagiadis, N.J, Alexander, J.I.D., and Reshotko, E. (2003), "Hybrid Reynolds-averaged Navier-Stokes/large-eddy simulations of supersonic turbulent mixing", *AIAA Journal*, vol. 41, no. 2, pp. 218-229.

Germano, M., Piomelli, U., Moin, P., and Carbot, W.H. (1991), "A dynamic subgrid-scale eddy viscosity model", *Physics of Fluids*, vol. 3, pp. 1760-1765.

Ham, H.J. (1998), *Turbulence effects on wind-induced building pressure*, Ph.D. Dissertation, Colorado State University, Fort Collins, CO, USA.

Ham, H.J. and Bienkiewicz, B. (1998), "Wind tunnel simulation of TTU flow and building roof pressure", *J. Wind Eng. & Ind. Aerodyn.*, vol. 77&78, pp. 119-133.

Hamed, A., Basu, D., and Das, K. (2003), "Detached eddy simulations of supersonic flow over cavity", *AIAA Paper*, no. 2003-0549, *41st AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada.

Hassid, S. and Poreh, M. (1975), "A turbulent energy model for flows with drag reduction", *J. Fluids Eng.*, vol. 97, no. 2, pp. 234-241.

Hoshiya, M. (1972), "Simulation of multi-correlated random processes and application to structural vibration problems", *Proc. of JSCE*, vol. 204, pp. 121-128.

Hwang, R.R. and Jaw, S.Y. (1988), "Second-order closure turbulence models: Their achievements and limitations", *Proc. of Natl. Sci. Counc. ROC(A)*, vol. 22, no. 6, pp. 703-722.

Iacovides, H. and Launder, B.E. (1987), "The numerical simulation of flow and heat transfer in tubes in orthogonal-mode rotation", *Proc. of 6th Symp. on Turbulent Shear Flows*.

Jacobsen, C.B. (1997), *Large eddy simulation of confined swirling flow: Numerical part*. Ph.D. Thesis, Institute of Energy Technology, Aalborg University, Denmark.

Jaw, S.Y. and Chen, C.J. (1998), "Present status of second-order closure turbulence models. I: Overview", *J. of Eng. Mech.*, vol. 124, no. 5, pp. 485-501.

Johansen, S.T., Wu, J., and Shyy, W. (2004), "Filter-based unsteady RANS computations",

Int. J. of Heat and Fluid Flow, vol. 25, pp. 10-21.

Kato, M. and Launder, B.E. (1993), "The modelling of turbulent flow around stationary and vibrating square cylinders", *Proc. of 9th Symp. on Turbulent Shear Flow*, Kyoto, Japan, pp. 10-14.

Kondo, K., Murakami, S., and Mochida, A. (1997), "Generation of velocity fluctuations for inflow boundary condition of LES", *J. Wind Eng. & Ind. Aerodyn.*, vol. 67&68, pp. 51-64.

Krajnović, S. and Davidson, L. (2002), "A mixed one-equation subgrid model for large-eddy simulation", *Int. J. of Heat and Fluid Flow*, vol. 23, pp. 413-425.

Lakehal, D. and Rodi, W. (1997), "Calculation of the flow past a surface-mounted cube with two-layer turbulence models", *J. Wind Eng. & Ind. Aerodyn.*, vol. 67&68, pp. 65-78.

Langhe, C.D., Merci, B. and Dick, E. (2002), "Hybrid RANS-LES modelling with the renormalization group", *Proc. of ASME FEDSM'02.*, FEDSM2002-31280, pp. 647-651.

Launder, B.E. and Sharma, B.I. (1974), "Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc", *Lett. Heat Mass Transf.*, vol. 1, pp. 131-138.

Launder, B.E. and Spalding, D.B. (1990), "The numerical computation of turbulent flows", *Comp. Meth. in App. Mech. Eng.*, vol. 3, pp.269-289.

Lee, S. (1997), *Large eddy simulation of flow past a square cylinder using finite element method*, Ph.D. Dissertation, Colorado State University, Fort Collins, CO, USA.

Lee, S. and Bienkiewicz, B. (1997), "Finite element implementation of large eddy simulation for separated flows", *Proc. of 8th US Nat. Conf. Wind Eng.*, Baltimore, MD, USA.

Leonard, A. (1974), "Energy cascade in large eddy simulations of turbulent fluid flow", *Adv. Geophys.*, vol. 18A, pp. 237-248.

Lilly, D.K. (1967), "The representation of small-scale turbulence in numerical simulation experiments", H.H. Goldstine editor, *Proc. of IBM Scientific Computing Symp. on Environmental Sciences*, Yorktown Heights, N.Y., pp. 195-210.

Lilly, D.K. (1992), "A proposed modification of the Germano subgrid-scale closure method", *Phys. Fluid A.*, vol. 4, pp. 633-635.

Lyn, D.A. (1989), "Phase-averaged turbulence measurements in the separated shear layer region of flow around a square cylinder", *Proc. of 23rd Cong. Int. Ass. Hydraulic Res.*, Ottawa, Canada, vol. A, pp. 85-92.

Lyn, D.A., Einav, S., Rodi, W., and Park, J.-H. (1995), "A laser-Doppler velocimetry study of ensemble-averaged characteristics of the turbulent near wake of a square cylinder", *J. Fluid Mech.*, vol. 304, pp. 285-319.

Lyn, D.A. and Rodi, W. (1994), "The flapping shear layer formed by flow separation from the forward corner of a square cylinder", *J. Fluid Mech.*, vol. 267, pp. 353-379.

Mani, M. (2004), "Hybrid turbulence models for unsteady flow simulation", *J. of Aircraft*, vol. 41, no. 1, pp. 110-118.

Martinuzzi, R. and Tropea, C. (1993), "The flow around surface-mounted, prismatic obstacles placed in a fully developed channel flow", *J. Fluids Eng.*, vol. 115, pp. 85-92.

Mathews, E.H. (1987), "Prediction of the wind-generated pressure distribution around buildings", *J. Wind Eng. & Ind. Aerodyn.*, vol. 25, pp. 219-228.

Menter, F.R. (1994), "Two-equation eddy-viscosity turbulence models for engineering applications", *AIAA Journal*, vol. 32, no. 8, pp. 1598-1605.

Mochida, A., Murakami, S., Shoji, M., and Ishida, Y. (1993), "Numerical simulation of flowfield around Texas Tech building by large eddy simulation", *J. Wind Eng. & Ind. Aerodyn.*, vol. 46&47, pp. 455-460.

Moin, P. and Kim, J. (1982), "Numerical investigation of turbulent channel flow", *J. Fluid Mech.*, vol. 118, pp. 341-377.

Morton, S.A., Forsythe, J.R., Mitchell, A.M., and Hajek, D. (2002), "DES and RANS simulations of delta wing vertical flows", *AIAA Paper*, no. 2002-0587.

Murakami, S. (1993), "Comparison of various turbulence models applied to a bluff body", *J. Wind Eng. & Ind. Aerodyn.*, vol. 46&47, pp. 21-36.

Murakami, S., Mochida, A., and Hibi, K. (1987), "Three-dimensional numerical simulation of airflow around a cubic model by means of large eddy simulation", *J. Wind Eng. & Ind. Aerodyn.*, vol. 25, pp. 291-305.

Murakami, S. and Mochida, A. (1988), "3-D numerical simulation of airflow around a cubic model by means of the k-ε model", *J. Wind Eng. & Ind. Aerodyn.*, vol. 31, pp. 283-303.

Murakami, S., Mochida, A., and Hayashi, Y. (1990), "Examining the k-ε model by means of a wind tunnel test and large-eddy simulation of the turbulence structure around a cube", *J. Wind Eng. & Ind. Aerodyn.*, vol. 35, pp. 87-100.

Murakami, S., Mochida, A., Hayashi, Y., and Sakamoto, S. (1992), "Numerical study on velocity-pressure field and wind forces for bluff bodies by k-ε, ASM and LES", *J. Wind Eng. & Ind. Aerodyn.*, vol. 44, pp. 2841-2852.

Murakami, S. and Mochida, A. (1995), "On turbulent vortex shedding flow past 2D square cylinder predicted by CFD", *J. Wind Eng. & Ind. Aerodyn.*, vol. 54&55, pp. 191-211.

Murakami, S., Mochida, A., and Iizuka, S. (1996), "New trends in turbulence models for prediction of wind effects on structures", *IWEF Workshop on CWE/CFD for Prediction of Wind Effects on Structures*, Fort Collins, CO, USA.

Murakami, S., Tsuchiya, M., Mochida, A., Kondo, K., and Ishida, Y. (1997), "Development of a new k-ε model for flow and pressure fields around bluff body", *J. Wind Eng. & Ind. Aerodyn.*, vol. 67&68, pp. 169-182.

Nikitin, N.V., Nicoud, F., Wasistho, B., Squires, K.D., and Spalart, P.R. (2000), "An approach to wall modeling in large-eddy simulations", *Physics of Fluids*, vol. 12, no. 7, pp. 1629-1632.

Norris, L.H. and Reynolds, W.C. (1975), "Turbulent channel flow with a moving wavy

boundary", *Rept. No. FM-10*, Dept. of Mech. Eng., Stanford University, Stanford, CA, USA.

Okada, H., Ha, Y.-C. (1992), "Comparison of wind tunnel and full-scale pressure measurement tests on the Texas Tech building", *J. Wind Eng. & Ind. Aerodyn.*, vol. 43, pp. 1601-1612.

Orlandi, P. (2001), *Fluid flow phenomena: A numerical toolkit*, Kluwer Academic Publishers, Dordrecht, Netherlands.

Patankar S.V. (1980), *Numerical heat transfer and fluid flow*, Hemisphere Publishing Co, Washington D.C., USA.

Paterson, D.A. and Apelt, C.J. (1986), "Computation of wind flows over three-dimensional buildings", *J. Wind Eng. & Ind. Aerodyn.*, vol. 24, pp. 193-213.

Paterson, D.A. and Holmes, J.D. (1992), "Computation of wind pressures on low-rise structures", *J. Wind Eng. & Ind. Aerodyn.*, vol. 43, pp. 1629-1640.

Pauley, L.L. (1988), *A numerical study of unsteady laminar boundary layer separation*, Ph.D. Dissertation, Stanford University, CA, USA.

Piomelli, U., Scotti, A., and Balaras, E. (2001), "Large-eddy simulations of turbulent flows, from desktop to supercomputer: Invited talk", *Vector and Parallel Processing - VECPAR 2000: 4th Int. Conf., Porto, Portugal, 2000* , G. Goos, J. Hartmanis, and J. van Leeuwen editors, pp. 551-557.

Pope, S.B. (2000), *Turbulent flows*, Cambridge University Press, Cambridge, UK.

Qasim A., Maxwell T.T., and Parameaswaran S. (1992), "Computational predictions of flow over a 2-D building", *J. Wind Eng. & Ind. Aerodyn.*, vol. 44, pp. 2839-2840.

Rodi, W. (1988), "Recent developments in turbulence modeling", *Proc. 3rd Int. Symposium on Refined Flow Modeling and Turbulence Measurements*, Y. Iwasa, N. Tamai, and A. Wada editors, Tokyo, Japan.

Rodi, W. (1997), "Comparison of LES and RANS calculations of the flow around bluff bodies", *J. Wind Eng. & Ind. Aerodyn.*, vol. 69&71, pp. 55-75.

Saha, A.K., Muralidhar, K., and Biswas, G. (2003), "Investigation of two- and three-dimensional models of transitional flow past a square cylinder", *J. of Eng. Mech., ASCE*, vol. 129, no. 11, pp. 1320-1329.

Schmidt, S. and Thiele, F. (2002), "Comparison of numerical methods applied to the flow over wall-mounted cubes", *Int. J. of Heat and Fluid Flow*, vol. 23, pp. 330-339.

Selvam, R.P. (1997), "Computation of pressures on Texas Tech University building using large eddy simulation", *J. Wind Eng. & Ind. Aerodyn.*, vol. 67&68, pp. 647-657.

Shur, M., Spalart, P.R., Strelets, M., and Travin, A. (1999), "Detached-eddy simulation of an airfoil at high angle of attack", *Proc. of 4th Int. Symp. Eng. Turb. Modeling and Measurements*, W. Rodi and D. Laurence editors, Corsica, France, 1999, pp. 669-678.

Simiu, E. and Scanlan, R.H. (1996), *Wind effects on structures*, John Wiley & Sons, Inc., USA.

Smagorinsky, J. (1963), "General circulation experiments with the primitive equation: I. The basic experiment", *Mon. Weather Rev.*, vol. 91, pp. 99-164.

Spalart, P.R., Allmaras, S.R. (1994), "A one-equation turbulence model for aerodynamic flows", *La Recherche Aerospatiale*, no. 1, pp. 5-21.

Spalart, P.R., Jou, W.-H., Strelets, M., and Allmaras, S.R. (1997), "Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach.", *1st AFOSR Int. Conf. on DNS/LES*, Ruston, LA, USA.

Spalding, D.B. (1972), "A novel finite difference formulation for differential expressions involving both first and second derivatives", *Int. J. Num. Meth. Eng.*, vol. 4, pp. 551-559.

Squires, K.D., Forsythe, J.R., Wurtzler, K.E., Strang, W.Z., Tomaro, R.F., Grismer, M.J., and Spalart, P.R. (2001), "Towards prediction of aircraft spin.", *High Performance Computing User's Group Conference*, Biloxi, MS, USA.

Strelets, M. (2001), "Detached eddy simulation of massively separated flows", *AIAA Paper*, no. 2001-0879.

Surry, D. (1991), "Pressure measurements on the Texas Tech building. Wind tunnel measurements and comparisons with full scale", *J. Wind Eng. & Ind. Aerodyn.*, vol. 38, pp. 235-247.

Tieleman, H.W., Surry, D., and Mehta, K.C. (1996), "Full/model-scale comparison of surface pressures on the Texas Tech experimental building", *J. Wind Eng. & Ind. Aerodyn.*, vol. 61, pp.1-23.

Travin, A., Shur, M., Strelets, M., and Spalart, P. (2000), "Detached-eddy simulations past a circular cylinder", *Flow, Turbulence and Combustion*, vol. 63, pp. 293-313.

Workshop on LES of bluff-body flows. (1995), organized by W. Rodi and J. Ferziger, Rottach-Egern, Germany.

Yeung, P.K. and Kot, S.C. (1985), "Computation of turbulent flows past arbitrary two-dimensional surface-mounted obstructions", *J. Wind Eng. & Ind. Aerodyn.*, vol. 18, pp.177-190.

Yoshizawa, A. (1993), "Bridging between eddy-viscosity-type and second-order models using a two-scale DIA", *Proc. of 9th Int. Symp. on Turbulent Shear Flow*, Kyoto, Japan, vol. 3, pp. 1-6.

Xiao, X., Edwards, J.R., Hassan, H.A., and Baurle, R.A. (2003), "Inflow boundary conditions for hybrid large eddy/Reynolds averaged Navier-Stokes simulations", *AIAA Journal*, vol. 41, no. 8, pp. 1481-1489.

APPENDIX

DESCRIPTION OF COMPUTER CODE

This appendix provides the developed computer code implementing the proposed hybrid TL/LES model. The listed code is written in the Fortran language. The flowchart the subroutines of this code are briefly described in Chapter 5. This program was developed to serve as a tool to investigate the performance of the proposed turbulence model.

