5-2012

Evaluation of Turbulence Variable Distributions for Incompressible Fully Rough Pipe Flows

Emilie B. Fowler
Utah State University

EVALUATION OF TURBULENCE VARIABLE DISTRIBUTIONS FOR FULLY ROUGH PIPE FLOW

by

Emilie B. Fowler

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Mechanical Engineering

Approved:

Dr. Warren F. Phillips Major Professor	Dr. Robert E. Spall Major Professor
Dr. Barton Smith Committee Member	Dr. Wenbin Yu Committee Member
Dr. Jan J. Sojka Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2012

ABSTRACT

Evaluation of Turbulence Variable Distributions

for Incompressible Fully Rough Pipe Flows

by

Emilie B. Fowler, Doctor of Philosophy

Utah State University, 2012

Major Professors: Dr. Warren F. Phillips and Dr. Robert E. Spall
Department: Mechanical and Aerospace Engineering

The specific turbulent kinetic energy, root-mean-square fluctuating vorticity, and mean-vortex-wavelength distributions are presented for fully rough pipe flow. The distributions of these turbulence variables are obtained from a proposed turbulence model. Many of the turbulence models commonly used for computational fluid dynamics are based on an analogy between molecular and turbulent transport. However, traditional k-ε and k-ω models fail to exhibit proper dependence on the molecular viscosity. Based on a rigorous application of the Boussinesq's hypothesis, Phillips proposed a vorticity-based transport equation for the turbulent kinetic energy. The foundation for this vorticity-based transport equation is presented. In future development of this model, a transport equation for the fluctuating vorticity is needed. In order to assess the model and evaluate closure coefficients, the resulting turbulent vorticity distribution must be compared to reference distributions. This dissertation presents reference distributions for the mean fluctuating vorticity and mean turbulent wavelength obtained for fully rough pipe flow. These distributions are obtained from a turbulence model, which involves the proposed transport equation for the turbulent kinetic energy and an empirical relation for the mean vortex wavelength. The empirical relation for the mean vortex wavelength requires numerous closure coefficients. These closure coefficients are determined through gradient-based optimization techniques. The current model gives excellent agreement with well established relations obtained for both the friction factor and velocity distribution.

(278 pages)

ACKNOWLEDGMENTS

First I would like to sincerely thank my major professor, Dr. Warren F. Phillips, for his endless hours and dedication to this research, which did not go unnnoticed. His excellent guidance and support during every step of the research broaden my knowledge in so many different fields. I am also grateful for all of those, such as Dr. Robert E. Spall, who have given their time and effort to teach and inspire me.

I wish to thank some of my friends and family for their continuous support. In particular, I am grateful to Michal Hradisky and Milada Majerova for all the good time we spent together. I wish to thank Doug Hunsaker for all the interesting conversations we had and for providing a stimulating and fun environment. I am particularly grateful to my parents whose examples and personal sacrifices have allowed me to achieve so much. I would like to thank my Lake George family for all the fun hours we had talking. It was a really relaxing time for me.

Lastly and most importantly, I wish to thank my husband, David Fowler, for all his support, love and patience throughout this dissertation. He always followed me and supported me for all the decisions I had to take. Finally I would like to thank my precious daughter, Jane Fowler, for all of the smiles she gave me while I was explaining to her the concept of turbulence. That she might find as much joy in learning as I do; to her, I dedicate this dissertation.

Emilie B. Fowler

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

BFGS	Broyden-Fletcher-Goldfarb-Shanno
CFD	Computational Fluid Dynamics
DNS	Direct Numerical Simulation
RANS	Reynolds-averaged Navier-Stokes
RMS	Root-Mean-Square

NOMENCLATURE

A	=	arbitrary coefficient, used in Eq. (C.16)
$A_{0\text{-}3}$	=	coefficients defined in Eq. (2.56)
a	=	arbitrary constant used in Eq. (2.60)
$A_{\lambda 0}$	=	mean-vortex-wavelength coefficient, used in Eq. (2.116), defined inEq. (3.18)
$A_{\lambda 10}$	=	mean-vortex-wavelength coefficient, used in Eq. (3.31), defined inEq. (3.19)
$A_{\lambda 11}$	=	mean-vortex-wavelength coefficient, used in Eq. (3.31), defined in Eq. (3.20)
$A_{\lambda 12}$	=	mean-vortex-wavelength coefficient, used in Eq. (3.31), defined in Eq. (3.20)
$A_{\lambda 13}$	=	mean-vortex-wavelength coefficient, used in Eq. (3.31), defined in Eq. (3.22)
$A_{\lambda 1}$	=	mean-vortex-wavelength coefficient, used in Eq. (2.116), defined in Eq. (3.31)
B	=	arbitrary coefficient, used in Eq. (C.16)
$B_{\lambda 00}$	=	mean-vortex-wavelength coefficient, used in Eq. (3.32), defined in Eq. (3.23)
$B_{\lambda 01}$	=	mean-vortex-wavelength coefficient, used in Eq. (3.32), defined in Eq. (3.24)
$B_{\lambda 02}$	=	mean-vortex-wavelength coefficient, used in Eq. (3.32), defined in Eq. (3.25)
$B_{\lambda 03}$	=	mean-vortex-wavelength coefficient, used in Eq. (3.32), defined in Eq. (3.26)
$B_{\lambda 0}$	=	mean-vortex-wavelength coefficient, used in Eq. (2.116), defined in Eq. (3.32)
$B_{\lambda 10}$	=	mean-vortex-wavelength coefficient, used in Eq. (3.32), defined in Eq. (3.27)
$B_{\lambda 11}$	=	mean-vortex-wavelength coefficient, used in Eq. (3.33), defined in Eq. (3.28)
$B_{\lambda 1}$	=	mean-vortex-wavelength coefficient, used in Eq. (3.33) , defined in Eq. (3.33)
$B_{\lambda 20}$	=	mean-vortex-wavelength coefficient, used in Eq. (3.34), defined in Eq. (3.29)
$B_{\lambda 21}$	=	mean-vortex-wavelength coefficient, used in Eq. (3.34), defined in Eq. (3.30)
$B_{\lambda 2}$	=	mean-vortex-wavelength coefficient, used in Eq. (2.116) , defined in Eq. (3.34)
$C_{\epsilon 1\text{-}2}$	=	turbulence model closure coefficients, used in Eq. (1.89)
C_0	=	arbitrary coefficient, used in Eq. (C.16)
$C_{0\text{–}7}$	=	closure coefficients for the δ function, used in Eq. (2.89)
$C_{0\text{–}4}$	=	arbitrary coefficients, used in Eqs. (3.21), (3.25) and (3.29)
C_f	=	Fanning friction factor, defined in Eq. (2.7)

C_λ = turbulence model closure coefficient, first used in Eq. (1.138)