SUBROUTINE MAIN

This is the main component of the program. It invokes several subroutines, which are subsequently listed. The arrays of all variables are converted from three-dimensional to one-dimensional to reduce computer storage storage/memory demands.

```
$DEBUG
C
C***** TWO-LAYER and LES MODEL
C
      PROGRAM MAIN
      USE PORTLIB
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAM.F'
```

```
C
C       DIMENSION REST(3)
        LOGICAL LUNST,LSTEADY
C
        LUNST=.TRUE.
        OPEN(4,FILE='STEADY.DAT')
        OPEN(88,FILE='TEST_TS.DAT')
C
C***** CALCULATE GEOMETRICAL QUANTITIES AND SET VARIABLES TO ZERO
        CALL GRID
C
C***** VARIABLES
        CALL VARI
C
C***** PARAMETERS AND CONTROL INDICES
        CALL CONTROL
C
C***** CONVERT WORKING ARRAY 2D TO 1D
C..... LI(I) : X-DIRECTION (STREAMWISE)
C..... LK(K) : K=DIRECTION (VERTICAL)
      NIJ=NI*NJ
      DO I=1,NI
        LI(I)=(I-1)*NJ
      ENDDO
      DO K=1,NK
        LK(K)=(K-1)*NIJ
      ENDDO
C
        CALL INIT
C
C***** INITIAL OUTPUT
C
C***** INITIAL CONDITION FOR INLET
C       U VELOCITY
        OPEN(70,FILE='INPUT.DAT')
        DO 100 K=2,NKM1
```

```
            DO 100 J=2,NJM1
            IJK=LK(K)+LI(2)+J
            U(IJK)=UIN*(Z(K)/H)**0.14
            V(IJK)=1.0/6.3*(LOG(Z(K))+0.68)
            W(IJK)=1.0/2.778*(LOG(Z(K))+1.508)
            TI(K)=0.145*(Z(K))**(-0.1521)
            TE(IJK-NJ)=3./2.*(U(IJK)*TI(K))**2.0
            TE(IJK)=TE(IJK-NJ)
            IF(J.EQ.NJ/2) THEN
            WRITE(70,77) K,Z(K),U(IJK),V(IJK),W(IJK),TI(K),TE(IJK-NJ)
            ENDIF
77          FORMAT(1X,I5,6(F10.5))
C           TE(IJK-NJ)=TEIN
100         CONTINUE
C***** RANDOM GENERATOR
C           DO K=2,NKM1
C           rru(k)=random(0)
C           rrv(k)=random(0)
C           rrw(k)=random(0)
C           rndu(k)=(rru(k)-0.5)*0.1
C           rndv(k)=(rrv(k)-0.5)*0.1
C           rndw(k)=(rrw(k)-0.5)*0.1
C           ENDDO
C           DO K=2,NKM1
C           DO J=2,NJM1
C           IJK=LK(K)+LI(2)+J
C           U(IJK)=U(IJK)+RNDU(K)
C           V(IJK)=RNDV(K)
C           W(IJK)=RNDW(K)
C           ENDDO
C           ENDDO
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C..... STARTING THE TIM LOOP
            BBETA=1.0
            TIM=0.0
            ITIM=0
```

```
      CALL VISCO

1     CONTINUE
C
      IF(TIM.LE.3.0)THEN
      DTIM=0.0004
      ELSEIF(TIM.GT.3.0.AND.TIM.LE.5.0) THEN
      DTIM=0.0004
c     ELSEIF(TIM.GT.5.0.AND.TIM.LE.20.0) THEN
c     DTIM=0.0004
      ENDIF
C
      TIM=TIM+DTIM
      ITIM=ITIM+1
C*********************************************************************
C     START THE INTERATION LOOP
C*********************************************************************
1000  CONTINUE
C
      IF(ITIM.LE.100) THEN
      SORMAX=1.E-4
      ELSEIF(ITIM.GT.10.AND.ITIM.LE.200) THEN
      SORMAX=1.E-4
      ELSE
      SORMAX=1.E-5
      ENDIF
C
      NITER=NITER+1
      CALL UMOM
      CALL VMOM
      CALL WMOM
      CALL CALCP
      CALL CALCTE
C
      CALL VISCO
```

```
C***** TERMINATION TESTS
      OPEN(111,FILE='ERR_HIS.DAT')
      ERR1=AMAX1(RESORU,RESORV,RESORW,RESORK)
      WRITE(111,*) NITER,RESORU,RESORV,RESORW,RESORK
      WRITE(*,*) NITER,ERR1
      IF(ERR1.GT.SORMAX) THEN
      GOTO 1000
      ELSE
      WRITE(*,600) ITIM,TIM
600   FORMAT(/,1X,'ITIM & TIM = ',I5,5X,F15.7)
      WRITE(*,601)
601   FORMAT(/,1X,' THE SOLUTION IS CONVERGED !!! ')
      GOTO 20
      ENDIF
C*************************************************************************
C       END OF ITERATION
C*************************************************************************
20    CONTINUE
C
C***** COMPUTE THE MEAN AND FLUCTUATION PARTS
      DO K=1,NK
      DO J=1,NJ
      DO I=1,NI
      IJK=LK(K)+LI(I)+J
      IF(ITIM.EQ.1) THEN
      UM(IJK)=U(IJK)
      VM(IJK)=V(IJK)
      WM(IJK)=W(IJK)
      PM(IJK)=P(IJK)
      TEM(IJK)=TE(IJK)
      ELSE
      UM(IJK)=UMO(IJK)+U(IJK)
      VM(IJK)=VMO(IJK)+V(IJK)
      WM(IJK)=WMO(IJK)+W(IJK)
      PM(IJK)=PMO(IJK)+P(IJK)
      TEM(IJK)=TEMO(IJK)+TE(IJK)
```

```
            ENDIF
            UMM(IJK)=UM(IJK)/FLOAT(ITIM)
            VMM(IJK)=VM(IJK)/FLOAT(ITIM)
            WMM(IJK)=WM(IJK)/FLOAT(ITIM)
            PMM(IJK)=PM(IJK)/FLOAT(ITIM)
            TEMM(IJK)=TEM(IJK)/FLOAT(ITIM)
            ENDDO
            ENDDO
            ENDDO
C
            IJKPP=LK(KSTP1)+LI(2)+NJ/2
            IJKP=LK(KSTP1)+LI(NIM1)+NJ/2
            WRITE(*,599) P(IJKPP),P(IJKP),PMM(IJKP),U(IJKP),UMM(IJKP)
599         FORMAT(1X,3(1X,F12.3),1X,2F10.5)

C..... CHECK THE STEADY SOLUTIONS
            ERRS=0.0
            IF(LUNST) THEN
            ERRSU=0.0
            ERRSV=0.0
            ERRSW=0.0
            DO J=2,NJM1
            DO I=2,NIM1
            IF(I.GE.ISTP1.AND.I.LE.ISTEP2) THEN
            KK=KSTP1
            ELSE
            KK=2
            ENDIF
            DO K=KK,NKM1
            IJK=LK(K)+LI(I)+J
C           RES=(U(IJK)-UO(IJK))**2+(V(IJK)-VO(IJK))**2+(W(IJK)-WO(IJK))**2
C           ERRS=ERRS+RES
            RESU=(U(IJK)-UO(IJK))**2
            RESV=(V(IJK)-VO(IJK))**2
            RESW=(W(IJK)-WO(IJK))**2
            ERRSU=ERRSU+RESU
```

```
            ERRSV=ERRSV+RESV
            ERRSW=ERRSW+RESW
            ENDDO
            ENDDO
            ENDDO
            ID=ISTP2-ISTEP1
            JD=JSTP2-JSTP1
            KD=KSTEP
C           ERRS=SQRT(ERRS/FLOAT((NI*NJ*NK)-(ID*JD*KD)))
          ERRSU=SQRT(ERRSU/FLOAT(NI*NJ*NK))
          ERRSV=SQRT(ERRSV/FLOAT(NI*NJ*NK))
          ERRSW=SQRT(ERRSW/FLOAT(NI*NJ*NK))
C           IF (ITIM.EQ.1) RES0=ERRS
C           RSMS=ERRS/RES0
            RSMS=AMAX1(ERRSU,ERRSV,ERRSW)
            WRITE(*,999) ITIM,TIM,RSMS
            WRITE(4,999) ITIM,TIM,RSMS
999         FORMAT(1X,I8,3X,F12.7,3X,F12.7)
C
            IF (RSMS.LT.RESMAX.OR.TIM.GT.5.0) THEN
            LSTEADY=.TRUE.
            WRITE(*,*) 'THE STEADY FLOW'
            GOTO 2000
            ENDIF
C..... PRINT OUT THE UNSTEADY SOLUTION
            CALL ARRANG
            CALL PRINTUN
            ENDIF
C
C..... ASSIGN NEW SOLUTIONS (U,V) TO (UO,VO)
            DO K=2,NKM1
            DO J=2,NJM1
            DO I=2,NIM1
            IJK=LK(K)+LI(I)+J
            UOO(IJK)=UO(IJK)
            VOO(IJK)=VO(IJK)
```

```
            WOO(IJK)=WO(IJK)
            UO(IJK)=U(IJK)
            VO(IJK)=V(IJK)
            WO(IJK)=W(IJK)
            TEOO(IJK)=TEO(IJK)
            TEO(IJK)=TE(IJK)
            PO(IJK)=P(IJK)
            IF(ITIM.EQ.1) THEN
            UMO(IJK)=U(IJK)
            VMO(IJK)=V(IJK)
            WMO(IJK)=W(IJK)
            PMO(IJK)=P(IJK)
            TEMO(IJK)=TE(IJK)
            ELSE
            UMO(IJK)=UM(IJK)
            VMO(IJK)=VM(IJK)
            WMO(IJK)=WM(IJK)
            PMO(IJK)=PM(IJK)
            TEMO(IJK)=TEM(IJK)
            ENDIF
            ENDDO
            ENDDO
            ENDDO

C..... BEGINS A NEW TIM STEP
            BBETA=1.0
            GOTO 1
C
C----- END OF OTERATION
C
C***** PRINT OUT THE STEADY SOLUTION (FINAL STEP)
2000   CONTINUE
            CALL ARRANG
            CALL PRINTST
            STOP
            END
```

PARAM.F

This subroutine determines all parameters and the sizes of arrays. To reduce storage in computer memory, all variables are defined in one-dimensional arrays.

```
 PARAMETER (NX=88,NY=30,NZ=52,NXYZ=NX*NY*NZ)
 COMMON
&/ALL/NI,NJ,NK,NIM1,NJM1,NKM1,NIM2,NJM2,NKM2,NITER
&/XGEOM/X(NX),DXEP(NX),DXPW(NX),SEW(NX),XU(NX)
&/YGEOM/Y(NY),DYNP(NY),DYPS(NY),SNS(NY),YV(NY)
&/ZGEOM/Z(NZ),DZTP(NZ),DZPB(NZ),STB(NZ),ZW(NZ),DZ
&/UVEL/RESORU,NSWPU,URFU,DXEPU(NX),DXPWU(NX),SEWU(NX),DU(NXYZ)
&/VVEL/RESORV,NSWPV,URFV,DYNPV(NY),DYPSV(NY),SNSV(NY),DV(NXYZ)
&/WVEL/RESORW,NSWPW,URFW,DZTPW(NZ),DZPBW(NZ),STBW(NZ),DW(NXYZ)
&/PCOR/RESORM,NSWPP,URFP,IPREF,JPREF,KPREF,KREFS,IPO1
&/TKE/ RESORK,NSWPK,URFK
&/VAR/U(NXYZ),V(NXYZ),W(NXYZ),P(NXYZ),PP(NXYZ),TE(NXYZ),ED(NXYZ),
&       VIS(NXYZ),VSGS(NXYZ),TES(NXYZ)
&/CP/CPF(NZ),CPFS(NZ),CPFO1(NZ),CPFS2(NZ),
&     CPR(NX),CPRS(NX),CPRO1(NX),CPRS2(NX),
&     CPB(NZ),CPBS(NZ),CPBO1(NZ),CPBS2(NZ),
&     CPF7F(NZ),CPFFS(NZ),CPFFO1(NZ),CPFFS2(NZ),
&     CPFR(NX),CPFRS(NX),CPFRO1(NX),CPFRS2(NX),
&     CPFB(NZ),CPFBS(NZ),CPFBO1(NZ),CPFBS2(NZ),
&     CPFY(NY),CPFYS(NY),CPFY1(NY),CPFYS2(NY),
&     CPSY(NX),CPSYS(NX),CPSY1(NX),CPSYS2(NX),
&     CPBY(NY),CPBYS(NY),CPBY1(NY),CPBYS2(NY)
&/FLUID/VISCOS,DENSIT,UIN,VIN,WIN,TEIN,EDIN,FLOWIN,FLOW,
       XMONIN
&/COEF/AP(NXYZ),AN(NXYZ),AS(NXYZ),AE(NXYZ),
&       AW(NXYZ),AT(NXYZ),AB(NXYZ),SU(NXYZ),SP(NXYZ)
&/CONTR/IMON,JMON,KMON,SORMAX,SORMAX1,RESMAX,GREAT,
         SMALL,IREF,JREF,KREF
&/TIME2/TIM,DTIM,APT(NXYZ),UO(NXYZ),VO(NXYZ),WO(NXYZ),
&       TEO(NXYZ),EDO(NXYZ),ITIMAX,ITIM,PPO(NXYZ),PO(NXYZ)
&/THREE/BBETA,UOO(NXYZ),VOO(NXYZ),WOO(NXYZ),TEOO(NXYZ),
         EDOO(NXYZ)
```

```
&/EXTRA/ZK(NZ),DX
&/CORREC/XX(NX),YY(NY),ZZ(NZ),US(NXYZ),VS(NXYZ),WS(NXYZ)
&/TRANS/XT(NX),XT1(NX),XT2(NX),XT3(NX),YT(NY),YT1(NY)
&/ONED/NIJK,LI(NXYZ),LK(NXYZ),NC,NIJ,RSM, XB(NX),XA(NX)
&/VIS/URFVIS,RE,PPSI(NXYZ)
&/TKE/GEN(NXYZ), SUKD(NXYZ),SPKD(NXYZ)
&/WALL/TAUB(NXYZ),TAUR(NXYZ),TAUL(NXYZ),
& TAVBB(NXYZ),TAVF(NXYZ),TAVB(NXYZ),
&TAWF(NXYZ),TAWB(NXYZ),TAWR(NXYZ),TAWL(NXYZ),
&ZPLUS(NXYZ),YPLUS(NXYZ),XPLUS(NXYZ),UPLUS(NXYZ)
&/WALL2/TAUT(NXYZ),TAUB(NXYZ),UTAUT(NXYZ),UTAUB(NXYZ),
        ZPPB(NZ),
&/KECONT/CMU,ELOG,CAPPA,C1,C2,CD,PRTE,PRED
&/LES/CS,UF(NXYZ),VF(NXYZ),WF(NXYZ),PF(NXYZ),VISOLD
&/FLUC/UFF(NXYZ),VFF(NXYZ),WFF(NXYZ)
&/ORI/UU(NXYZ),VV(NXYZ),WW(NXYZ)
&/BD/ISTEP1,ISTP1,ISTM1,ISTEP2,ISTP2,ISTM2,FRONT,BACK,ZSTEP,
&KSTEP,KSTM1,KSTP1,KSTP2,XMAX,YMAX,ZMAX,ZLARGE,HEIGHT,
  XINLET,JSTEP1,JSTP1,JSTM1,JSTEP2,JSTP2,JSTM2,IROOF
&/REY/RUU(NXYZ),RUV(NXYZ),RUW(NXYZ),RVV(NXYZ),RVW(NXYZ),RWW(NXYZ)
&/NOND/XN(NX),YN(NY),ZN(NZ),DSM,SMK,H,VOR(NXYZ)
&/RANDOM/RRU(NZ),RRV(NZ),RRW(NZ),RNDU(NZ),RNDV(NZ),RNDW(NZ)
&/MM/UM(NXYZ),VM(NXYZ),WM(NXYZ),UMM(NXYZ),VMM(NXYZ),WMM(NXYZ),
&       USMM(NXYZ),VSMM(NXYZ),WSMM(NXYZ),PM(NXYZ),PMM(NXYZ)
&/TT/CK,CE,DP(NXYZ),TEM(NXYZ),TEMM(NXYZ)
&/MEAN/UMO(NXYZ),VMO(NXYZ),WMO(NXYZ),PMO(NXYZ),
        TEMO(NXYZ),TI(NZ)
```

SUBROUTINE GRID

This subroutine computes grids in x- ,y-, and z- directions. The dimensions of the bluff body are defined in here.

```
C
$DEBUG
      SUBROUTINE GRID
      INCLUDE 'PARAM.F'
      OPEN(100,FILE='GRIDXYZ.DAT')
      OPEN(101,FILE='X.DAT')
      OPEN(102,FILE='Y.DAT')
      OPEN(103,FILE='Z.DAT')
C     GEOMETRIC VARIABLES
      H=0.078
      ZMAX=5.0*H
      XINLET=10.0*H
      XWIDTH=2.35*H
      XOUTLET=20.*H
      XMAX=XINLET+XWIDTH+XOUTLET
C
      YWIDTH=3.52*H
      YL=5.*H
      YR=5.*H
      YMAX=YR+YWIDTH+YL
C
C***** LATERAL DIRECTION (J)
      JSTEP1=10
      JSTP1=JSTEP1+1
      JSTM1=JSTEP1-1
      DY1=YR/FLOAT(JSTEP1)
      Y(1)=-0.5*DY1
      DO J=2,JSTEP1-1
      Y(J)=Y(J-1)+DY1
      ENDDO
C     JROOF=10
```

```
      JSTEP2=JROOF+JSTEP1
!THE TOTAL NUMBER OF GRIDS UP TO THE BACK OF BUILDING
      JSTP2=JSTEP2+1
      JSTM2=JSTEP2-1
      DY2=YWIDTH/FLOAT(JROOF)
      Y(JSTEP1)=YR-0.5*DY2
      Y(JSTP1)=YR+0.5*DY2
      DO J=JSTP1+1,JSTEP2-1
      Y(J)=Y(J-1)+DY2
      ENDDO
C
      JL=10
      NJ=JSTEP1+JROOF+JL
      NJM1=NJ-1
      DY3=YL/JL
      Y(JSTEP2)=YL+YWIDTH-0.5*DY3
      Y(JSTP2)=YL+YWIDTH+0.5*DY3
      DO J=JSTP2+1,NJ
      Y(J)=Y(J-1)+DY3
      ENDDO
C     HORIZONTAL DIRECTION UP TO THE FRONT WALL
      ISTEP1=42
      ISTP1=ISTEP1+1
      ISTM1=ISTEP1-1
      DX1=XINLET/FLOAT(ISTEP1)
      X(1)=-0.5*DX1
      DO I=2,ISTM1
      X(I)=X(I-1)+DX1
      WRITE(*,*) I,X(I)
      ENDDO
      IROOF=26
      ISTEP2=IROOF+ISTEP1
      !THE TOTAL NUMBER OF GRIDS UP TO THE BACK OF BUILDING
      ISTP2=ISTEP2+1
      ISTM2=ISTEP2-1
      DX2=XWIDTH/FLOAT(IROOF)
```

```
      X(ISTEP1)=XINLET-0.5*DX2
      X(ISTP1)=XINLET+0.5*DX2
      DO I=ISTP1+1,ISTEP2-1
      X(I)=X(I-1)+DX2
      WRITE(*,*) I,X(I)
      ENDDO
C
      IOUT=20
      NI=ISTEP1+IROOF+IOUT
      NIM1=NI-1
      EPSX=1.1
      SUMX=0.5*EPSX**(IOUT-4)+(EPSX**(IOUT-3)-1.0)/(EPSX-1.0)+0.5
      DX3=XOUTLET/SUMX
      X(ISTEP2)=(XINLET+XWIDTH)-0.1*DX3
      X(ISTP2)=(XINLET+XWIDTH)+0.1*DX3
      DO I=ISTP2+1,NIM1
      X(I)=X(I-1)+DX3
      DX3=EPSX*DX3
      WRITE(*,*) I,X(I)
      ENDDO
      X(NI)=X(NIM1)-X(NIM1-1)+X(NIM1)
C..... VERTICAL DIRECTION(K)
      NK=52
      NKM1=NK-1
      KSTEP=21
      KSTP1=KSTEP+1
      KSTP2=KSTEP+2
      KSTM1=KSTEP-1
C
      DZ=H/FLOAT(KSTM1)
      Z(1)=-DZ
      Z(2)=-Z(1)
      DO K=3,KSTM1
      Z(K)=Z(K-1)+DZ
      ENDDO
      NKS=NK-KSTEP
```

```
      EPSZ=1.10
      SUMZ=0.5*EPSZ**(NKS-4)+(EPSZ**(NKS-3)-1.0)/(EPSZ-1.0)+0.5
      DZ=(ZMAX-H)/SUMZ
      Z(KSTEP)=H-DZ
      Z(KSTP1)=H+DZ
      DO K=KSTP1+1,NKM1
      Z(K)=Z(K-1)+DZ
      DZ=EPSZ*DZ
      WRITE(*,*) K,Z(K)
      ENDDO
      Z(NK)=Z(NKM1)-Z(NKM1-1)+Z(NKM1)
C..... PRINT OUT
      WRITE(*,*) ISTEP1,X(ISTEP1),ISTEP2,X(ISTEP2),
      &                    KSTEP,Z(KSTEP)
C
      do i=1,ni
      DO J=1,NJ
      do k=1,nk
      write(100,110) i,J,k,x(i),Y(J),z(k)
      enddo
      enddo
      ENDDO
110   format(1x,3i5,3(1x,f7.3))

      DO I=1,NI
      WRITE(101,111) I,X(I)
      ENDDO
      DO J=1,NJ
      WRITE(102,111) J,Y(J)
      ENDDO
      DO K=1,NK
      WRITE(103,111) K,Z(K)
      ENDDO
111   FORMAT(1X,I5,F10.5)
      CLOSE(101)
      CLOSE(102)
```

```
CLOSE(103)
CLOSE(100)
RETURN
end
```

SUBROUTINE INIT

This subroutine generates the staggered grid system in x-, y-, and z- directions as referred in Figure 3.2.

```
C
      SUBROUTINE INIT
      INCLUDE 'PARAM.F'
C..... X-PLANE
      DXPW(1)=0.0
      DXEP(NI)=0.0
      DO I=1,NIM1
      DXEP(I)=X(I+1)-X(I)
      DXPW(I+1)=DXEP(I)
      ENDDO
C..... Y-PLANE
      DYPS(1)=0.0
      DYNP(NJ)=0.0
      DO J=1,NJM1
      DYNP(J)=Y(J+1)-Y(J)
      DYPS(J+1)=DYNP(J)
      ENDDO
C..... Z-PLANE
      DZPB(1)=0.0
      DZTP(NK)=0.0
      DO K=1,NKM1
      DZTP(K)=Z(K+1)-Z(K)
      DZPB(K+1)=DZTP(K)
      ENDDO
C..... AREAS OF CELL
C.....X
      SEW(1)=0.0
      SEW(NI)=0.0
      DO I=2,NIM1
      SEW(I)=0.5*(DXEP(I)+DXPW(I))
      ENDDO
```

```
C.....Y
      SNS(1)=0.0
      SNS(NJ)=0.0
      DO J=2,NJM1
      SNS(J)=0.5*(DYNP(J)+DYPS(J))
      ENDDO
C.....Z
      STB(1)=0.0
      STB(NK)=0.0
      DO K=2,NKM1
      STB(K)=0.5*(DZTP(K)+DZPB(K))
      ENDDO
C
      DO K=2,NKM1
      DO J=2,NJM1
      DO I=2,NIM1
      SMI=AMIN1(SEW(I),SNS(J),STB(K))
      SMJ=AMIN1(SMI,SEW(I),SNS(J),STB(K))
      SMK=AMIN1(SMJ,SEW(I),SNS(J),STB(K))
      ENDDO
      ENDDO
      ENDDO
C.....X
      XU(1)=0.0
      DO I=2,NI
      XU(I)=0.5*(X(I)+X(I-1))
      ENDDO
      DXPWU(1)=0.0
      DXPWU(2)=0.0
      DXEPU(1)=0.0
      DXEPU(NI)=0.0
      DO I=2,NIM1
      DXEPU(I)=XU(I+1)-XU(I)
      DXPWU(I+1)=DXEPU(I)
      ENDDO
      SEWU(1)=0.0
```

```
      DO I=2,NI
      SEWU(I)=(X(I)-X(I-1))
      ENDDO
C.....Y
      YV(1)=0.0
      DO J=2,NJ
      YV(J)=0.5*(Y(J)+Y(J-1))
      ENDDO
      DYPSV(1)=0.0
      DYPSV(2)=0.0
      DYNPV(1)=0.0
      DYNPV(NJ)=0.0
      DO J=2,NJM1
      DYNPV(J)=YV(J+1)-YV(J)
      DYPSV(J+1)=DYNPV(J)
      ENDDO
      SNSV(1)=0.0
      DO J=2,NJ
      SNSV(J)=(Y(J)-Y(J-1))
      ENDDO
C.....Z
      ZW(1)=0.0
      DO K=2,NK
      ZW(K)=0.5*(Z(K)+Z(K-1))
      ENDDO
      DZPBW(1)=0.0
      DZPBW(2)=0.0
      DZTPW(1)=0.0
      DZTPW(NK)=0.0
      DO K=2,NKM1
      DZTPW(K)=ZW(K+1)-ZW(K)
      DZPBW(K+1)=DZTPW(K)
      ENDDO
      STBW(1)=0.0
      DO K=2,NK
      STBW(K)=(Z(K)-Z(K-1))
```

```
      ENDDO
      RETURN
      END
C
```

SUBROUTINE ZERO

This subroutine initializes the whole domain to zero values. Before the first run of the program, the whole computational domain set to zero except the inlet boundary.

```
      SUBROUTINE ZERO
      INCLUDE 'PARAM.F'
C..... SET VARIABLES TO ZERO
C
      DO 200 J=2,NJM1
      DO 200 I=2,NIM1
      IF((I.GE.ISTP1.AND.I.LE.ISTEP2).AND.(J.GE.JSTP1.AND.J.LE.JSTEP2))
     &      THEN
      K2=KSTP1
      ELSE
      K2=2
      ENDIF
      DO 200 K=K2,NKM1
      IJK=LK(K)+LI(I)+J
C
      U(IJK)=UIN*(Z(K)/H)**0.14
      TI(K)=0.145*(Z(K))**(-0.1521)
      TE(IJK)=3./2.*(U(IJK)*TI(K))**2.0
C
      VIS(IJK)=VISCOS
      UO(IJK)=0.0
      UOO(IJK)=0.0
C     TE(IJK)=0.0
      TEO(IJK)=0.0
      TEOO(IJK)=0.0
      V(IJK)=0.0
      VO(IJK)=0.0
      VOO(IJK)=0.0
      W(IJK)=0.0
      WO(IJK)=0.0
      WOO(IJK)=0.0
```

```
            P(IJK)=0.0
            PP(IJK)=0.0
            SU(IJK)=0.0
            SP(IJK)=0.0
            SUKD(IJK)=0.0
            SPKD(IJK)=0.0
            DU(IJK)=0.0
            DV(IJK)=0.0
            DW(IJK)=0.0
C
            UM(IJK)=0.0
            UMM(IJK)=0.0
            VM(IJK)=0.0
            VMM(IJK)=0.0
            WMM(IJK)=0.0
            WM(IJK)=0.0
            TEM(IJK)=0.0
            TEMM(IJK)=0.0
            PMM(IJK)=0.0
  200 CONTINUE
            RETURN
            END
```

SUBROUTINE VARI

This subroutine defines the constants and reference values of the hybrid TL/LES model.

```
$DEBUG
C
      SUBROUTINE VARI
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAM.F'
C
C***** FLUID PROPERTIES AT 300K
      DENSIT=1.177
      VISCOS=1.846E-5
C
C***** TURBULENCE CONSTANTS
      CMU=0.09
      CD=1.0
      PRTE=1.0
      PRED=1.3
      CS=0.12
      CK=0.1
      CE=1.05
C***** REFERENCE BOUNDAY VALUES
      UIN=10.6
      VIN=0.1*UIN
      WIN=0.1*UIN
      TEIN=0.03*UIN**2
C     DTIM=0.0002
      RETURN
      END
```

SUBROUTINE CONTROL

This subroutine pre-defines the control parameters. The inner iteration step, the reference position, and the error criteria are defined.

```
$DEBUG
C
      SUBROUTINE CONTROL
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAM.F'
C
      NITER=0
      NSWPU=2
      NSWPV=2
      NSWPW=2
      NSWPP=3
      NSWPK=3
      NSWPE=3
C***** PRESSURE CORECCTION
      IPREF=2
      JPREF=NJ/2
      KREFS=KSTEP
C***** PROGRAM CONTROL AND MONITOR
      SORMAX1=1.E-3
      SORMAX=1.E-5
      RESMAX=1.E-3
      GREAT=1.E30
      RETURN
      END
```

SUBROUTINE VISCO

This subroutine solves the eddy viscosity of the hybrid TL/LES model and computes the vertical dimensionless distance from the wall. The switching line is pre-defined as $z^+ = 106.0$. This switching line may be changed using the length scale at each time step.

```
$DEBUG
      SUBROUTINE VISCO
C     USE MSIMSL
      INCLUDE 'PARAM.F'
C
      DO 100 J=2,NJM1
      DO 100 I=2,NIM1
      IF(I.GE.ISTP1.AND.I.LE.ISTEP2.AND.J.GE.JSTP1.AND.J.LE.JSTEP2) THEN
      K2=KSTP1
      ELSE
      K2=2
      ENDIF
      DO 100 K=K2,NKM1
      IJK=LK(K)+LI(I)+J
      ZPLUS(IJK)=DENSIT*(CMU**0.25)*SQRT(TE(IJK))*(Z(K)-ZW(K2))/VISCOS
      IF(ZPLUS(IJK).LE.106.0) THEN
      REZ(IJK)=DENSIT*SQRT(TE(IJK))*(Z(K)-ZW(K2))/VISCOS
      CL=CAPPA*CMU**(-3./4.)
      ALU=CL*(Z(K)-ZW(K2))*(1.0-EXP(-REZ(IJK)/50.5))     ! TWOLAYER BY RODI
      VSGS(IJK)=CMU*SQRT(TE(IJK))*ALU
      ELSE
      DELTA=(SEW(I)*SNS(J)*STB(K))**(1./3.)
      VSGS(IJK)=CK*SQRT(TE(IJK))*DELTA
      ENDIF
      VIS(IJK)=DENSIT*VSGS(IJK)+VISCOS
100   CONTINUE
      RETURN
      END
```