C_μ = turbulence model closure coefficient, used in Eq. (1.88)

$C_{\omega 1-\omega 2}$ = turbulence model closure coefficients, used in Eq. (4.6)

D_{1-3} = change of variables used to define the δ function, used in Eq. (2.94)

e = specific energy, defined in Eq. (1.9)

e = arbitrary coefficient, used in Eq. (C.16)

$\mathbf{f}$ = force vector per unit volume

$\mathbf{f}_g$ = gravity force per unit volume defined in Eq. (1.5)

f_{1-2} = turbulence model damping function, used in Eqs. (4.6) and (4.12)

f_k = turbulence model damping function, used in Eqs. (4.6) and (4.12)

f_μ = turbulence model damping function, used in Eqs. (4.6) and (4.12)

g = acceleration of gravity

g_o = acceleration of gravity at standard sea level

H = geometric altitude

$\mathbf{i}$ = unit vector

$\vec{\vec{\mathbf{J}}}$ = Jacobian tensor of a vector field defined in Eq. (1.67)

k = turbulent kinetic energy per unit mass defined in Eq. (1.39)

k^+ = wall-scaled dimensionless turbulent kinetic energy, $k^+ \equiv k/u_\tau^2$

k^+_{wall} = value of k^+ at the wall

k_r = dimensionless relative roughness, $k_r \equiv k_s/(2R)$

k_s = equivalent sand-grain roughness height

$\hat{k}_s$ = pipe-scaled dimensionless equivalent sand-grain roughness height, $\hat{k}_s \equiv k_s / R$, called the roughness ratio

k_s^+ = wall-scaled dimensionless equivalent sand-grain roughness height, $k_s^+ \equiv k_s u_\tau/\nu$, commonly called the roughness Reynolds number

ℓ = mixing length

ℓ^+ = wall-scaled mixing length, $\ell^+ \equiv \ell u_\tau/\nu$

m	=	mass
Ni	=	Nikuradse number defined in Eq. (2.22)
p	=	thermodynamic pressure
p^+	=	wall-scaled dimensionless pressure gradient, $p^+ \equiv (d\hat{\overline{p}}/dx)(\nu/\rho u_\tau^3)$
$\hat{\overline{p}}$	=	pseudo mean pressure defined in Eq. (1.41)
$\hat{p}$	=	pseudo hydrostatic pressure defined in Eq. (1.7)
$\overline{\hat{p}}$	=	mean pseudo hydrostatic pressure
$\widetilde{\overline{p}}$	=	term defined in Eq. (1.53)
$\hat{Q}$	=	pipe-scale volume flow rate, defined in Eq. (1.165)
$\mathbf{q}$	=	heat flux vector used in Eq. (1.16)
R	=	pipe radius
$\mathrm{R_e}$	=	Reynolds number, defined in Eq. (2.10)
$\hat{\mathrm{R}}_s$	=	nondimensional parameter defined in Eq. (2.13)
R_τ	=	shear Reynolds number, $R_\tau \equiv Ru_\tau/\nu$
r	=	radial distance defined in cylindrical coordinates
$\hat{r}$	=	pipe-scaled radial coordinate, $\hat{r} \equiv r/R$
$\overline{\overline{\mathbf{S}}}$	=	strain rate tensor defined in Eq. (1.4)
t	=	time
trace()	=	function trace, sum of the diagonal elements of a tensor
U'''_e	=	volumetric heating per unit volume used in Eq. (1.10)
u_e	=	specific internal energy used in Eq. (1.9)
u	=	x-component of velocity
u_τ	=	friction velocity, $u_\tau \equiv \sqrt{\tau_w/\rho}$
u^+	=	wall-scaled dimensionless velocity, $u^+ \equiv \overline{V}_z/u_\tau$
u_m^+	=	nondimensional bulk velocity
$\mathbf{V}$	=	velocity vector
V	=	velocity magnitude

V_c = periodic wave velocity scale, used in Eq. (1.86)

V_m = bulk velocity

x = coordinate measured along the x axis

y = coordinate measured along the y axis

$\hat{y}$ = nondimensional coordinate measured along the y axis, $\hat{y} = y/R$

y^+ = wall-scaled dimensionless distance, $y^+ \equiv yu_\tau/\nu$

Z = geopotential altitude

z = coordinate measured along the z axis

$\overline{\overline{\boldsymbol{\gamma}}}$ = tensor defined in Eq. (1.54)

γ_k = trace of $\overline{\overline{\boldsymbol{\gamma}}}$, defined in Eq. (1.62)

γ = Nikuradse constant

δ = deviation function, defined in Eq. (2.89)

$\overline{\overline{\boldsymbol{\delta}}}$ = Kronecker delta

∇ = gradient operator

$\overline{\overline{\boldsymbol{\varepsilon}}}$ = tensor defined in Eq. (1.56)

ε_k = trace of $\overline{\overline{\boldsymbol{\varepsilon}}}$, defined in Eq. (1.63)

ε = approximate turbulent-kinetic-energy dissipation defined in Eq. (1.81)

$\widetilde{\varepsilon}$ = exact turbulent-kinetic-energy dissipation defined in Eq. (1.131)

ζ = turbulence enstrophy

θ = azimuth angle defined in cylindrical coordinates

κ = Von Kármán constant

λ = mean vortex wavelength defined in Eq. (1.138)

λ^+ = wall-scaled mean vortex wavelength, $\lambda^+ \equiv \lambda u_\tau/\nu$

$\hat{\lambda}$ = pipe-scaled nondimensional mean vortex wavelength, $\hat{\lambda} \equiv \lambda / R$

λ_ϵ = mean dissipation wavelength defined in Eq. (4.16)

$\hat{\lambda}_\epsilon$ = pipe-scaled nondimensional mean dissipation wavelength, $\hat{\lambda}_\varepsilon \equiv \lambda_\varepsilon / R$

μ = dynamic viscosity

μ_t	=	dynamic eddy viscosity
ν	=	kinematic viscosity, $\nu \equiv \mu/\rho$
$\hat{\nu}$	=	pipe-scaled dimensionless kinematic viscosity, $\hat{\nu} \equiv \nu/u_\tau R$
ν_t	=	turbulent eddy viscosity, $\nu_t \equiv \mu_t/\rho$
$\hat{\nu}_t$	=	pipe-scaled dimensionless turbulent eddy viscosity, $\hat{\nu}_t \equiv \nu_t/u_\tau R$
$\hat{\nu}_{tc}$	=	value of the pipe-scaled dimensionless turbulent eddy viscosity at the centerline
ν^+	=	ratio of the turbulent eddy viscosity over the molecular viscosity, $\nu^+ \equiv \nu_t/\nu$
$\bar{\bar{\boldsymbol{\pi}}}$	=	tensor defined in Eq. (1.57)
π_k	=	trace of $\bar{\bar{\boldsymbol{\pi}}}$, defined in Eq. (1.65)
ρ	=	fluid density
$\bar{\bar{\boldsymbol{\sigma}}}$	=	fluid stress tensor defined in Eq. (1.34)
σ_k	=	turbulence model closure coefficient, first used in Eq. (1.83)
σ_ε	=	turbulence model closure coefficient, first used in Eq. (1.89)
σ_ω	=	turbulence model closure coefficient, used in Eq. (1.93)
$\bar{\bar{\boldsymbol{\tau}}}$	=	Reynolds stress tensor defined in Eq. (1.35)
τ_{xy}	=	Reynolds shear stress, $\tau_{xy} \equiv -\rho\overline{\tilde{V}_x\tilde{V}_y}$
τ_w	=	wall shear stress, defined in Eq. (2.2)
$\bar{\bar{\boldsymbol{\chi}}}$	=	tensor defined in Eq. (1.55)
χ_k	=	trace of $\bar{\bar{\boldsymbol{\chi}}}$, defined in Eq. (1.64)
χ_k	=	change of variable defined in Eq. (4.7)
$\boldsymbol{\Omega}$	=	rotation tensor, defined in Eq. (A.18)
ω	=	turbulence dissipation frequency
$\tilde{\omega}$	=	turbulent fluctuating vorticity defined in Eq. (1.132)
$\hat{\omega}$	=	dimensionless turbulent fluctuating vorticity, $\hat{\omega} \equiv R\omega/u_\tau$
ω_c	=	periodic wave angular velocity, used in Eq. (1.86)

Superscripts

T	=	transpose

Subscripts

i	=	index variable
j	=	index variable
k	=	associated with the turbulence variable, k
r	=	radial direction
rms	=	root-mean-square
t	=	associated with a turbulent variable
x	=	associated with the x-component or axis
y	=	associated with the y-component or axis
z	=	associated with the z-component or axis
ε	=	associated with the turbulence variable, ε
ω	=	associated with the turbulence variable, ω
θ	=	azimuth direction

Other notation

—	=	(overbar) ensemble-averaged value
~	=	(tilde) fluctuating value
′	=	(prime) denotes differentiation
.	=	(dot) denotes differentiation with time
bold	=	denotes an array or vector

CHAPTER 1

PHILLIPS TURBULENT-KINETIC-ENERGY TRANSPORT EQUATION

I. Introduction

Turbulent fluid motion is one of the most difficult yet fundamental problems encountered in applied mathematics. Turbulent flows are time-dependant, three-dimensional, diffusive and dissipative phenomena. One of the most challenging aspects in describing and predicting turbulence is that the velocity fluctuates over a large range of coupled spatial and temporal scales. The turbulent quantities change so rapidly in space and time that experimental data are not easily obtained. Due to its practical importance, many scientists have attempted to describe the motion of a turbulent fluid through numerical models. Although the basic equations describing turbulence are well known general principles, their solutions are incredibly complex and require tremendous computing power. No solution that unconditionally predicts the motion of a turbulent fluid has yet been found.

The basic equations describing the motion of a turbulent fluid are well known and have been studied for many generations. Euler (1757) [1] was one the earliest scientist to describe the motion of inviscid fluids. A major step was the development of the Navier-Stokes equations named after the work of Navier (1823) [2] and Stokes (1845) [3]. This set of nonlinear coupled partial differential equations describes the rate of change of momentum in a viscous fluid. Reynolds (1895) [4] rewrote the Navier-Stokes equations in a form where the main variables are expressed as a sum of the mean and fluctuating components. This approach, known as the Reynolds-Averaged Navier-Stokes equations, set the traditional mathematical framework used commonly today. Prandtl (1925) [5] introduced the additional concepts of the mixing length and boundary layer derived after the work of Boussinesq (1877) [6] who postulated that the momentum transfer caused by turbulent eddies can be modeled as a linear function of the mean strain rate. Taylor (1938) [7] was one of the earliest scientists to introduce the Eulerian viewpoint into turbulence modeling. The developments of these scientists continue to play significant roles in current turbulence research.

Turbulent flows are characterized by irregular fluctuations that can be approached through the use of statistical models. Turbulence consists of a continuous spectrum of scales ranging from the largest to the

smallest. Kolmogorov (1942) [8] developed a theory based on energy spectrum analysis, which describes how energy is transferred from larger to smaller eddies. Following earlier work of Kolmogorov on statistical turbulence, Kraichnan (1958) [9] developed a field-theory approach to fluid flow.

Many turbulence models are based on the Reynolds-averaged Navier-Stokes equations. The Reynolds decomposition yields a set of equations governing the average flow field instead of the exact turbulent flow field. Although this decomposition has the benefit of reducing the multi-scale problem to a scale of 1 or 2, more unknowns have been added (such as the turbulent fluxes and turbulent stresses). Turbulence modeling consists of representing the scales of the flow that are not resolved.

Most turbulence models are developed by analogy between molecular and turbulent transport. The analogy is based on the Boussinesq (1877) [6] approximation, which states that the Reynolds stress can be expressed as a linear function of the mean strain rate, in the same way that the molecular stress is expressed as a linear function of the total strain rate. The proportionality coefficient between the Reynolds stress and the mean strain rate is now commonly called the eddy viscosity. Since the turbulent eddy viscosity is a flow property and not a fluid property, Bousinesq-based turbulence modeling consists of devising a model for the turbulent eddy viscosity.

The Boussinesq approximation [6] published in 1877 was not explicitly expressed in terms of a tensor equation and was redeveloped in 1895 by Reynolds [4] who undertook a more rigorous approach and made use of a tensor notation from which the name of the "Reynolds stress tensor" was derived. In his 1877 publication, Boussinesq performed a temporal average of the Navier-Stokes equation. Many of the approximations he used are not explicitly stated. He introduced the elements of a stress tensor without explicitly identifying it. This stress tensor was identified by Reynolds in 1895. From an approach similar to that used in the kinetic theory of gases, Boussinesq proposed an expression for the turbulent stresses. This expression is in fact a tensorial closure equation that obeys the Navier-Stokes equations and includes a parameter that today is commonly referred to as the turbulent eddy viscosity. Boussinesq did not give a name to this parameter but remarked that this parameter is a function of the degree of turbulence in the fluid. Boussinesq proposed an expression for this parameter based on a characteristic length and velocity. This formed the basis of the mixing length formulation that was later developed in detail by Prandtl. Boussinesq's hypothesis was originally mentioned as a tensorial relation with a justification linked to

mixing length arguments. Boussinesq did not mention the name “mixing length” in his publication. Boussinesq performed simultaneously an average of the Navier-Stokes equations and a tensor closure of the eddy-viscosity type using an analogy to kinetic theory. In fact, Boussinesq did not notice that he was assuming a strong hypothesis when he performed his analogy with kinetic theory. Although in his 1895 publication Reynolds did not cite the 1877 Boussinesq publication, he did use the same method of averaging the Navier-Stokes equations and also proposed a tensor notation for the stresses.