SUBROUTINE UMOM

This subroutine solves *x*-momentum equation using the SIMPLE algorithm and the three-time level method. The *y*- and *z*- momentum equations have the same coding format as the x-momentum equation.

```
$DEBUG
      SUBROUTINE UMOM
      INCLUDE 'PARAM.F'
C
      DO 100 K=2,NKM1
      DO 100 J=2,NJM1
      DO 100 I=3,NIM1
      IF(I.GE.ISTP1.AND.I.LE.ISTP2.AND.J.GE.JSTP1.AND.J.LE.JSTEP2.AND.
     &   K.LE.KSTEP) THEN
      GOTO 100
      ENDIF
      IJK=LK(K)+LI(I)+J
      AREAE=SNS(J)*STB(K)
      AREAW=AREAE
      AREAT=SEWU(I)*SNS(J)
      AREAB=AREAT
      AREAN=SEWU(I)*STB(K)
      AREAS=AREAN
C
C----- CALCULATE CONVECTION COEFFICIENTS
      FE=0.5*DENSIT*(U(IJK+NJ)+U(IJK))*AREAE
      FW=0.5*DENSIT*(U(IJK)+U(IJK-NJ))*AREAW
      FN=0.5*DENSIT*(V(IJK+1)+V(IJK-NJ+1))*AREAN
      FS=0.5*DENSIT*(V(IJK)+V(IJK-NJ))*AREAS
      FT=0.5*DENSIT*(W(IJK+NIJ)+W(IJK+NIJ-NJ))*AREAT
      FB=0.5*DENSIT*(W(IJK)+W(IJK-NJ))*AREAB
C----- CALCULATE DIFFUSION COEFFICIENTS
      DE=VIS(IJK)*AREAE/DXEPU(I)
      DDW=VIS(IJK-NJ)*AREAW/DXPWU(I)
      VIST=0.25*(VIS(IJK)+VIS(IJK+NIJ)+VIS(IJK-NJ)+VIS(IJK-NJ+NIJ))
```

```
      VISB=0.25*(VIS(IJK)+VIS(IJK-NIJ)+VIS(IJK-NJ)+VIS(IJK-NJ-NIJ))
      DT=VIST*AREAT/DZTP(K)
      DB=VISB*AREAB/DZPB(K)
      VISN=0.25*(VIS(IJK)+VIS(IJK+1)+VIS(IJK-NJ)+VIS(IJK-NJ+1))
      VISS=0.25*(VIS(IJK)+VIS(IJK-1)+VIS(IJK-NJ)+VIS(IJK-NJ-1))
      DN=VISN*AREAN/DYNP(J)
      DS=VISS*AREAS/DYPS(J)
C
      PEE=FE/DE
      PEW=FW/DDW
      PEN=FN/DN
      PES=FS/DS
      PET=FT/DT
      PEB=FB/DB
C
C----- ASSEMBLE MAIN COEFFICIENTS
      AE(IJK)=DE*(1.0-0.5*ABS(PEE))+AMAX1(-FE,0.0)
      AW(IJK)=DDW*(1.0-0.5*ABS(PEW))+AMAX1( FW,0.0)
      AT(IJK)=DT*(1.0-0.5*ABS(PET))+AMAX1(-FT,0.0)
      AB(IJK)=DB*(1.0-0.5*ABS(PEB))+AMAX1( FB,0.0)
      AN(IJK)=DN*(1.0-0.5*ABS(PEN))+AMAX1(-FN,0.0)
      AS(IJK)=DS*(1.0-0.5*ABS(PES))+AMAX1( FS,0.0)
      DU(IJK)=0.50*(AREAE+AREAW)
C
C----- SOURCE TERM
C
      SMP=FN-FS+FE-FW+FT-FB
      CP=AMAX1(0.0,SMP)
      CPO=CP
      SP(IJK)=-CP
C..... THREE TIME LEVELS
      VOL=SEWU(I)*STB(K)*SNS(J)
      APT(IJK)=DENSIT*VOL/DTIM
      SU(IJK)=CPO*U(IJK)+DU(IJK)*(P(IJK-NJ)-P(IJK))
      SU(IJK)=SU(IJK)+APT(IJK)*((1.+BBETA)*UO(IJK)-0.5*BBETA*UOO(IJK))
C
```

```
C***** ADD TURBULENCE TERMS ON THE SOURCE TERM
      DUDXP=(U(IJK+NJ)-U(IJK))/SEW(I)
      DUDXM=(U(IJK)-U(IJK-NJ))/SEW(I-1)
      SU(IJK)=SU(IJK)+(VIS(IJK)*DUDXP-VIS(IJK-NJ)*DUDXM)/SEW(I)*VOL
      DWDXP=(W(IJK+NIJ)-W(IJK+NIJ-NJ))/DXPW(I)
      DWDXM=(W(IJK)-W(IJK-NJ))/DXPW(I)
      SU(IJK)=SU(IJK)+(VIST*DWDXP-VISB*DWDXM)/STB(K)*VOL
      DVDXP=(V(IJK+1)-V(IJK+1-NJ))/DXPW(I)
      DVDXM=(V(IJK)-V(IJK-NJ))/DXPW(I)
      SU(IJK)=SU(IJK)+(VISN*DVDXP-VISS*DVDXM)/SNS(J)*VOL
100   CONTINUE
C
C***** BOUNDARY CONDITIONS FOR U-MOMENTUM EQ.**************************
      CALL BOUNDU
C**********************************************************************
      RESORU=0.0
      DO 300 K=2,NKM1
      DO 300 J=2,NJM1
      DO 300 I=3,NIM1
      IJK=LI(I)+LK(K)+J
      IF(I.GE.ISTP1.AND.I.LE.ISTP2.AND.J.GE.JSTP1.AND.J.LE.JSTEP2.AND.
     &   K.LE.KSTEP) THEN
      GOTO 300
      ENDIF
      AP(IJK)=AE(IJK)+AW(IJK)+AT(IJK)+AB(IJK)+
     &          AN(IJK)+AS(IJK)-SP(IJK)
C..... THREE TIM LEVELS
      AP(IJK)=AP(IJK)+APT(IJK)*(1.+0.5*BBETA)
      DU(IJK)=DU(IJK)/AP(IJK)
      RESOR=AE(IJK)*U(IJK+NJ)+AW(IJK)*U(IJK-NJ)+
     &          AN(IJK)*U(IJK+1)+AS(IJK)*U(IJK-1)+
     &          AT(IJK)*U(IJK+NIJ)+AB(IJK)*U(IJK-NIJ)-AP(IJK)*U(IJK)+SU(IJK)
      VOL=SEWU(I)*SNS(J)*STB(K)
      SORVOL=GREAT*VOL
      IF(-SP(IJK).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
      RESORU=RESORU+ABS(RESOR)**2
```

```
  300 CONTINUE
      RESORU=SQRT((RESORU)/NI/NJ/NK)
C
C     SOLUTION OF DIFFERENCE EQUATION
      DO I=1,NSWPU
      CALL TDMA(3,2,2,ISTP1,ISTP2,JSTP1,JSTEP2,KSTP1,U)
      ENDDO
      RETURN
      END
$DEBUG
```

SUBROUTINE VMOM

```
      SUBROUTINE VMOM
C     USE MSIMSL
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAM.F'
C
      DO 100 K=2,NKM1
      DO 100 J=3,NJM1
      DO 100 I=2,NIM1
      IF(I.GE.ISTP1.AND.I.LE.ISTEP2.AND.J.GE.JSTP1.AND.J.LE.JSTP2.AND.
     &   K.LE.KSTEP) THEN
      GOTO 100
      ENDIF
      IJK=LK(K)+LI(I)+J
      AREAE=SNSV(J)*STB(K)
      AREAW=AREAE
      AREAT=SEW(I)*SNSV(J)
      AREAB=AREAT
      AREAN=SEW(I)*STB(K)
      AREAS=AREAN
C
C----- CALCULATE CONVECTION COEFFICIENTS
      FE=0.5*DENSIT*(U(IJK+NJ)+U(IJK+NJ-1))*AREAE
      FW=0.5*DENSIT*(U(IJK)+U(IJK-1))*AREAW
      FN=0.5*DENSIT*(V(IJK+1)+V(IJK))*AREAN
      FS=0.5*DENSIT*(V(IJK)+V(IJK-1))*AREAS
      FT=0.5*DENSIT*(W(IJK+NIJ)+W(IJK+NIJ-1))*AREAT
      FB=0.5*DENSIT*(W(IJK)+W(IJK-1))*AREAB
C
C----- CALCULATE DIFFUSION COEFFICIENTS
      VISE=0.25*(VIS(IJK)+VIS(IJK+NJ)+VIS(IJK-1)+VIS(IJK-1+NJ))
      VISW=0.25*(VIS(IJK)+VIS(IJK-NJ)+VIS(IJK-1)+VIS(IJK-1-NJ))
      DE=VISE*AREAE/DXEP(I)
      DDW=VISW*AREAW/DXPW(I)
```

```
      DN=VIS(IJK)*AREAN/DYNPV(J)
      DS=VIS(IJK-1)*AREAS/DYPSV(J)
      VIST=0.25*(VIS(IJK)+VIS(IJK+NIJ)+VIS(IJK-1)+VIS(IJK-1+NIJ))
      VISB=0.25*(VIS(IJK)+VIS(IJK-NIJ)+VIS(IJK-1)+VIS(IJK-1-NIJ))
      DT=VIST*AREAT/DZTP(K)
      DB=VISB*AREAB/DZPB(K)
C
      PEE=FE/DE
      PEW=FW/DDW
      PEN=FN/DN
      PES=FS/DS
      PET=FT/DT
      PEB=FB/DB
C----- ASSEMBLE MAIN COEFFICIENTS
      AE(IJK)=DE*(1.0-0.5*ABS(PEE))+AMAX1(-FE,0.0)
      AW(IJK)=DDW*(1.0-0.5*ABS(PEW))+AMAX1( FW,0.0)
      AT(IJK)=DT*(1.0-0.5*ABS(PET))+AMAX1(-FT,0.0)
      AB(IJK)=DB*(1.0-0.5*ABS(PEB))+AMAX1( FB,0.0)
      AN(IJK)=DN*(1.0-0.5*ABS(PEN))+AMAX1(-FN,0.0)
      AS(IJK)=DS*(1.0-0.5*ABS(PES))+AMAX1( FS,0.0)
      DV(IJK)=0.5*(AREAN+AREAS)
C
      SMP=FN-FS+FE-FW+FT-FB
      CP=AMAX1(0.0,SMP)
      CPO=CP
      SU(IJK)=CPO*V(IJK)+DV(IJK)*(P(IJK-1)-P(IJK))
C     THREE TIM LEVELS
      VOL=SEW(I)*SNSV(J)*STB(K)
      APT(IJK)=DENSIT*VOL/DTIM
      SU(IJK)=SU(IJK)+APT(IJK)*((1.+BBETA)*VO(IJK)-0.5*BBETA*VOO(IJK))
      SP(IJK)=-CP
C***** ADD TURBULENCE TERMS ON THE SOURCE TERM
      DUDYP=(U(IJK+NJ)-U(IJK+NJ-1))/DYPS(J)
      DUDYM=(U(IJK)-U(IJK-1))/DYPS(J)
      SU(IJK)=SU(IJK)+(VISE*DUDYP-VISW*DUDYM)/SEW(I)*VOL
      DVDYP=(V(IJK+1)-V(IJK))/SNS(J)
```

```
      DVDYM=(V(IJK)-V(IJK-1))/SNS(J-1)
      SU(IJK)=SU(IJK)+(VIS(IJK)*DVDYP-VIS(IJK-1)*DVDYM)/SNS(J)*VOL
      DWDYP=(W(IJK+NIJ)-W(IJK-1+NIJ))/DYPS(J)
      DWDYM=(W(IJK)-W(IJK-1))/DYPS(J)
      SU(IJK)=SU(IJK)+(VIST*DWDYP-VISB*DWDYM)/STB(K)*VOL
  100 CONTINUE
C
C***** BOUNDARY CONDITIONS FOR V-MOMENTUM EQ.**************************
      CALL BOUNDV
C
C***********************************************************************
      RESORV=0.0
      DO 300 K=2,NKM1
      DO 300 J=3,NJM1
      DO 300 I=2,NIM1
      IJK=LI(I)+LK(K)+J
      IF(I.GE.ISTP1.AND.I.LE.ISTEP2.AND.J.GE.JSTP1.AND.J.LE.JSTP2.AND.
     &   K.LE.KSTEP) THEN
      GOTO 300
      ENDIF
      AP(IJK)=AE(IJK)+AW(IJK)+AT(IJK)+AB(IJK)+
     &                AN(IJK)+AS(IJK)-SP(IJK)
C..... THREE TIM LEVELS
      AP(IJK)=AP(IJK)+APT(IJK)*(1.+0.5*BBETA)
      DV(IJK)=DV(IJK)/AP(IJK)
      RESOR=AE(IJK)*V(IJK+NJ)+AW(IJK)*V(IJK-NJ)+
     &      AN(IJK)*V(IJK+1)+AS(IJK)*V(IJK-1)+
     &      AT(IJK)*V(IJK+NIJ)+AB(IJK)*V(IJK-NIJ)-AP(IJK)*V(IJK)+SU(IJK)
      VOL=SEW(I)*SNSV(J)*STB(K)
      SORVOL=GREAT*VOL
      IF(-SP(IJK).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
      RESORV=RESORV+ABS(RESOR)**2
  300 CONTINUE
      RESORV=SQRT((RESORV)/NI/NJ/NK)
C
```

```
C
      DO I=1,NSWPV
      CALL TDMA(2,3,2,ISTP1,ISTEP2,JSTP1,JSTP2,KSTP1,V)
      ENDDO
      RETURN
      END
```