Zero- and one-equation turbulent models are based on the mixing-length theory developed by Prandtl [5, 10]. By analogy with the kinetic theory of gases, which describes the molecular viscosity as proportional to the product of a molecular mean free path and the root mean square of the molecular velocity, Prandtl postulated that the turbulent eddy viscosity should be equal to the product of a characteristic length, called the mixing length, and a characteristic velocity which could be the root mean square of the fluctuating velocity. The mixing length can be viewed as the average distance traveled by an eddy before it gives up its momentum by mixing with the main flow. One-equation models derive a partial differential equation for the turbulent kinetic energy and use the square root of the turbulent kinetic energy as the characteristic velocity for the eddy viscosity model. In zero- and one-equation turbulent models, the mixing length is determined experimentally.

Two-equation models derive a partial differential equation for both the turbulent kinetic energy and a second turbulence variable. Jones and Launder (1972) [11] first derived the k-ε model which uses the dissipation of the turbulent kinetic energy as a second turbulence variable. The closure coefficients were empirically derived based on fully developed flow. Menter (1997) [12] has shown that the model performs poorly for complex flows involving severe pressure gradient or separation. Another widely used model, which uses the specific dissipation rate as the second turbulence variable is the k-ω model originally developed by Kolmogorov [8]. The model demonstrates superior performance for wall-bounded and low Reynolds number flows. Wilcox [13, 14] later derived several versions of the k-ω model. Menter’s (1992) [15] shear stress transport model combines the original Wilcox k-ω model [16], for use near walls, and the traditional k-ε model away from walls using a blending function. External flows, such as aerodynamic flow, include models developed by Spalart and Allmaras (1992) [17] and Baldwin and Barth (1990) [18].

More complex models are based on partial differential equations for all of the Reynolds stresses and turbulent fluxes and as a result require more computational power. An example is the Reynolds stress model (RSM) which involves the modeling of turbulent diffusion, pressure strain correlation and the turbulent dissipation rate. This modeling approach originates from the work by Launder (1974) [19]. For a discussion on the performance, applicability and limitations of the RSM, see the work of Speziale (1990) [20] or Wilcox (2006) [21].

Recent advances in large-scale scientific computing have made Direct Numerical Simulations (DNS) of the Navier-Stokes equations possible without incorporating a turbulence model. Direct numerical simulation is mainly used for relatively small domains in which the small scales of turbulence are large in comparison to the grid element size of the domain of interest because the whole range of spatial and temporal scales of the turbulence must be resolved. Even at low Reynolds numbers, the computational cost of DNS is always very high. Discussions on DNS involve the work of Bernard and Wallace (2002) [22] or Moin and Mahesh (1998) [23].

Retracing the traditional development of the turbulent-kinetic-energy equation, Phillips [24] developed an alternative turbulent-kinetic-energy transport equation based on fewer approximations than the traditional model and implementing strict definitions of the turbulent-kinetic-energy dissipation and molecular transport. This chapter focuses on addressing a few identifiable concerns with the traditional turbulence modeling based on two transport equations and proposes an alternate turbulent-kinetic-energy transport equation. To set this work in perspective relative to previous work, an overview of traditional turbulence modeling is included.

II. Governing Equation of Fluid Motion

The laws that govern fluid motion have been understood since the mid 1800s and the vector mathematics needed to fully analyze three-dimensional fluid mechanics was sufficiently understood only a few decades later. See for example the work of Navier [2], Stokes [3], Hamilton [25, 26, 27] and Boyer and Merzback [28]. The motion of a fluid with Newtonian properties is governed by the conservation of mass (continuity equation) and Newton's second law (or Navier-Stokes equations) combined with the appropriate boundary and initial conditions. This section develops the Reynolds-Averaged Navier-Stokes

equations (or RANS equations) for a Newtonian fluid which is a mathematical model of the fluid motion suitable for use in numerical calculations. A Newtonian fluid is defined as a fluid whose stress versus strain rate, or equivalently velocity gradient, curve is linear. The constant of proportionality is the called the dynamic viscosity.

A. Conservation of Mass: The Continuity Equation

The continuity equation describes the conservation of mass in a closed system. The mass of an isolated system cannot be changed as a result of processes acting inside the system. The differential form of the continuity equation is given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0 \tag{1.1}$$

where ρ is the fluid density, and $\mathbf{V}$ the flow velocity vector field.

B. Conservation of Momentum: Newton's Second Law

The law of conservation of linear momentum is a fundamental law of nature, and states that if no external force acts on a closed system of objects, the momentum of the closed system is constant. Momentum is defined as the mass of an object multiplied by its velocity. The second law states that the acceleration of an object is dependent upon the net force acting on the object and the mass of the object. After applying the continuity equation, Newton's second law for a flow field can be stated as

$$\rho\left[\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla)\mathbf{V}\right] = \nabla \cdot \bar{\bar{\boldsymbol{\sigma}}} - \rho g_0 \nabla Z \tag{1.2}$$

where Z is the geopotential altitude, the acceleration of gravity at sea level is $g_0 = 9.806645\ \text{m/s}^2$ and $\bar{\bar{\boldsymbol{\sigma}}}$ is the fluid stress tensor.

For Newtonian fluids, the Navier-Stokes equations are a set of nonlinear partial differential equations that describe the flow of a fluid whose stress depends linearly on velocity gradients and pressures. The Navier-Stokes equations ignore the fact that the fluid is made up of discrete molecules. The unsimplified Navier-Stokes equations do not have a general closed-form solution. However, the equations can be

simplified in a number of ways for problems with simple geometry, boundary conditions and an initial flow field. An example of such flow would be incompressible, inviscid, laminar, steady flow in a pipe. The Navier-Stokes equations assume that the fluid is Newtonian and continuum, that is, all length scales are large compared to the molecular mean free path. The Navier-Stokes equations developed from Newton's second law given in Eq. (1.2) can be written as follows assuming that the Earth's gravity is the only body force

$$\rho\left[\frac{\partial \mathbf{V}}{\partial t}+(\mathbf{V}\cdot\nabla)\mathbf{V}\right]=-\nabla\left(p+\frac{2}{3}\mu\nabla\cdot\mathbf{V}\right)+\nabla\cdot\left[2\mu\bar{\bar{\mathbf{S}}}(\mathbf{V})\right]+\mathbf{f}_g \tag{1.3}$$

where p is the thermodynamic pressure, μ the dynamic viscosity, $\mathbf{f}_g$ the gravity force vector per unit volume, ∇ is the gradient operator and $\bar{\bar{\mathbf{S}}}(\mathbf{V})$ the strain rate tensor which describes the rate at which a fluid element is deforming. The strain rate tensor in Cartesian coordinates is

$$\bar{\bar{\mathbf{S}}}(\mathbf{V})=\frac{1}{2}\begin{bmatrix}\left(\frac{\partial V_x}{\partial x}+\frac{\partial V_x}{\partial x}\right) & \left(\frac{\partial V_y}{\partial x}+\frac{\partial V_x}{\partial y}\right) & \left(\frac{\partial V_z}{\partial x}+\frac{\partial V_x}{\partial z}\right)\\ \left(\frac{\partial V_x}{\partial y}+\frac{\partial V_y}{\partial x}\right) & \left(\frac{\partial V_y}{\partial y}+\frac{\partial V_y}{\partial y}\right) & \left(\frac{\partial V_z}{\partial y}+\frac{\partial V_y}{\partial z}\right)\\ \left(\frac{\partial V_x}{\partial z}+\frac{\partial V_z}{\partial x}\right) & \left(\frac{\partial V_y}{\partial z}+\frac{\partial V_z}{\partial y}\right) & \left(\frac{\partial V_z}{\partial z}+\frac{\partial V_z}{\partial z}\right)\end{bmatrix} \tag{1.4}$$

The gravity force can be expressed in several different ways

$$\mathbf{f}_g=-\rho g\nabla H=-\rho g_0\nabla Z=-\nabla(g_0\rho Z)+g_0 Z\nabla(\rho) \tag{1.5}$$

where g is the acceleration of gravity, H is the geometric altitude, Z is the geopotential altitude and g_0 is the acceleration of gravity at sea level. The Navier-Stokes equations given in Eq. (1.2) with the gravity force expressed in terms of geopotential altitude given in Eq. (1.5) is

$$\rho\left[\frac{\partial \mathbf{V}}{\partial t}+(\mathbf{V}\cdot\nabla)\mathbf{V}\right]=-\nabla\left(p+g_0\rho Z+\frac{2}{3}\mu\nabla\cdot\mathbf{V}\right)+\nabla\cdot\left[2\mu\bar{\bar{\mathbf{S}}}(\mathbf{V})\right]+g_0 Z\nabla(\rho) \tag{1.6}$$

For convenience, we define a change of variables that combines several of the terms on the right-hand side of Eq. (1.6) and can be referred to as the total hydrostatic pressure

$$\hat{p} = p + g_0 \rho Z + \frac{2}{3}\mu \nabla \cdot \mathbf{V} \tag{1.7}$$

Using Eq. (1.7) in Eq. (1.6), the Navier-Stokes equations for a Newtonian fluid can be written as

$$\rho\left[\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V}\cdot\nabla)\mathbf{V}\right] = -\nabla\hat{p} + \nabla\cdot\left[2\mu\bar{\bar{\mathbf{S}}}(\mathbf{V})\right] + g_0 Z\nabla(\rho) \tag{1.8}$$

C. Conservation of Energy

The law of conservation of energy states that the total amount of energy in an isolated system remains constant over time. Energy can neither be created nor destroyed. It can only be transformed from one state to another. The specific total energy e is defined as

$$e \equiv u_e + \frac{1}{2}V^2 + g_0 Z \tag{1.9}$$

where u_e is the specific internal energy and V is the magnitude of the velocity vector. The first law of thermodynamics is an expression of the principle of conservation of energy. It usually can be formulated by stating that the change in the internal energy of a system is equal to the amount of heat supplied to the system minus the amount of work performed by the system on its surroundings. The first law of thermodynamics can be stated as

$$\frac{\partial \rho e}{\partial t} + \nabla\cdot(\rho \mathbf{V} e) = -\nabla\cdot\mathbf{q} + \nabla\cdot\left(\bar{\bar{\boldsymbol{\sigma}}}\cdot\mathbf{V}\right) + U_e''' \tag{1.10}$$

where $\mathbf{q}$ is the heat flux vector and U_e''' includes any other volumetric heating. The second term on the right-hand side is the net work per unit volume done on the fluid by the stress tensor. This includes the flow work, which is commonly added to the specific internal energy to define the thermodynamic property called enthalpy. Applying the vector identities (A.1), (A.3) and (A.5) given in Appendix A to Eq. (1.10) yields

$$\rho\frac{\partial e}{\partial t}+(\rho\mathbf{V}\cdot\nabla)e+e\frac{\partial\rho}{\partial t}+e[\nabla\cdot(\rho\mathbf{V})]=-\nabla\cdot\mathbf{q}+\bar{\bar{\boldsymbol{\sigma}}}\cdot(\nabla\mathbf{V})+\mathbf{V}\cdot(\nabla\cdot\bar{\bar{\boldsymbol{\sigma}}})+U_e'''\tag{1.11}$$

Applying the continuity equation and substituting the specific total energy, the first law of thermodynamics in the Eulerian reference frame results in

$$\rho\left[\frac{\partial}{\partial t}\left(u_e+\frac{1}{2}V^2\right)+(\mathbf{V}\cdot\nabla)\left(u_e+\frac{1}{2}V^2+g_0Z\right)\right]=-\nabla\cdot\mathbf{q}+\bar{\bar{\boldsymbol{\sigma}}}\cdot(\nabla\mathbf{V})+\mathbf{V}\cdot(\nabla\cdot\bar{\bar{\boldsymbol{\sigma}}})+U_e'''\tag{1.12}$$

Another useful transport equation is obtained by taking the dot product of Newton's second law with the fluid vector. Starting with Newton's second law given in Eq. (1.2) gives

$$\rho\mathbf{V}\cdot\left[\frac{\partial\mathbf{V}}{\partial t}+(\mathbf{V}\cdot\nabla)\mathbf{V}\right]=\mathbf{V}\cdot(\nabla\cdot\bar{\bar{\boldsymbol{\sigma}}})-\rho g_0(\mathbf{V}\cdot\nabla)Z\tag{1.13}$$

Applying the vector identity (A.4) from Appendix A to the left-hand side of this equation yields

$$\rho\left[\frac{\partial}{\partial t}\left(\frac{1}{2}V^2\right)+(\mathbf{V}\cdot\nabla)\left(\frac{1}{2}V^2\right)\right]=\mathbf{V}\cdot(\nabla\cdot\bar{\bar{\boldsymbol{\sigma}}})-\rho g_0(\mathbf{V}\cdot\nabla)Z\tag{1.14}$$

Because g_0 is a constant, this equation can be rearranged to obtain what is commonly called the mechanical energy equation.

$$\rho\left[\frac{\partial}{\partial t}\left(\frac{1}{2}V^2\right)+(\mathbf{V}\cdot\nabla)\left(\frac{1}{2}V^2+g_0Z\right)\right]=\mathbf{V}\cdot(\nabla\cdot\bar{\bar{\boldsymbol{\sigma}}})\tag{1.15}$$

Subtracting Eq. (1.15) from the first law of thermodynamics as expressed in Eq. (1.12), we can obtain another transport equation often referred to as the thermal energy equation

$$\rho\left[\frac{\partial u_e}{\partial t}+(\mathbf{V}\cdot\nabla)u_e\right]=-\nabla\cdot\mathbf{q}+\bar{\bar{\boldsymbol{\sigma}}}\cdot(\nabla\mathbf{V})+U_e'''\tag{1.16}$$

The equations of motion that describe the behavior of a system represent the conservation of mass, conservation of momentum and conservation of energy and are given in Eqs. (1.1), (1.2) and (1.16),

respectively. This set of three equations assumes a Newtonian fluid. Gravitational effect from planets other than the Earth, body forces such as electromagnetic, centrifugal and Coriolis forces were neglected.