SUBROUTINE WMOM

```
$DEBUG
C
      SUBROUTINE WMOM
C     USE MSIMSL
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAM.F'
C
      DO 100 K=3,NKM1
      DO 100 J=2,NJM1
      DO 100 I=2,NIM1
      IF(I.GE.ISTP1.AND.I.LE.ISTEP2.AND.J.GE.JSTP1.AND.J.LE.JSTEP2.AND.
     &   K.LE.KSTP1) THEN
      GOTO 100
      ENDIF
      IJK=LK(K)+LI(I)+J
      AREAE=SNS(J)*STBW(K)
      AREAW=AREAE
      AREAT=SEW(I)*SNS(J)
      AREAB=AREAT
      AREAN=SEW(I)*STBW(K)
      AREAS=AREAN
C
C..... CALCULATE CONVECTION COEFFICIENTS
      FE=0.5*DENSIT*(U(IJK+NJ)+U(IJK+NJ-NIJ))*AREAE
      FW=0.5*DENSIT*(U(IJK)+U(IJK-NIJ))*AREAW
      FN=0.5*DENSIT*(V(IJK+1)+V(IJK+1-NIJ))*AREAN
      FS=0.5*DENSIT*(V(IJK)+V(IJK-NIJ))*AREAS
      FT=0.5*DENSIT*(W(IJK+NIJ)+W(IJK))*AREAT
      FB=0.5*DENSIT*(W(IJK)+W(IJK-NIJ))*AREAB
C
C..... CALCULATE DIFFUSION COEFFICIENTS
      VISE=0.25*(VIS(IJK)+VIS(IJK+NJ)+VIS(IJK-NIJ)+VIS(IJK-NIJ+NJ))
      VISW=0.25*(VIS(IJK)+VIS(IJK-NJ)+VIS(IJK-NIJ)+VIS(IJK-NIJ-NJ))
```

```
      DE=VISE*AREAE/DXEP(I)
      DDW=VISW*AREAW/DXPW(I)
      VISN=0.25*(VIS(IJK)+VIS(IJK+1)+VIS(IJK-NIJ)+VIS(IJK-NIJ+1))
      VISS=0.25*(VIS(IJK)+VIS(IJK-1)+VIS(IJK-NIJ)+VIS(IJK-NIJ-1))
      DN=VISN*AREAN/DYNP(J)
      DS=VISS*AREAS/DYPS(J)
      DT=VIS(IJK)*AREAT/DZTPW(K)
      DB=VIS(IJK-NIJ)*AREAB/DZPBW(K)
C
      PEE=FE/DE
      PEW=FW/DDW
      PEN=FN/DN
      PES=FS/DS
      PET=FT/DT
      PEB=FB/DB
C..... ASSEMBLE MAIN COEFFICIENTS
      AE(IJK)=DE*(1.0-0.5*ABS(PEE))+AMAX1(-FE,0.0)
      AW(IJK)=DDW*(1.0-0.5*ABS(PEW))+AMAX1( FW,0.0)
      AT(IJK)=DT*(1.0-0.5*ABS(PET))+AMAX1(-FT,0.0)
      AB(IJK)=DB*(1.0-0.5*ABS(PEB))+AMAX1( FB,0.0)
      AN(IJK)=DN*(1.0-0.5*ABS(PEN))+AMAX1(-FN,0.0)
      AS(IJK)=DS*(1.0-0.5*ABS(PES))+AMAX1( FS,0.0)
      DW(IJK)=0.5*(AREAT+AREAB)
C
      SMP=FN-FS+FE-FW+FT-FB
      CP=AMAX1(0.0,SMP)
      CPO=CP
      SU(IJK)=CPO*W(IJK)+DW(IJK)*(P(IJK-NIJ)-P(IJK))
C..... THREE TIM LEVELS
      VOL=SEW(I)*SNS(J)*STBW(K)
      APT(IJK)=DENSIT*VOL/DTIM
      SU(IJK)=SU(IJK)+APT(IJK)*((1.+BBETA)*WO(IJK)-0.5*BBETA*WOO(IJK))
      SP(IJK)=-CP
C
C***** ADD TURBULENCE TERMS ON THE SOURCE TERM
      DUDZP=(U(IJK+NJ)-U(IJK+NJ-NIJ))/DZPB(K)
```

```
          DUDZM=(U(IJK)-U(IJK-NIJ))/DZPB(K)
          SU(IJK)=SU(IJK)+(VISE*DUDZP-VISW*DUDZM)/SEW(I)*VOL
          DVDZP=(V(IJK+1)-V(IJK+1-NIJ))/DZPB(K)
          DVDZM=(V(IJK)-V(IJK-NIJ))/DZPB(K)
          SU(IJK)=SU(IJK)+(VISN*DVDZP-VISS*DVDZM)/SNS(J)*VOL
          DWDZP=(W(IJK+NIJ)-W(IJK))/STB(K)
          DWDZM=(W(IJK)-W(IJK-NIJ))/STB(K-1)
          SU(IJK)=SU(IJK)+(VIS(IJK)*DWDZP-VIS(IJK-NIJ)*DWDZM)/STB(K)*VOL
  100 CONTINUE
C
C***** BOUNDARY CONDITIONS FOR W-MOMENTUM EQ.**************************
          CALL BOUNDW
C
C*********************************************************************
          RESORW=0.0
          DO 300 K=3,NKM1
          DO 300 J=2,NJM1
          DO 300 I=2,NIM1
          IJK=LI(I)+LK(K)+J
          IF(I.GE.ISTP1.AND.I.LE.ISTEP2.AND.J.GE.JSTP1.AND.J.LE.JSTEP2.AND.
          &     K.LE.KSTP1) THEN
          GOTO 300
          ENDIF
          AP(IJK)=AE(IJK)+AW(IJK)+AT(IJK)+AB(IJK)+AN(IJK)+AS(IJK)-SP(IJK)
C..... THREE TIM LEVELS
          AP(IJK)=AP(IJK)+APT(IJK)*(1.+0.5*BBETA)
          DW(IJK)=DW(IJK)/AP(IJK)
          RESOR=AE(IJK)*W(IJK+NJ)+AW(IJK)*W(IJK-NJ)+
          &              AN(IJK)*W(IJK+1)+AS(IJK)*W(IJK-1)+
          &              AT(IJK)*W(IJK+NIJ)+AB(IJK)*W(IJK-NIJ)-AP(IJK)*W(IJK)+SU(IJK)
          VOL=SEW(I)*SNS(J)*STBW(K)
          SORVOL=GREAT*VOL
          IF(-SP(IJK).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
          RESORW=RESORW+ABS(RESOR)**2
  300 CONTINUE
          RESORW=SQRT((RESORW)/NI/NJ/NK)
```

```
C
CHAPTER 4. SOLUTION OF DIFFERENCE EQUATION
C
      DO I=1,NSWPW
      CALL TDMA(2,2,3,ISTP1,ISTEP2,JSTP1,JSTEP2,KSTEP+2,W)
      ENDDO
      RETURN
      END
```

SUBROUTINE CALCP

This subroutine computes the pressure-correction combined with *u*-, *v*- , *w*- momentum equation.

```
$DEBUG
C
      SUBROUTINE CALCP
C     USE MSIMSL
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAM.F'
C
      RESORM=0.0
      DO 100 K=2,NKM1
      DO 100 J=2,NJM1
      DO 100 I=2,NIM1
C
      IJK=LK(K)+LI(I)+J
      IF(I.GE.ISTP1.AND.I.LE.ISTEP2.AND.J.GE.JSTP1.AND.J.LE.JSTEP2.AND.
     &    K.LE.KSTEP) THEN
      GOTO 100
      ENDIF
      AREAE=SNS(J)*STB(K)
      AREAW=AREAE
      AREAT=SEW(I)*SNS(J)
      AREAB=AREAT
      AREAN=SEW(I)*STB(K)
      AREAS=AREAN
C---- CALCULATE COEFFICIENTS
      AE(IJK)=DENSIT*AREAE*DU(IJK+NJ)
      AW(IJK)=DENSIT*AREAW*DU(IJK)
      AN(IJK)=DENSIT*AREAN*DV(IJK+1)
      AS(IJK)=DENSIT*AREAS*DV(IJK)
      AT(IJK)=DENSIT*AREAT*DW(IJK+NIJ)
      AB(IJK)=DENSIT*AREAB*DW(IJK)
C
```

```
C----- CALCULATE SOURCE TERMS
      FE=DENSIT*AREAE*U(IJK+NJ)
      FW=DENSIT*AREAW*U(IJK)
      FN=DENSIT*AREAN*V(IJK+1)
      FS=DENSIT*AREAS*V(IJK)
      FT=DENSIT*AREAT*W(IJK+NIJ)
      FB=DENSIT*AREAB*W(IJK)
      SMP=FN-FS+FE-FW+FT-FB
      SP(IJK)=0.0
      SU(IJK)=-SMP
C
C----- COMPUTE SUM OF ABSOLUTE SOURCES
      RESORM=RESORM+ABS(SMP)**2
  100 CONTINUE
      RESORM=SQRT((RESORM)/NI/NJ/NK)
C
      DO 300 K=2,NKM1
      DO 300 J=2,NJM1
      DO 300 I=2,NIM1
      IJK=LI(I)+LK(K)+J
      IF(I.GE.ISTP1.AND.I.LE.ISTEP2.AND.J.GE.JSTP1.AND.J.LE.JSTEP2.AND.
     &   K.LE.KSTEP) THEN
      GOTO 300
      ENDIF
      AP(IJK)=AE(IJK)+AW(IJK)+AT(IJK)+AB(IJK)+
     &                  AN(IJK)+AS(IJK)-SP(IJK)
  300 CONTINUE
C
      DO N=1,NSWPP
      CALL TDMA(2,2,2,ISTP1,ISTEP2,JSTP1,JSTEP2,KSTP1,PP)
      ENDDO
C----- VELOCITIES
      DO 500 K=2,NKM1
      DO 500 J=2,NJM1
      DO 500 I=2,NIM1
      IJK=LK(K)+LI(I)+J
```

```
            IF(I.GE.ISTP1.AND.I.LE.ISTEP2.AND.J.GE.JSTP1.AND.J.LE.JSTEP2.AND.
            &     K.LE.KSTEP) THEN
            GOTO 500
            ELSE
            IF (I.NE.2) U(IJK)=U(IJK)+DU(IJK)*(PP(IJK-NJ)-PP(IJK))
          V(IJK)=V(IJK)+DV(IJK)*(PP(IJK-1)-PP(IJK))
          W(IJK)=W(IJK)+DW(IJK)*(PP(IJK-NIJ)-PP(IJK))
            ENDIF
  500 CONTINUE
C
C----- PRESSURES
            IJKPP=LK(KREFS)+LI(IPREF)+JPREF
            PPREF=PP(IJKPP)
            DO 510 K=2,NKM1
            DO 510 J=2,NJM1
            DO 510 I=2,NIM1
            IJK=LK(K)+LI(I)+J
            IF(I.GE.ISTP1.AND.I.LE.ISTEP2.AND.J.GE.JSTP1.AND.J.LE.JSTEP2.AND.
            &     K.LE.KSTEP) THEN
            GOTO 510
            ENDIF
C           P(IJK)=P(IJK)+URFP*(PP(IJK)-PPREF)
            P(IJK)=P(IJK)+(PP(IJK)-PPREF)
            PP(IJK)=0.0
  510 CONTINUE
            RETURN
            END
```

SUBROUTINE CALCTE

This subroutine computes the turbulent kinetic energy equation. The dissipation term is determined from the switching line, which is computed in subroutine VISCO, between two-layer model and the SGS one-equation model.

```
$DEBUG
C***** SUBROUTINE CALCTE
C
      SUBROUTINE CALCTE
C
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAM.F'
C
      DO 100 J=2,NJM1
      DO 100 I=2,NIM1
      IF((I.GE.ISTP1.AND.I.LE.ISTEP2).AND.(J.GE.JSTP1.AND.J.LE.JSTEP2))
      &THEN
      K2=KSTP1
      ELSE
      K2=2
      ENDIF
C
      DO 100 K=K2,NKM1
      AREAE=SNS(J)*STB(K)
      AREAW=AREAE
      AREAT=SEW(I)*SNS(J)
      AREAB=AREAT
      AREAN=SEW(I)*STB(K)
      AREAS=AREAN
      VOL=SEW(I)*SNS(J)*STB(K)
      IJK=LK(K)+LI(I)+J
C----- CALCULATE CONVECTION COEFFICIENTS
      FE=DENSIT*U(IJK+NJ)*AREAE
      FW=DENSIT*U(IJK)*AREAW
      FT=DENSIT*W(IJK+NIJ)*AREAT
```

```
      FB=DENSIT*W(IJK)*AREAB
      FN=DENSIT*V(IJK+1)*AREAN
      FS=DENSIT*V(IJK)*AREAS
C----- CALCULATE DIFFUSION COEFFICIENTS
      ZPLUS(IJK)=DENSIT*(CMU**0.25)*SQRT(TE(IJK))*(Z(K)-ZW(K2))/VISCOS
      IF(ZPLUS(IJK).LE.120.0) THEN
      GAME=0.5*(VIS(IJK)+VIS(IJK+NJ))/PRTE
      GAMW=0.5*(VIS(IJK)+VIS(IJK-NJ))/PRTE
      GAMT=0.5*(VIS(IJK)+VIS(IJK+NIJ))/PRTE
      GAMB=0.5*(VIS(IJK)+VIS(IJK-NIJ))/PRTE
      GAMN=0.5*(VIS(IJK)+VIS(IJK+1))/PRTE
      GAMS=0.5*(VIS(IJK)+VIS(IJK-1))/PRTE
      ELSE
      GAME=0.5*(VIS(IJK)+VIS(IJK+NJ))
      GAMW=0.5*(VIS(IJK)+VIS(IJK-NJ))
      GAMT=0.5*(VIS(IJK)+VIS(IJK+NIJ))
      GAMB=0.5*(VIS(IJK)+VIS(IJK-NIJ))
      GAMN=0.5*(VIS(IJK)+VIS(IJK+1))
      GAMS=0.5*(VIS(IJK)+VIS(IJK-1))
      ENDIF
      DE=GAME*AREAE/DXEP(I)
      DDW=GAMW*AREAW/DXPW(I)
      DT=GAMT*AREAT/DZTP(K)
      DB=GAMB*AREAB/DZPB(K)
      DN=GAMN*AREAN/DYNP(J)
      DS=GAMS*AREAS/DYPS(J)
C
C----- SOURCE TEMS
      SMP=FN-FS+FE-FW+FT-FB
      CP=AMAX1(0.0,SMP)
      CPO=CP
C
      DUDX=(U(IJK+NJ)-U(IJK))/SEW(I)
      DVDY=(V(IJK+1)-V(IJK))/SNS(J)
      DWDZ=(W(IJK+NIJ)-W(IJK))/STB(K)
      DUDY=(0.25*(U(IJK)+U(IJK+NJ)+U(IJK+NJ+1)+U(IJK+1))-
```

```
     &          0.25*(U(IJK)+U(IJK+NJ)+U(IJK+NJ-1)+U(IJK-1)))/SNS(J)
     DVDX=(0.25*(V(IJK)+V(IJK+1)+V(IJK+NJ+1)+V(IJK+NJ))-
     &          0.25*(V(IJK)+V(IJK+1)+V(IJK-NJ+1)+V(IJK-NJ)))/SEW(I)
     DVDZ=(0.25*(V(IJK)+V(IJK+1)+V(IJK+1+NIJ)+V(IJK+NIJ))-
     &          0.25*(V(IJK)+V(IJK+1)+V(IJK+1-NIJ)+V(IJK-NIJ)))/STB(K)
     DWDY=(0.25*(W(IJK)+W(IJK+NIJ)+W(IJK+1+NIJ)+W(IJK+1))-
     &          0.25*(W(IJK)+W(IJK+NIJ)+W(IJK-1+NIJ)+W(IJK-1)))/SNS(J)
     DUDZ=(0.25*(U(IJK)+U(IJK+NJ)+U(IJK+NJ+NIJ)+U(IJK+NIJ))-
     &          0.25*(U(IJK)+U(IJK+NJ)+U(IJK+NJ-NIJ)+U(IJK-NIJ)))/STB(K)
     DWDX=(0.25*(W(IJK)+W(IJK+NIJ)+W(IJK+NJ+NIJ)+W(IJK+NJ))-
     &          0.25*(W(IJK)+W(IJK+NIJ)+W(IJK-NJ+NIJ)+W(IJK-NJ)))/SEW(I)
     SABS=SQRT(2.0*(DUDX**2.+DVDY**2.+DWDZ**2.)+
     &           (DUDZ+DWDZ)**2.+(DUDY+DVDX)**2.+(DVDZ+DWDY)**2.)
C
C..... TURBULENT GENERATION TERM
     GEN(IJK)=VIS(IJK)*SABS*SABS
C
C..... ASSENMBLE MAIN COEFFICIENTS
     PEE=FE/DE
     PEW=FW/DDW
     PEN=FN/DN
     PES=FS/DS
     PET=FT/DT
     PEB=FB/DB

     AE(IJK)=DE*AMAX1(0.0,(1.-0.1*ABS(PEE))**5)+AMAX1(-FE,0.0)
     AW(IJK)=DDW*AMAX1(0.0,(1.-0.1*ABS(PEW))**5)+AMAX1( FW,0.0)
     AT(IJK)=DT*AMAX1(0.0,(1.-0.1*ABS(PET))**5)+AMAX1(-FT,0.0)
     AB(IJK)=DB*AMAX1(0.0,(1.-0.1*ABS(PEB))**5)+AMAX1( FB,0.0)
     AN(IJK)=DN*AMAX1(0.0,(1.-0.1*ABS(PEN))**5)+AMAX1(-FN,0.0)
     AS(IJK)=DS*AMAX1(0.0,(1.-0.1*ABS(PES))**5)+AMAX1( FS,0.0)
C
C
C..... SOURCE TERMS
     SU(IJK)=CPO*TE(IJK)
     SUKD(IJK)=SU(IJK)
```

```
      SU(IJK)=SU(IJK)+GEN(IJK)*VOL
      SP(IJK)=-CP
      SPKD(IJK)=SP(IJK)
C..... THREE TIME LEVELS
      APT(IJK)=DENSIT*VOL/DTIM
      SU(IJK)=SU(IJK)+APT(IJK)*((1.+BBETA)*TEO(IJK)-0.5*BBETA*TEOO(IJK))
C     ZPLUS(IJK)=DENSIT*(CMU**0.25)*SQRT(TE(IJK))*(Z(K)-ZW(K2))/VISCOS
      IF(ZPLUS(IJK).LE.120.0) THEN
C     VAN DRIEST MIXING LENGTH MODEL
C     AML=CAPPA*(Z(K)-ZW(K2))*(1.0-EXP(-ZPLUS(IJK)/26.0))
C     NORRIS AND REYNOLDS MODEL
      CL=CAPPA*CMU**(-3./4.)
      REZ(IJK)=DENSIT*SQRT(TE(IJK))*(Z(K)-ZW(K2))/VISCOS
      ALE=CL*(Z(K)-ZW(K2))/(1.0+5.3/REZ(IJK))
      SP(IJK)=SP(IJK)-DENSIT*(TE(IJK)**1.5)/ALE*VOL
      ELSE
C     DELTA=AMIN1(SEW(I),SNS(J),STB(K))
      DELTA=(SEW(I)*SNS(J)*STB(K))**(1./3.)
      SP(IJK)=SP(IJK)-DENSIT*CE*(TE(IJK)**1.5)/DELTA*VOL
      ENDIF
100   CONTINUE
C
      CALL BOUNDK
C
      RESORK=0.0
      DO 300 J=2,NJM1
      DO 300 I=2,NIM1
C
      IF((I.GE.ISTP1.AND.I.LE.ISTEP2).AND.(J.GE.JSTP1.AND.J.LE.JSTEP2))
      &THEN
      K2=KSTP1
      ELSE
      K2=2
      ENDIF
C
      DO 300 K=K2,NKM1
```

```
      IJK=LI(I)+LK(K)+J
      AP(IJK)=AE(IJK)+AW(IJK)+AT(IJK)+AB(IJK)+
     &                AN(IJK)+AS(IJK)-SP(IJK)
C..... THREE TIME LEVELS
      AP(IJK)=AP(IJK)+APT(IJK)*(1.+0.5*BBETA)
      RESOR=AE(IJK)*TE(IJK+NJ)+AW(IJK)*TE(IJK-NJ)+
     &      AN(IJK)*TE(IJK+1)+AS(IJK)*TE(IJK-1)+
     &      AT(IJK)*TE(IJK+NIJ)+AB(IJK)*TE(IJK-NIJ)-
     &      AP(IJK)*TE(IJK)+SU(IJK)
C
      VOL=SEW(I)*SNS(J)*STB(K)
      SORVOL=GREAT*VOL
      IF(-SP(IJK).GT.0.50*SORVOL) RESOR=RESOR/SORVOL
      RESORK=RESORK+ABS(RESOR)**2
C
C----- UNDER-RELAXATION
C     WRITE(*,*) I,J,K,AP(IJK)
C     AP(IJK)=AP(IJK)/URFK
C     SU(IJK)=SU(IJK)+(1.0-URFK)*AP(IJK)*TE(IJK)
  300 CONTINUE
      RESORK=SQRT((RESORK)/NI/NJ/NK)
C
CHAPTER 4. SOLUTION OF DIFFERENCE EQUATION
C
      DO I=1,NSWPK
      CALL TDMA(2,2,2,ISTP1,ISTEP2,JSTP1,JSTEP2,KSTP1,TE)
      ENDDO
C
      RETURN
      END
```