III. The Reynolds-Averaged Navier-Stokes Equations

Due to the numerous velocity and length scales that are involved in turbulent flows, the numerical solution of the Navier-Stokes equations is extremely difficult and require such a fine mesh that the computational time becomes significantly infeasible for calculation. As is the case with laminar flows, turbulent flows must satisfy the continuity equations and the Navier-Stokes equations at every point in the flow field and at every instant in time. Solutions to these equations for turbulent flows can be obtained directly from computational fluid dynamics (CFD). However, obtaining such CFD solutions, which are commonly referred to as direct numerical simulations (DNS), are computationally very expensive. This is because such solutions require a computational grid fine enough to resolve the smallest turbulent eddies over the entire computational domain associated with the problem. For a complete review of DNS, see Moin and Malesh [23]. With today's technology, such solutions are computationally prohibitive for all but the simplest problems at very low Reynolds numbers. A less costly alternative to DNS is to solve only for the mean pressure and mean velocity fields using the Reynolds-Averaged Navier-Stokes equations. In this section, the Reynolds-Averaged Navier-Stokes equations of motion for turbulent flows are developed.

The Reynolds-Averaged Navier-Stokes equations are time averaged equations for a fluid flow. The idea behind the equations is Reynolds decomposition [4] where an instantaneous quantity is decomposed into its time-averaged and fluctuating quantities. The fluctuating part is defined such that its time-average is zero. The Reynolds decomposition can be viewed in mathematical statistics as the ensemble average of a random variable. The concept of an ensemble average is based upon the existence of independent statistical events. A random variable is distributed about the mean. Therefore, the ensemble average about the mean is zero, whereas, the ensemble average of the square of the fluctuation is not zero. Ensemble average properties are given in Eqs. (A.12)–(A.15) from Appendix A.

For a homogeneous fluid, the velocity vector and thermodynamic pressure can be written as a sum of a mean value and a turbulent fluctuating component

$$\mathbf{V} = \overline{\mathbf{V}} + \widetilde{\mathbf{V}} \tag{1.17}$$

$$p = \overline{p} + \widetilde{p} \tag{1.18}$$

where the overscore denotes an ensemble-mean value. It should be noted that

$$\overline{\rho} = \rho \tag{1.19}$$

Using Eq. (1.17) we have

$$\overline{\mathbf{V}} = \overline{\left(\overline{\mathbf{V}} + \widetilde{\mathbf{V}}\right)} = \overline{\overline{\mathbf{V}}} + \overline{\widetilde{\mathbf{V}}} = \overline{\mathbf{V}} + \overline{\widetilde{\mathbf{V}}} \quad \Rightarrow \quad \overline{\widetilde{\mathbf{V}}} = 0 \tag{1.20}$$

$$\overline{p} = \overline{\overline{p} + \widetilde{p}} = \overline{\overline{p}} + \overline{\widetilde{p}} \quad \Rightarrow \quad \overline{\widetilde{p}} = 0 \tag{1.21}$$

Equations (1.20) and (1.21) show that the time-averaged of the fluctuating part is zero. For a detail discussion on averaging, see Phillips [24] or Wilcox [29].

A. Ensemble Average of the Continuity Equation

The ensemble average of the continuity equation given in (1.1) yield

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \overline{\mathbf{V}}\right) = 0 \tag{1.22}$$

An equation for the fluctuating components is obtained by substituting Eq. (1.17) in Eq. (1.1)

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \overline{\mathbf{V}} + \rho \widetilde{\mathbf{V}}\right) = 0 \tag{1.23}$$

Subtracting Eq. (1.22) from Eq. (1.23) yields the continuity equation for the fluctuations

$$\nabla \cdot \left(\rho \widetilde{\mathbf{V}}\right) = 0 \tag{1.24}$$

The ensemble-averaged steady-state incompressible form of the continuity equation is obtained from Eq. (1.132)

$$\nabla \cdot \overline{\mathbf{V}} = 0 \tag{1.25}$$

B. Ensemble Average of the Navier-Stokes Equations

Similarly to the continuity equation, the Navier-Stokes equations can be simplified using the Reynolds decomposition and substituting in the sum of the steady component and perturbations to the velocity profile and taking the mean value. The resulting equation contains a nonlinear term, known as Reynolds stresses, which gives rise to turbulence. The ensemble average of the linear strain rate tensor is simply

$$\overline{\overline{\overline{\mathbf{S}}}(\mathbf{V})} = \overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}) \tag{1.26}$$

The ensemble average of the Navier-Stokes equations given in Eq. (1.6) with the gravity force expressed in terms of geopotential altitude is

$$\overline{\rho\left[\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V}\cdot\nabla)\mathbf{V}\right]} = \overline{-\nabla\left(p + \frac{2}{3}\mu\nabla\cdot\mathbf{V}\right) + \nabla\cdot\left[2\mu\overline{\overline{\mathbf{S}}}(\mathbf{V})\right] - \nabla(g_0\rho Z) + g_0 Z\nabla(\rho)} \tag{1.27}$$

which can be simplified into

$$\rho\left[\frac{\partial \overline{\mathbf{V}}}{\partial t} + \overline{(\mathbf{V}\cdot\nabla)\mathbf{V}}\right] = -\nabla\left(\overline{p} + \rho g_0 Z + \frac{2}{3}\mu\nabla\cdot\overline{\mathbf{V}}\right) + \nabla\cdot\left[2\mu\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}})\right] + g_0 Z\nabla(\rho) \tag{1.28}$$

Equation (1.17) yields the mathematical identity

$$\begin{aligned}\overline{(\mathbf{V}\cdot\nabla)\mathbf{V}} &= \overline{\left[(\overline{\mathbf{V}}+\widetilde{\mathbf{V}})\cdot\nabla\right](\overline{\mathbf{V}}+\widetilde{\mathbf{V}})} \\ &= \overline{(\overline{\mathbf{V}}\cdot\nabla)\overline{\mathbf{V}} + (\overline{\mathbf{V}}\cdot\nabla)\widetilde{\mathbf{V}} + (\widetilde{\mathbf{V}}\cdot\nabla)\overline{\mathbf{V}} + (\widetilde{\mathbf{V}}\cdot\nabla)\widetilde{\mathbf{V}}} \\ &= (\overline{\mathbf{V}}\cdot\nabla)\overline{\mathbf{V}} + \overline{(\widetilde{\mathbf{V}}\cdot\nabla)\widetilde{\mathbf{V}}}\end{aligned} \tag{1.29}$$

Using Eq. (1.29) in Eq. (1.28) gives

$$\begin{aligned}\rho\left[\frac{\partial \overline{\mathbf{V}}}{\partial t} + (\overline{\mathbf{V}}\cdot\nabla)\overline{\mathbf{V}}\right] &= -\nabla\left(\overline{p} + \rho g_0 Z + \frac{2}{3}\mu\nabla\cdot\overline{\mathbf{V}}\right) \\ &+ \nabla\cdot\left[2\mu\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}})\right] + g_0 Z\nabla(\rho) - \rho\overline{(\widetilde{\mathbf{V}}\cdot\nabla)\widetilde{\mathbf{V}}}\end{aligned} \tag{1.30}$$

The last term in Eq. (1.30) is the only term that involves the turbulent fluctuations. Using the mathematical identity (A.1) from Appendix A, this last term can be rewritten as

$$\rho\overline{\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}}=\overline{\left(\rho\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}}=\overline{\nabla\cdot\left(\rho\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}\right)}-\overline{\widetilde{\mathbf{V}}\left(\nabla\cdot\left(\rho\widetilde{\mathbf{V}}\right)\right)} \tag{1.31}$$

where $\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}$ is a second order tensor given in Cartesian coordinates by

$$\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}=\begin{bmatrix}\widetilde{V}_x\widetilde{V}_x & \widetilde{V}_x\widetilde{V}_y & \widetilde{V}_x\widetilde{V}_z\\ \widetilde{V}_y\widetilde{V}_x & \widetilde{V}_y\widetilde{V}_y & \widetilde{V}_y\widetilde{V}_z\\ \widetilde{V}_z\widetilde{V}_x & \widetilde{V}_z\widetilde{V}_y & \widetilde{V}_z\widetilde{V}_z\end{bmatrix} \tag{1.32}$$

From the continuity equation for the fluctuations given in Eq. (1.24), Eq. (1.31) becomes

$$\rho\overline{\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}}=\nabla\cdot\left(\rho\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}\right) \tag{1.33}$$

For a Newtonian fluid, the pseudo molecular stress tensor resulting from the mean fluid motion can be expressed as

$$\overline{\overline{\boldsymbol{\sigma}}}=2\mu\,\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}})-\left(\overline{p}+\frac{2}{3}\mu\nabla\cdot\overline{\mathbf{V}}\right)\overline{\overline{\boldsymbol{\delta}}} \tag{1.34}$$

where $\overline{\overline{\boldsymbol{\delta}}}$ is the kronecker delta.

First suggested by Reynolds (1895) [4], the symmetric turbulent stress tensor is defined in Cartesian coordinates as

$$\overline{\overline{\tau}}=-\rho\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}=-\rho\begin{bmatrix}\overline{\widetilde{V}_x\widetilde{V}_x} & \overline{\widetilde{V}_x\widetilde{V}_y} & \overline{\widetilde{V}_x\widetilde{V}_z}\\ \overline{\widetilde{V}_y\widetilde{V}_x} & \overline{\widetilde{V}_y\widetilde{V}_y} & \overline{\widetilde{V}_y\widetilde{V}_z}\\ \overline{\widetilde{V}_z\widetilde{V}_x} & \overline{\widetilde{V}_z\widetilde{V}_y} & \overline{\widetilde{V}_z\widetilde{V}_z}\end{bmatrix} \tag{1.35}$$

The turbulent stresses $\overline{\widetilde{V}_i\widetilde{V}_j}$ are called the Reynolds stresses. The Reynolds stress tensor is the stress tensor due to random turbulent fluctuations in fluid momentum. The stress is obtained from an average over these fluctuations. The Reynolds stress tensor is a symmetric tensor whose six components are unknown. Many

scientists have tried to model the Reynolds stress in various different ways and a transport equation for the Reynolds stress is often proposed. See the work of Jones and Launder [30] or Wilcox [16]. Substituting Eq. (1.35) in Eq. (1.33), the turbulent momentum becomes

$$\rho\overline{\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}} = -\nabla\cdot\bar{\bar{\boldsymbol{\tau}}} \tag{1.36}$$

Using Eq. (1.36) in Eq. (1.30), the Reynolds-averaged momentum equations, or RANS equations are

$$\begin{aligned}\rho\left[\frac{\partial\overline{\mathbf{V}}}{\partial t}+\left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\mathbf{V}}\right] &= -\nabla\left(\bar{p}+\rho g_0 Z+\frac{2}{3}\mu\nabla\cdot\overline{\mathbf{V}}\right)\\ &+ \nabla\cdot\left[2\mu\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})+\bar{\bar{\tau}}\right]+g_0 Z\nabla(\rho)\end{aligned} \tag{1.37}$$

Combining the RANS equations given in Eq. (1.37) with the mean continuity equation will not produce a complete formulation for the mean velocity and pressure fields, unless one has some method by which to express the six unknown components of the symmetric Reynolds stress tensor in terms of the mean flow. The purpose of a turbulence model is to provide mathematical relations that express the components of the Reynolds stress tensor in terms of the mean flow.

IV. Traditional Turbulence Closure

The nonlinear Reynolds stress term from the convective acceleration requires additional modeling to close the RANS equation and has led to the creation of many different turbulent models. The most sophisticated turbulence models commonly available today are based on the use of separate modeled transport equations for each of the six unknown components of the Reynolds stress tensor. Turbulence models of this type are classified as stress-transport models. Because of their complexity and the large number of differential equations involved, stress-transport models impose a large computational burden. Furthermore, the stress-transport models that have been developed to date have not shown the significant improvements in accuracy over simpler models, which was originally anticipated. For these reasons, such detailed turbulence models are not extensively used for CFD analysis at the present time. A simpler model consists in relating the Reynolds stresses to the mean velocity gradients, a common practice based on the Boussinesq's hypothesis.