SUBROUTINE LISOLV

This subroutine describes the Tri-Diagonal Matrix Algorithm (TDMA) solution method. This method is linked with each momentum equation.

```
$DEBUG
C
C***** SUBROUTINE LISOLV
C
      SUBROUTINE TDMA(ISTART,JSTART,KSTART,I1,I2,J1,J2,KKK,PHI)
C
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAM.F'

      DIMENSION PHI(NXYZ),A(NXYZ),B(NXYZ),C(NXYZ),D(NXYZ)
C
      DO J=JSTART,NJM1
      DO I=ISTART,NIM1
      IF(I.GE.I1.AND.I.LE.I2.AND.J.GE.J1.AND.J.LE.J2) THEN
      KSTM1=KKK-1
      ELSE
      KSTM1=KSTART-1
      ENDIF
      IJKSTM1=LK(KSTM1)+LI(I)+J
      A(IJKSTM1)=0.0
      ENDDO
      ENDDO
C
      DO 100 J=JSTART,NJM1
      DO 100 I=ISTART,NIM1
      IF(I.GE.I1.AND.I.LE.I2.AND.J.GE.J1.AND.J.LE.J2) THEN
      KSTM1=KKK-1
      K3=KKK
      ELSE
      KSTM1=KSTART-1
      K3=KSTART
```

```
            ENDIF
            IJKSTM1=LK(KSTM1)+LI(I)+J
            C(KSTM1)=PHI(IJKSTM1)
C---- COMMENCE B-T TRAVERSE
            DO 110 K=K3,NKM1
            IJK=LK(K)+LI(I)+J
C
C-----ASSEMBLE TDMA COEFFICIENTS
            A(K)=AT(IJK)
            B(K)=AB(IJK)
            C(K)=AE(IJK)*PHI(IJK+NJ)+AW(IJK)*PHI(IJK-NJ)+
            &           AN(IJK)*PHI(IJK+1)+AS(IJK)*PHI(IJK-1)+SU(IJK)
            D(K)=AP(IJK)
C
C----- CALCULATE COEFFICIENTS OF RECURRENCE FORMULA
C           WRITE(*,*) I,J,K,IJK,A(K),B(K),D(K)
            TERM=1.0/(D(K)-B(K)*A(K-1))
            A(K)=A(K)*TERM
            C(K)=(C(K)+B(K)*C(K-1))*TERM
  110 CONTINUE
C
C----- OBTAIN NEW PHI'S BY BACK SUBSTITUTION
            DO 120 KK=K3,NKM1
            K=NK+KSTM1-KK
            IJK=LK(K)+LI(I)+J
            PHI(IJK)=A(K)*PHI(IJK+NIJ)+C(K)
  120 CONTINUE
  100 CONTINUE
1000   CONTINUE
            RETURN
            END
```

SUBROUTINE OUTPUT

This subroutine prints out the specified variables, for the steady state and the unsteady state. The instantaneous variables saved at every time step and time averaged variables are saved after the steady state is reached.

```
$DEBUG
C
      SUBROUTINE ARRANG
C     USE MSIMSL
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAM.F'
C
      OPEN(5,FILE='GRID_XYZ.DAT')
C
      IF (ITIM.GT.1) GOTO 6
C
C----- ARRANGEMENT FOR STAGGERED GRID
      X(1)=(X(1)+X(2))/2.0
      X(NI)=(X(NI)+X(NIM1))/2.0
      Y(1)=(Y(1)+Y(2))/2.0
      Y(NJ)=(Y(NJ)+Y(NJM1))/2.0
      Z(1)=(Z(1)+Z(2))/2.0
      Z(NK)=(Z(NK)+Z(NKM1))/2.0
C
      DO 5 I=1,NI
      DO 5 J=1,NJ
      DO 5 K=1,NK
      XN(I)=(X(I)-XINLET)/H
      YN(J)=Y(J)/H
      ZN(K)=Z(K)/H
      WRITE(5,110) I,J,K,X(I),Y(J),Z(K),XN(I),YN(J),ZN(K)
5     CONTINUE
      CLOSE(5)
110   FORMAT(1X,3I5,6(1X,F12.7))
C
```

```
6       CONTINUE
C----- ARRANGEMENT FOR STAGGERED VELOCITIES
C       U VELOCITY
C       I=1
        DO 10 K=1,NK
        DO 10 J=1,NJ
        IJK1=LK(K)+LI(1)+J
        IJK2=LI(K)+LI(2)+J
        US(IJK1)=U(IJK2)
        USMM(IJK1)=UMM(IJK2)
10      CONTINUE
C
        DO 11 K=1,NK
        DO 11 J=1,NJ
        DO 11 I=2,NIM1
        IJK=LK(K)+LI(I)+J
        US(IJK)=(U(IJK)+U(IJK+NJ))/2.0
        USMM(IJK)=(UMM(IJK)+UMM(IJK+NJ))/2.0
11      CONTINUE
C
        DO 12 K=1,NK
        DO 12 J=1,NJ
        IJK=LK(K)+LI(NI)+J
        US(IJK)=U(IJK)
        USMM(IJK)=UMM(IJK)
12      CONTINUE
C
C       V VELOCITY
        DO 20 K=1,NK
        DO 20 I=1,NI
        IJK1=LK(K)+LI(I)+1
        IJK2=LK(K)+LI(I)+2
        VS(IJK1)=V(IJK2)
        VSMM(IJK1)=VMM(IJK2)
20      CONTINUE
C
```

```
      DO 21 K=1,NK
      DO 21 J=2,NJM1
      DO 21 I=1,NI
      IJK=LK(K)+LI(I)+J
      VS(IJK)=(V(IJK)+V(IJK+1))/2.0
      VSMM(IJK)=(VMM(IJK)+VMM(IJK+1))/2.0
21    CONTINUE
C
      DO 22 K=1,NK
      DO 22 I=1,NI
      IJK=LK(K)+LI(I)+NJ
      VS(IJK)=V(IJK)
      VSMM(IJK)=VMM(IJK)
22    CONTINUE
C
C     W VELOCITY
      DO 30 J=1,NJ
      DO 30 I=1,NI
      IJK1=LK(1)+LI(I)+J
      IJK2=LK(2)+LI(I)+J
      WS(IJK1)=W(IJK2)
      WSMM(IJK1)=WMM(IJK2)
30    CONTINUE
C
      DO 31 K=2,NKM1
      DO 31 J=1,NJ
      DO 31 I=1,NI
      IJK=LK(K)+LI(I)+J
      WS(IJK)=(W(IJK)+W(IJK+NIJ))/2.0
      WSMM(IJK)=(WMM(IJK)+WMM(IJK+NIJ))/2.0
31    CONTINUE
C
      DO 32 J=1,NJ
      DO 32 I=1,NI
      IJK=LK(NK)+LI(I)+J
      WS(IJK)=W(IJK)
```

```
            WSMM(IJK)=WMM(IJK)
32          CONTINUE
C
C***** STREAM FUNCTION
            MIJKJ=NJ/2
            DO J=1,NJ
            IJK=LK(1)+LI(1)+MIJKJ
            PPSI(IJK)=0.0
            ENDDO
C
            DO I=2,NI
            IF(I.GE.ISTP1.AND.I.LE.ISTEP2) THEN
            KK=KSTP1
            ELSE
            KK=2
            ENDIF
            IJK=LK(KK)+LI(I)+MIJKJ
            PPSI(IJK)=PPSI(IJK-NJ)-(WS(IJK)+WS(IJK-NJ))/2.0*(X(I)-X(I-1))
            ENDDO
C
            DO I=1,NI
            DO K=2,NK
            IJK=LK(K)+LI(I)+MIJKJ
            PPSI(IJK)=PPSI(IJK-NIJ)+(US(IJK)+US(IJK-NIJ))/2.0*(Z(K)-Z(K-1))
            ENDDO
            ENDDO
C
C***** VORTICITY
            DO I=2,NIM1
            IF(I.GE.ISTP1.AND.I.LE.ISTEP2) THEN
            KK=KSTP1
            ELSE
            KK=2
            ENDIF
            DO K=KK,NKM1
            IJK=LK(K)+LI(I)+MIJKJ
```

```
            DWDX=(0.5*(WS(IJK)+WS(IJK+NJ))-0.5*(WS(IJK)+WS(IJK-NJ)))/SEW(I)
            DUDZ=(0.5*(US(IJK)+US(IJK+NIJ))-0.5*(US(IJK)+US(IJK-NIJ)))/STB(K)
            VOR(IJK)=DWDX-DUDZ
            ENDDO
            ENDDO
            RETURN
            END
C
C-----      END OF THE ARRAGEMENTS
C
C*********************************************************************
C
C           PRINT FOR THE STEADY SOLUTIONS
C
C
C           THE VALUES OF THE REY_STRESS ARE NON-DIMENSIONALIZED BY UIN^2
C*********************************************************************
            SUBROUTINE PRINTST
C
C           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            INCLUDE 'PARAM.F'
C
            OPEN(9, FILE='MIDJ_P_FLUC.DAT')
            OPEN(10,FILE='U_FLUC.DAT')
            OPEN(11,FILE='V_FLUC.DAT')
            OPEN(12,FILE='W_FLUC.DAT')
            OPEN(13,FILE='VISCOS.DAT')
            OPEN(14,FILE='MIDJ_VELTEP_FLUC.DAT')
            OPEN(15,FILE='MIDK_VELTEP_FLUC.DAT')
            OPEN(17,FILE='MIDJ-STREAM_FC.DAT')
            OPEN(19,FILE='MIDJ_VISCOS.DAT')
            OPEN(20,FILE='TOTAL-VELTEP_FLUC.DAT')
            OPEN(21,FILE='GEN_INFLOWFL.DAT')
            OPEN(22,FILE='MIDJ_VORT.DAT')
            OPEN(30,FILE='MIDJ_VELTEP_MEAN.DAT')
            OPEN(31,FILE='MIDK_VELTEP_MEAN.DAT')
```

```
C
      MIJKJ=NJ/2
      UN=UIN
C
      DO J=1,NJ
      DO I=1,NI
      DO K=1,NK
      IJK=LK(K)+LI(I)+J
C
      IF(J.EQ.MIJKJ) THEN
      WRITE(9,121)I,J,K,XN(I),YN(J),ZN(K),P(IJK)
      WRITE(14,120)I,J,K,XN(I),YN(J),ZN(K),US(IJK),VS(IJK),
     &              WS(IJK),TE(IJK),P(IJK)
      WRITE(17,121)I,J,K,XN(I),YN(J),ZN(K),PPSI(IJK)
      WRITE(19,101)I,J,K,XN(I),YN(J),ZN(K),VIS(IJK)
      WRITE(22,100)I,J,K,XN(I),YN(J),ZN(K),VOR(IJK)
      WRITE(30,120)I,J,K,XN(I),YN(J),ZN(K),USMM(IJK),VSMM(IJK),
     &            WSMM(IJK),TEMM(IJK),PMM(IJK)
      ENDIF
C
      WRITE(10,100) I,J,K,XN(I),YN(J),ZN(K),US(IJK)
      WRITE(11,100) I,J,K,XN(I),YN(J),ZN(K),VS(IJK)
      WRITE(12,100) I,J,K,XN(I),YN(J),ZN(K),WS(IJK)
      WRITE(13,101) I,J,K,XN(I),YN(J),ZN(K),VIS(IJK)
      WRITE(20,120) I,J,K,XN(I),YN(J),ZN(K),US(IJK),VS(IJK),WS(IJK),
     &              TE(IJK),P(IJK)
C
      ENDDO
      ENDDO
      ENDDO
C
C***** XZ-PLANE
      DO I=1,NI
      DO J=1,NJ
      K=(KSTEP-1)/2
      IJK=LK(K)+LI(I)+J
```

```
      WRITE(15,122)I,J,XN(I),YN(J),ZN(K),US(IJK),VS(IJK),WS(IJK),TE(IJK)
     &                ,P(IJK)
      WRITE(31,122)I,J,XN(I),YN(J),ZN(K),USMM(IJK),VSMM(IJK),WSMM(IJK),
     &                TEMM(IJK),PMM(IJK)
      ENDDO
      ENDDO
C
C*****  GENERATE THE INLET VELOCITY
      DO K=2,NKM1
      DO J=2,NJM1
      IJKIN=LK(K)+LI(2)+J
      WRITE(21,122) J,K,YN(J),ZN(K),US(IJKIN),VS(IJKIN),WS(IJKIN),
     &                TE(IJKIN),P(IJKIN)
      ENDDO
      ENDDO
C
C***** PRINT OUT IN NEAR BUILDING ON THE MIDDLE AXIS
C
      OPEN(500,FILE='NEARXZ_FLUC.DAT')
      OPEN(501,FILE='NEARXZ_MEAN.DAT')

      DO I=ISTEP1-15,ISTEP2+10
      DO K=1,NKM1
      IJK=LK(K)+LI(I)+MIJKJ
      WRITE(500,122) I,K,XN(I),ZN(K),US(IJK),VS(IJK),WS(IJK),TE(IJK),
     &                P(IJK)
      WRITE(501,122) I,K,XN(I),ZN(K),USMM(IJK),VSMM(IJK),WSMM(IJK),
     &                TEMM(IJK),PMM(IJK)
      ENDDO
      ENDDO
C
100   FORMAT(1X,3I5,3(1X,F7.3),1X,F7.3)
101   FORMAT(1X,3I5,3(1X,F7.3),1X,E10.6)
120   FORMAT(1X,3I5,3(1X,F7.3),3(1X,F7.3),2(1X,F12.5))
121   FORMAT(1X,3I5,3(1X,F7.3),1X,F12.5)
122   FORMAT(1X,2I5,3(1X,F7.3),3(1X,F7.3),2(1X,F12.5))
```

```
C
      CLOSE(9)
      CLOSE(10)
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      CLOSE(14)
      CLOSE(15)
      CLOSE(17)
      CLOSE(19)
      CLOSE(20)
      CLOSE(21)
      CLOSE(22)
      CLOSE(30)
      CLOSE(31)
      CLOSE(500)
      CLOSE(501)
C
C
C***** PRINT OUT OF REYNOLDS STRESSES
      CALL REYSTRESS
C
      OPEN(60,FILE='UU_FLUC.DAT')
      OPEN(61,FILE='UV_FLUC.DAT')
      OPEN(62,FILE='UW_FLUC.DAT')
      OPEN(63,FILE='VV_FLUC.DAT')
      OPEN(64,FILE='VW_FLUC.DAT')
      OPEN(65,FILE='WW_FLUC.DAT')
      OPEN(66,FILE='MID_REYSTRESSF.DAT')

C
      DO J=1,NJ
      DO I=1,NI
      DO K=1,NK
      IJK=LK(K)+LI(I)+J
      WRITE(60,601) I,J,K,XN(I),YN(J),ZN(K),RUU(IJK)
```

```
            WRITE(61,601) I,J,K,XN(I),YN(J),ZN(K),RUV(IJK)
            WRITE(62,601) I,J,K,XN(I),YN(J),ZN(K),RUW(IJK)
            WRITE(63,601) I,J,K,XN(I),YN(J),ZN(K),RVV(IJK)
            WRITE(64,601) I,J,K,XN(I),YN(J),ZN(K),RVW(IJK)
            WRITE(65,601) I,J,K,XN(I),YN(J),ZN(K),RWW(IJK)
            ENDDO
            ENDDO
            ENDDO
601         FORMAT(1X,3I5,4(1X,F7.3))
C
            DO I=2,NIM1
            DO K=2,NKM1
            IJK=LK(K)+LI(I)+MIJKJ
            WRITE(66,602) I,K,XN(I),ZN(K),RUU(IJK),RUV(IJK),RUW(IJK),RVV(IJK),
            &                                          RVW(IJK),RWW(IJK)
            ENDDO
            ENDDO
602         FORMAT(1X,2I5,8(1X,F7.3))
            CLOSE(60)
            CLOSE(61)
            CLOSE(62)
            CLOSE(63)
            CLOSE(64)
            CLOSE(65)
            CLOSE(66)
C
C***** PRESSURE COEFFICIENTS ALONG WITH MID_XZ PLANES
C
            OPEN(300,FILE='CPXZ_W_MEAN.DAT')
            OPEN(301,FILE='CPXZ_W_FLUC.DAT')
            OPEN(310,FILE='CPXZ_T_MEAN.DAT')
            OPEN(311,FILE='CPXZ_T_FLUC.DAT')
            OPEN(320,FILE='CPXZ_E_MEAN.DAT')
            OPEN(321,FILE='CPXZ_E_FLUC.DAT')
C
C***** REFERENCE PRESSURE AND VELOCITY AT THE BUIDING HEIGHT
```

```
C       I=2, J=NJ/2, K=KSTEP
        IPREF=2
        JPREF=NJ/2
        KREFS=KSTEP

        IJKREFS=LK(KREFS)+LI(IPREF)+JPREF
        PREFS=PMM(IJKREFS)
        UREFS=UMM(IJKREFS)
        OPEN(302,FILE='REF_VALUES.DAT')
        WRITE(302,*) PREFS,UREFS,PREFS,UREFS
        CLOSE(302)
C       FRONT WALL (WEST WALL)
        I=ISTEP1
        J=NJ/2
        DO K=2,KSTEP
        IJK=LK(K)+LI(I)+J
        CPFS(K)=(PMM(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
        CPFFS(K)=(P(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
        WRITE(300,1001) I,J,K,XN(I),YN(J),ZN(K),PMM(IJK),CPFS(K)
        WRITE(301,1001) I,J,K,XN(I),YN(J),ZN(K),P(IJK),CPFFS(K)
        ENDDO
C       ROOF (TOP WALL)
        J=NJ/2
        K=KSTP1
        DO I=ISTP1,ISTEP2
        IJK=LK(K)+LI(I)+J
        CPRS(I)=(PMM(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
        CPFRS(I)=(P(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
        WRITE(310,1001) I,J,K,XN(I),YN(J),ZN(K),PMM(IJK),CPRS(I)
        WRITE(311,1001) I,J,K,XN(I),YN(J),ZN(K),P(IJK),CPFRS(I)
        ENDDO
C       BACKWARD WALL (EAST WALL)
        I=ISTP2
        J=NJ/2
        DO K=KSTEP,2,-1
        IJK=LK(K)+LI(I)+J
```

```
      CPBS(K)=(PMM(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
      CPFBS(K)=(P(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
      WRITE(320,1001) I,J,K,XN(I),YN(J),ZN(K),PMM(IJK),CPBS(K)
      WRITE(321,1001) I,J,K,XN(I),YN(J),ZN(K),P(IJK),CPFBS(K)
      ENDDO
1001  FORMAT(1X,3I5,3(1X,F7.3),2(1X,F7.3))

      CLOSE(300)
      CLOSE(310)
      CLOSE(320)
      CLOSE(301)
      CLOSE(311)
      CLOSE(321)
C*********************************************************************
C***** FLUCTUATION CP
C
      OPEN(400,FILE='CPXZ_W_RMS.DAT')
      OPEN(410,FILE='CPXZ_T_RMS.DAT')
      OPEN(420,FILE='CPXZ_E_RMS.DAT')
C     FRONT WALL (WEST WALL)
      I=ISTEP1
      J=NJ/2
      DO K=2,KSTEP
      IJK=LK(K)+LI(I)+J
      CPFS(K)=SQRT((P(IJK)-PMM(IJK))**2.)/(0.5*DENSIT*UREFS**2)
      WRITE(400,1002) I,J,K,XN(I),YN(J),ZN(K),(P(IJK)-PMM(IJK)),
     &                CPFS(K)
      ENDDO
C     ROOF (TOP WALL)
      J=NJ/2
      K=KSTP1
      DO I=ISTP1,ISTEP2
      IJK=LK(K)+LI(I)+J
      CPRS(I)=SQRT((P(IJK)-PMM(IJK))**2.)/(0.5*DENSIT*UREFS**2)
      WRITE(410,1002) I,J,K,XN(I),YN(J),ZN(K),(P(IJK)-PMM(IJK)),
     &                CPRS(I)
```

```
      ENDDO
C     BACKWARD WALL (EAST WALL)
      I=ISTP2
      J=NJ/2
      DO K=KSTEP,2,-1
      IJK=LK(K)+LI(I)+J
      CPBS(K)=SQRT((P(IJK)-PMM(IJK))**2.)/(0.5*DENSIT*UREFS**2)
      WRITE(420,1002) I,J,K,XN(I),YN(J),ZN(K),(P(IJK)-PMM(IJK)),
     &                CPBS(K)
      ENDDO
1002  FORMAT(1X,3I5,3(1X,F7.3),2(1X,F7.3))
      CLOSE(400)
      CLOSE(410)
      CLOSE(420)
C*************************************************************
C     CP ALONG WITH XY PLANE
C     FRONT WALL (WEST WALL)
      K=(KSTEP-1)/2
      I=ISTEP1
      DO J=JSTEP2,JSTP1,-1
      IJK=LK(K)+LI(I)+J
      OPEN(330,FILE='CPXY_W_MEAN.DAT')
      CPFYS(J)=(PMM(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
      WRITE(330,1003) I,J,K,XN(I),YN(J),ZN(K),PMM(IJK),CPFYS(J)
      ENDDO
C     SIDE WALL (SOUTH WALL)
      K=(KSTEP-1)/2
      J=JSTEP1
      DO I=ISTP1,ISTEP2
      IJK=LK(K)+LI(I)+J
      OPEN(340,FILE='CPXY_S_MEAN.DAT')
      CPSYS(I)=(PMM(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
      WRITE(340,1003) I,J,K,XN(I),YN(J),ZN(K),PMM(IJK),CPSYS(I)
      ENDDO
C     BACKWARD WALL (EAST WALL)
      K=(KSTEP-1)/2
```

```
      I=ISTP2
      DO J=JSTP1,JSTEP2
      IJK=LK(K)+LI(I)+J
      OPEN(350,FILE='CPXY_E_MEAN.DAT')
      CPBYS(J)=(PMM(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
      WRITE(350,1003) I,J,K,XN(I),YN(J),ZN(K),PMM(IJK),CPBYS(J)
      ENDDO
1003  FORMAT(1X,3I5,5(1X,F7.3))
      CLOSE(330)
      CLOSE(340)
      CLOSE(350)
C
C***** THE VALUES OF SUREFACES
      OPEN(5000,FILE='SURFMEAN_W.DAT')
      OPEN(5010,FILE='SURFMEAN_E.DAT')
      DO K=2,KSTEP
      DO J=JSTP1,JSTEP2
      I=ISTEP1
      IJK=LK(K)+LI(I)+J
      WRITE(5000,2000) J,K,YN(J),ZN(K),USMM(IJK),VSMM(IJK),WSMM(IJK),
     &                 TEMM(IJK),PMM(IJK)
      I=ISTP2
      IJK=LK(K)+LI(I)+J
      WRITE(5010,2000) J,K,YN(J),ZN(K),USMM(IJK),VSMM(IJK),WSMM(IJK),
     &                 TEMM(IJK),PMM(IJK)
      ENDDO
      ENDDO
C
      OPEN(5020,FILE='SURFMEAN_S.DAT')
      OPEN(5030,FILE='SURFMEAN_N.DAT')
      DO K=2,KSTEP
      DO I=ISTP1,ISTEP2
      J=JSTEP1
      IJK=LK(K)+LI(I)+J
      WRITE(5020,2000) I,K,XN(I),ZN(K),USMM(IJK),VSMM(IJK),WSMM(IJK),
     &                 TEMM(IJK),PMM(IJK)
```

```
      J=JSTP2
      IJK=LK(K)+LI(I)+J
      WRITE(5030,2000) I,K,XN(J),ZN(K),USMM(IJK),VSMM(IJK),WSMM(IJK),
     &                 TEMM(IJK),PMM(IJK)
      ENDDO
      ENDDO
C
      OPEN(5040,FILE='SURFMEAN_T.DAT')
      DO J=JSTP1,JSTEP2
      DO I=ISTP1,ISTEP2
      K=KSTP1
      IJK=LK(K)+LI(I)+J
      WRITE(5040,2000) I,J,XN(I),YN(J),USMM(IJK),VSMM(IJK),WSMM(IJK),
     &                 TEMM(IJK),PMM(IJK)
      ENDDO
      ENDDO
C
      OPEN(5050,FILE='SURFFLUC_W.DAT')
      OPEN(5060,FILE='SURFFLUC_E.DAT')
      DO K=2,KSTEP
      DO J=JSTP1,JSTEP2
      I=ISTEP1
      IJK=LK(K)+LI(I)+J
      WRITE(5050,2000)J,K,YN(J),ZN(K),US(IJK)-USMM(IJK),
     &                VS(IJK)-VSMM(IJK),
     &                WS(IJK)-WSMM(IJK),TE(IJK)-TEMM(IJK),P(IJK)-PMM(IJK)
      I=ISTP2
      IJK=LK(K)+LI(I)+J
      WRITE(5060,2000)J,K,YN(J),ZN(K),US(IJK)-USMM(IJK),
     &                VS(IJK)-VSMM(IJK),
     &                WS(IJK)-WSMM(IJK),TE(IJK)-TEMM(IJK),P(IJK)-PMM(IJK)
      ENDDO
      ENDDO
C
      OPEN(5070,FILE='SURFFLUC_S.DAT')
      OPEN(5080,FILE='SURFFLUC_N.DAT')
```

```
      DO K=2,KSTEP
      DO I=ISTP1,ISTEP2
      J=JSTEP1
      IJK=LK(K)+LI(I)+J
      WRITE(5070,2000)I,K,XN(I),ZN(K),US(IJK)-USMM(IJK),
     &          VS(IJK)-VSMM(IJK),
     &          WS(IJK)-WSMM(IJK),TE(IJK)-TEMM(IJK),P(IJK)-PMM(IJK)
      J=JSTP2
      IJK=LK(K)+LI(I)+J
      WRITE(5080,2000)I,K,XN(I),ZN(K),US(IJK)-USMM(IJK),
     &          VS(IJK)-VSMM(IJK),
     &          WS(IJK)-WSMM(IJK),TE(IJK)-TEMM(IJK),P(IJK)-PMM(IJK)
      ENDDO
      ENDDO
C
      OPEN(5090,FILE='SURFFLUC_T.DAT')
      DO J=JSTP1,JSTEP2
      DO I=ISTP1,ISTEP2
      K=KSTP1
      IJK=LK(K)+LI(I)+J
      WRITE(5090,2000)I,J,XN(I),YN(J),US(IJK)-USMM(IJK),
     &          VS(IJK)-VSMM(IJK),
     &          WS(IJK)-WSMM(IJK),TE(IJK)-TEMM(IJK),P(IJK)-PMM(IJK)
      ENDDO
      ENDDO
2000  FORMAT (1X,2I5,7(1X,F7.3))
C
C***** WPLUS
      OPEN(400,FILE='WPLUS.DAT')
      DO K=2,NKM1
      I=2
      J=NJ/2
      IJK=LK(K)+LI(I)+J
      IJK2=LK(2)+LI(2)+J
      ZPLUS(IJK)=DENSIT*(CMU**0.25)*SQRT(TE(IJK))*(Z(K)-ZW(2))/VISCOS
      AML=CAPPA*(Z(K)-ZW(2))*(1.0-EXP(-ZPLUS(IJK)/26.0))
```

```
      UPLUS(IJK)=USMM(IJK)/USMM(IJK2)
      WRITE(400,401) I,K,U(IJK2),TE(IJK2),AML,ZPLUS(IJK),UPLUS(IJK)
      ENDDO
401   FORMAT(1X,2(1X,I5),5(1X,F7.3))
      RETURN
      END
C*********************************************************************
C*********************************************************************
C
C..... PRINT OUT THE UNSTEADY DATA
C
C*********************************************************************
C
      SUBROUTINE PRINTUN
C
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAM.F'
C
      INTEGER(4) ITIM
      CHARACTER*8 PNTIME
      CHARACTER*26 FILOTIME,FILOTIME2,FILOTIME3,FILOTIME4,FILOTIME5,
     &                    FILOTIME6,FILOTIME7,FILOTIME8,FILOTIME9,
     &                    FILOTIME10,FILOTIME11,FILOTIME12,
     &                    FILOTIME13,FILOTIME14,FILOTIME15,
     &                    FILOTIME16,FILOTIME17,FILOTIME18,
     &                    FILOTIME19,FILOTIME20,FILOTIME21,FILOTIME22,
     &                    FILOTIME23,FILOTIME24,FILOTIME25,FILOTIME26
C
C***** ARRAY 1D TO 2D WITH TIME
C
      MIJKJ=NJ/2
      MIJKI=NI/2
C
      IF(ITIM.LE.1000) THEN
      NMOD=100
      GOTO 1110
```

```
      ELSEIF (ITIM.GT.1000.AND.ITIM.LE.20000) THEN
      NMOD=200
      GOTO 1110
      ELSEIF (ITIM.GT.1000) THEN
      NMOD=200
      GOTO 1110
      ENDIF
      GOTO 2000
1110  IF (MOD(ITIM,NMOD).EQ.0) THEN
C
      CALL REYSTRESS
      WRITE(PNTIME,77) ITIM

      FILOTIME   ='TIME_J_A_F'//PNTIME//'.DAT'
      FILOTIME2 ='TIME_J_PSI_F'//PNTIME//'.DAT'
      FILOTIME3 ='TIME_RUVW_F'//PNTIME//'.DAT'
      FILOTIME4 ='TIME_VORT_F'//PNTIME//'.DAT'
      FILOTIME9 ='TIME_J_A_IN'//PNTIME//'.DAT'
      FILOTIME10='TIME_J_A_M'//PNTIME//'.DAT'

C
      DO 70 I=1,NI
      DO 70 K=1,NK
      J=MIJKJ
      IJK=LK(K)+LI(I)+J
C
      OPEN(50,FILE=FILOTIME)  !ALL
      OPEN(52,FILE=FILOTIME2)          !STREAM FUNCTION
      OPEN(66,FILE=FILOTIME3)          !REYNOLDS STRESSES
      OPEN(77,FILE=FILOTIME4)          !VORTICITY
      OPEN(74,FILE=FILOTIME9)          !INSTANT
      OPEN(53,FILE=FILOTIME10) !MEAN
C
      WRITE(50,78) I,J,K,XN(I),YN(J),ZN(K),US(IJK),VS(IJK),WS(IJK),
     &                          TE(IJK),P(IJK)
      WRITE(52,79)I,J,K,XN(I),YN(J),ZN(K),PPSI(IJK)
```

```
      WRITE(66,602) I,K,XN(I),ZN(K),RUU(IJK),RUV(IJK),RUW(IJK),RVV(IJK),
     &                                          RVW(IJK),RWW(IJK)
      WRITE(77,79)I,J,K,XN(I),YN(J),ZN(K),VOR(IJK)
      WRITE(74,76) I,J,K,XN(I),YN(J),ZN(K),US(IJK)-USMM(IJK),
     &   VS(IJK)-VSMM(IJK),WS(IJK)-WSMM(IJK),TE(IJK)-TEMM(IJK)

      WRITE(53,78)I,J,K,XN(I),YN(J),ZN(K),USMM(IJK),VSMM(IJK),WSMM(IJK),
     &                TEMM(IJK),PMM(IJK)
C
70    CONTINUE
602   FORMAT(1X,2I5,2(1X,F7.3),6(1X,F9.3))
C
      FILOTIME4='NEARXZ_FL'//PNTIME//'.DAT'
      FILOTIME7='NEARXZ_IN'//PNTIME//'.DAT'
      FILOTIME11='NEARXZ_M'//PNTIME//'.DAT'
C
      DO I=ISTEP1-15,ISTEP2+10
      DO K=1,NKM1
      J=NJ/2
      IJK=LK(K)+LI(I)+J
C
      OPEN(80,FILE=FILOTIME4)
      OPEN(81,FILE=FILOTIME7)
      OPEN(82,FILE=FILOTIME11)
C
      WRITE(80,78) I,J,K,XN(I),YN(J),ZN(K),US(IJK),VS(IJK),WS(IJK),
     &                          TE(IJK),P(IJK)
      WRITE(82,78) I,J,K,XN(I),YN(J),ZN(K),USMM(IJK),VSMM(IJK),WSMM(IJK)
     &           ,TEMM(IJK),PMM(IJK)
      WRITE(81,76) I,J,K,XN(I),YN(J),ZN(K),US(IJK)-USMM(IJK),
     &   VS(IJK)-VSMM(IJK),WS(IJK)-WSMM(IJK),TE(IJK)-TEMM(IJK)
      ENDDO
      ENDDO
C
C***** ALONG WITH XY PLANE
      FILOTIME6='NEARXY_IN'//PNTIME//'.DAT'
```

```
      FILOTIME22='NEARDXY_M'//PNTIME//'.DAT'
      FILOTIME26='NEARDXY_FL'//PNTIME//'.DAT'
C
      DO 71 I=ISTEP1-10,ISTEP2+10
      DO 71 J=2,NJM1
      K=(KSTEP-1)/2
      IJK=LK(K)+LI(I)+J
C
      OPEN(72,FILE=FILOTIME6)
      OPEN(99,FILE=FILOTIME22)
      OPEN(51,FILE=FILOTIME26)

      WRITE(51,78) I,J,K,XN(I),YN(J),ZN(K),US(IJK),VS(IJK),WS(IJK),
     &                          TE(IJK),P(IJK)
      WRITE(99,78) I,J,K,XN(I),YN(J),ZN(K),UMM(IJK),VMM(IJK),WMM(IJK),
     &                          TEMM(IJK),PMM(IJK)
      WRITE(72,76) I,J,K,XN(I),YN(J),ZN(K),US(IJK)-USMM(IJK),
     &  VS(IJK)-VSMM(IJK),WS(IJK)-WSMM(IJK),TE(IJK)-TEMM(IJK)
71    CONTINUE
C
      FILOTIME5='XYFL'//PNTIME//'.DAT'
      FILOTIME12='XYM'//PNTIME//'.DAT'
      FILOTIME8='XYINST'//PNTIME//'.DAT'
C
      DO I=1,NI
      DO J=1,NJ
C****
C****
      K=(KSTEP-1)/2
C****
C****
      IJK=LK(K)+LI(I)+J
C
      OPEN(52,FILE=FILOTIME5)
      OPEN(91,FILE=FILOTIME12)
      OPEN(73,FILE=FILOTIME8)
```

```
C
      WRITE(52,78) I,J,K,XN(I),YN(J),ZN(K),US(IJK),VS(IJK),WS(IJK),
     &             TE(IJK),P(IJK)
      WRITE(91,78) I,J,K,XN(I),YN(J),ZN(K),USMM(IJK),VSMM(IJK),WSMM(IJK)
     &             ,TEMM(IJK),PMM(IJK)
      WRITE(73,76) I,J,K,XN(I),YN(J),ZN(K),US(IJK)-USMM(IJK),
     &  VS(IJK)-VSMM(IJK),WS(IJK)-WSMM(IJK),TE(IJK)-TEMM(IJK)
C
      ENDDO
      ENDDO
C
C***** YZ PLANE
      FILOTIME16='MIDIYZ_FL'//PNTIME//'.DAT'
      FILOTIME17='MIDIYZ_M'//PNTIME//'.DAT'
      FILOTIME18='MIDIYZ_IN'//PNTIME//'.DAT'
C
      DO J=1,NJ
      DO K=1,NK
      I=ISTEP1+IROOF/2
      IJK=LK(K)+LI(I)+J
C
      OPEN(16,FILE=FILOTIME16)
      OPEN(17,FILE=FILOTIME17)
      OPEN(18,FILE=FILOTIME18)
C
      WRITE(16,78) I,J,K,XN(I),YN(J),ZN(K),US(IJK),VS(IJK),WS(IJK),
     &             TE(IJK),P(IJK)
      WRITE(17,78) I,J,K,XN(I),YN(J),ZN(K),USMM(IJK),VSMM(IJK),WSMM(IJK)
     &             ,TEMM(IJK),PMM(IJK)
      WRITE(18,76) I,J,K,XN(I),YN(J),ZN(K),US(IJK)-USMM(IJK),
     &  VS(IJK)-VSMM(IJK),WS(IJK)-WSMM(IJK),TE(IJK)-TEMM(IJK)
C
      ENDDO
      ENDDO

76    FORMAT(1X,3I5,7(1X,F12.5))
```

```
77      FORMAT(I8)
78      FORMAT(1X,3I5,7(1X,F7.3),1X,F12.5)
79      FORMAT(1X,3I5,3(1X,F7.3),1X,F12.5)
C
        CLOSE(16)
        CLOSE(17)
        CLOSE(18)
        CLOSE(50)
        CLOSE(51)
        CLOSE(52)
        CLOSE(53)
        CLOSE(66)
        CLOSE(72)
        CLOSE(73)
        CLOSE(74)
        CLOSE(76)
        CLOSE(77)
        CLOSE(80)
        CLOSE(81)
        CLOSE(51)
        CLOSE(82)
        CLOSE(99)
        CLOSE(91)
C
        IPREF=2
        JPREF=NJ/2
        KREFS=KSTEP

        FILOTIME13='CPXZ_W_M'//PNTIME//'.DAT'
        FILOTIME14='CPXZ_T_M'//PNTIME//'.DAT'
        FILOTIME15='CPXZ_E_M'//PNTIME//'.DAT'
        FILOTIME23='CPXZ_W_F'//PNTIME//'.DAT'
        FILOTIME24='CPXZ_T_F'//PNTIME//'.DAT'
        FILOTIME25='CPXZ_E_F'//PNTIME//'.DAT'
C       I=2, J=NJ/2, K=KSTEP
        IJKREFS=LK(KREFS)+LI(IPREF)+JPREF
```

```
      PREFS=PMM(IJKREFS)
      UREFS=USMM(IJKREFS)
C     FRONT WALL (WEST WALL)
      I=ISTEP1
      J=NJ/2
      DO K=2,KSTEP
      IJK=LK(K)+LI(I)+J
      OPEN(300,FILE=FILOTIME13)
      OPEN(301,FILE=FILOTIME23)
      CPFS(K)=(PMM(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
      WRITE(300,1001) I,J,K,XN(I),YN(J),ZN(K),PMM(IJK),CPFS(K)
      CPFFS(K)=(P(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
      WRITE(301,1001) I,J,K,XN(I),YN(J),ZN(K),P(IJK),CPFFS(K)
      ENDDO
C     ROOF (TOP WALL)
      J=NJ/2
      K=KSTP1
      DO I=ISTP1,ISTEP2
      OPEN(310,FILE=FILOTIME14)
      OPEN(311,FILE=FILOTIME24)
      IJK=LK(K)+LI(I)+J
      CPRS(I)=(PMM(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
      WRITE(310,1001) I,J,K,XN(I),YN(J),ZN(K),PMM(IJK),CPRS(I)
      CPFRS(I)=(P(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
      WRITE(311,1001) I,J,K,XN(I),YN(J),ZN(K),P(IJK),CPFRS(I)
      ENDDO
C     BACKWARD WALL (EAST WALL)
      I=ISTP2
      J=NJ/2
      DO K=KSTEP,2,-1
      IJK=LK(K)+LI(I)+J
      OPEN(320,FILE=FILOTIME15)
      OPEN(321,FILE=FILOTIME25)
      CPBS(K)=(PMM(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
      WRITE(320,1001) I,J,K,XN(I),YN(J),ZN(K),PMM(IJK),CPBS(K)
      CPFBS(K)=(P(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
```

```
      WRITE(321,1001) I,J,K,XN(I),YN(J),ZN(K),P(IJK),CPFBS(K)
      ENDDO
1001  FORMAT(1X,3I5,5(1X,F12.5))
      CLOSE(300)
      CLOSE(310)
      CLOSE(320)
C
C     CP ALONG WITH XY PLANE
      FILOTIME19='CPXY_W_M'//PNTIME//'.DAT'
      FILOTIME20='CPXY_S_M'//PNTIME//'.DAT'
      FILOTIME21='CPXY_E_M'//PNTIME//'.DAT'
C     FRONT WALL      (WEST WALL)
      K=(KSTEP-1)/2
      I=ISTEP1
      DO J=JSTEP2,JSTP1,-1
      IJK=LK(K)+LI(I)+J
      OPEN(330,FILE=FILOTIME19)
      CPFYS(J)=(PMM(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
      WRITE(330,1002) I,J,K,XN(I),YN(J),ZN(K),PMM(IJK),CPFYS(J)
      ENDDO
C     SIDE WALL   (SIDE WALL)
      K=(KSTEP-1)/2
      J=JSTEP1
      DO I=ISTP1,ISTEP2
      OPEN(340,FILE=FILOTIME20)
      IJK=LK(K)+LI(I)+J
      CPSYS(I)=(PMM(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
      WRITE(340,1002) I,J,K,XN(I),YN(J),ZN(K),PMM(IJK),CPSYS(I)
      ENDDO
C     BACKWARD WALL   (EAST WALL)
      K=(KSTEP-1)/2
      I=ISTP2
      DO J=JSTP1,JSTEP2
      IJK=LK(K)+LI(I)+J
      OPEN(350,FILE=FILOTIME21)
      CPBYS(J)=(PMM(IJK)-PREFS)/(0.5*DENSIT*UREFS**2)
```

```
      WRITE(350,1002) I,J,K,XN(I),YN(J),ZN(K),PMM(IJK),CPBYS(J)
      ENDDO
1002  FORMAT(1X,3I5,5(1X,F7.3))
      CLOSE(330)
      CLOSE(340)
      CLOSE(350)
      ENDIF
C
      OPEN(22,FILE='TIMEI_UVWFIL.DAT')
      OPEN(23,FILE='TIMEI_UVWFLUC.DAT')
      OPEN(24,FILE='TIMEI_UVWMEAN.DAT')
      OPEN(220,FILE='TIMEO_UVWFIL.DAT')
      OPEN(230,FILE='TIMEO_UVWFLUC.DAT')
      OPEN(240,FILE='TIMEO_UVWMEAN.DAT')

      IJKI=LK(KSTEP)+LI(3)+NJ/2
      WRITE(22,80) TIM,US(IJKI),VS(IJKI),WS(IJKI),P(IJKI),TE(IJKI)   !FILTERED U
      WRITE(24,80) TIM,USMM(IJKI),VSMM(IJKI),WSMM(IJKI),PMM(IJKI),
     &   TEMM(IJKI)
      WRITE(23,80) TIM,US(IJKI)-USMM(IJKI),
     &   VS(IJKI)-VSMM(IJKI),WS(IJKI)-WSMM(IJKI),P(IJKI)-PMM(IJKI),
     &   TE(IJKI)-TEMM(IJKI)

      IJKO=LK(KSTEP)+LI(NJM1)+NJ/2
      WRITE(220,80) TIM,US(IJKO),VS(IJKO),WS(IJKO),P(IJKO),TE(IJKO)        !FILTERED U
      WRITE(240,80) TIM,USMM(IJKO),VSMM(IJKO),WSMM(IJKO),PMM(IJKO),
     &   TEMM(IJKO)
      WRITE(230,80) TIM,US(IJKO)-USMM(IJKO),
     &   VS(IJKO)-VSMM(IJKO),WS(IJKO)-WSMM(IJKO),P(IJKO)-PMM(IJKO),
     &   TE(IJKO)-TEMM(IJKO)
80    FORMAT(1X,F12.7,4(1X,F10.3),1X,F12.5)
C
C
C**********************************************************************
2000  RETURN
      END
```