A. Modeling the Reynolds Stress Tensor from the Boussinesq's Hypothesis

Commonly employed turbulence models, such as the k-ω or k-ε models, are based on a simplifying hypothesis for the Reynolds stress tensor, which was first introduced by Boussinesq (1877) [6]. The fact that turbulence transport seems to be diffusive in nature suggest that we might express the Reynolds stress tensor as a linear function of the mean strain rate tensor. Boussinesq was one of the earliest scientists to propose an equation that relates what are now called the Reynolds stresses to the mean flow and close the system of equation. Boussinesq postulated that the momentum transfer caused by turbulent eddies can be modeled using a parameter that is now called the turbulent eddy viscosity. This is in direct analogy with how the momentum transfer caused by the molecular motion in a gas can be described by a molecular viscosity. Boussinesq's hypothesis in its general form states that the Reynolds stress tensor is given by

$$\bar{\bar{\boldsymbol{\tau}}} = 2\mu_t \bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) - \frac{2}{3}\left(\rho k + \mu_t \nabla \cdot \overline{\mathbf{V}}\right)\bar{\bar{\boldsymbol{\delta}}} \tag{1.38}$$

where the coefficient μ_t is called the dynamic eddy viscosity and k is the turbulent kinetic energy per unit mass and is traditionally defined as one half of the mean square magnitude of the turbulent velocity fluctuations

$$k \equiv \frac{1}{2}\overline{\widetilde{\mathbf{V}} \cdot \widetilde{\mathbf{V}}} = \frac{1}{2}\left(\overline{\widetilde{V}_x{}^2} + \overline{\widetilde{V}_y{}^2} + \overline{\widetilde{V}_z{}^2}\right) = \frac{1}{2}\overline{\widetilde{V}^2} \tag{1.39}$$

The Boussinesq hypothesis is sometimes presented in the literature in a form that includes only the first term on the right-hand side of Eq. (1.38). However, this is not consistent with the fundamental definition in Eq. (1.35). The trace of the turbulent stress tensor $\bar{\bar{\boldsymbol{\tau}}}$ given in Eq. (1.35) is $-2\rho k$. The trace of the strain rate tensor $\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})$ given in Eq. (1.4) is $\text{trace}\left(\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})\right) = \nabla \cdot \overline{\mathbf{V}}$. The trace of the right-hand side of Eq. (1.38) is $2\mu_t \nabla \cdot \overline{\mathbf{V}} - 2\rho k - 2\mu_t \nabla \cdot \overline{\mathbf{V}} = -2\rho k = \text{trace}(\bar{\bar{\boldsymbol{\tau}}})$. The additional diagonal term in Eq. (1.38) must be included to make the two definitions consistent. Boussinesq's hypothesis can be viewed as an analogy between the Reynolds stress tensor and a relation for the molecular stress tensor obtained from the kinetic theory of gases. The molecular stress tensor in an ideal gas can be related statistically to the molecular velocity

components in precisely the same way that the Reynolds stress tensor was related to the turbulent velocity fluctuations.

According to Boussinesq's hypothesis, the turbulent eddy viscosity is assumed to be a scalar like the molecular viscosity. Originally, Boussinesq assumed only that the turbulent eddy viscosity was directionally independent. However, in the application of his theory, he assumed that the eddy viscosity was spatially constant as well. Such a constant value for the turbulent eddy viscosity could occur only if the turbulence were homogeneous. A constant eddy viscosity is not expected in general. For example, in turbulent shear flows near a smooth wall the eddy viscosity cannot be constant, because both the mean and fluctuating components of fluid velocity must vanish at a no-slip surface. Hence, the Reynolds stress tensor must vanish at a no-slip surface, in spite of the fact that the mean-strain-rate tensor remains finite.

The eddy viscosity is not a thermodynamic property of the fluid in the same way that molecular viscosity is. The eddy viscosity is a function of the turbulent motion just as the molecular viscosity is a function of the molecular motion. However, the random motion of molecules in an ideal gas depends only on the absolute temperature. For many flows of practical interest, the temperature variations within the flow are small compared with the level of temperature above the absolute zero. Hence, the molecular viscosity is often approximated as being constant throughout the flow. In contrast, the eddy viscosity can seldom be treated as a constant. The turbulent velocity fluctuations will always vanish in the laminar region of a shear flow immediately adjacent to a smooth surface, whereas the random molecular motion in an ideal gas would cease at a solid wall only if the temperature of the gas went to absolute zero at the surface. In this dissertation, the eddy viscosity is assumed to be isotropic but not necessarily homogeneous. This is not to say that the turbulence itself is isotropic because the mean-strain-rate tensor is not isotropic. We are simply assuming that the eddy viscosity at any given point in space and time is independent of direction.

Using Eq. (1.38) in Eq. (1.37), the Boussinesq-RANS equations are

$$\begin{aligned} \rho\left[\frac{\partial\overline{\mathbf{V}}}{\partial t}+\left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\mathbf{V}}\right] &= -\nabla\left(\overline{p}+\rho g_0 Z+\frac{2}{3}\rho k+\frac{2}{3}\left(\mu+\mu_t\right)\nabla\cdot\overline{\mathbf{V}}\right) \\ &+ \nabla\cdot\left[2\left(\mu+\mu_t\right)\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}})\right]+g_0 Z\,\nabla\rho \end{aligned} \tag{1.40}$$

For convenience, we define a change of variables that combines several of the terms on the right-hand side of Eq. (1.40) to form the pseudo mean pressure

$$\hat{\bar{p}} = \bar{p} + \rho g_0 Z + \frac{2}{3}\rho k + \frac{2}{3}(\mu + \mu_t)\nabla \cdot \overline{\mathbf{V}} \tag{1.41}$$

The steady-state incompressible Boussinesq-RANS equations become

$$\left(\overline{\mathbf{V}} \cdot \nabla\right)\overline{\mathbf{V}} = -\frac{1}{\rho}\nabla\hat{\bar{p}} + \nabla \cdot \left[2(\nu + \nu_t)\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})\right] \tag{1.42}$$

The Boussinesq hypothesis allows us to express the six independent components of the Reynolds stress tensor in terms of the mean velocity vector and two scalars; the eddy viscosity ν_t and the turbulent kinetic energy per unit mass k. Both of these scalars are unknown functions of space and time. In order to close the formulation with the Boussinesq's hypothesis, the dynamic eddy viscosity and the turbulent kinetic energy per unit mass must be related to other flow properties. Several methods have been suggested for closing the formulation based on Boussinesq's hypothesis and are commonly classified as zero, one, or two-equation models, based on the number of differential equation used.

Most zero-equation models, also known as algebraic models, are based on a hypothesis first made by Prandtl [5]. This hypothesis is referred to as the mixing-length theory. Prandtl hypothesized that the turbulence characteristics were related to a characteristic length and velocity scale associated with the turbulent fluctuations. Thus, the closing equations are simply algebraic relationships between the turbulence parameters. Zero-equations models are classified as incomplete models because they require properties of the turbulent flow field to be known a priori.

Most one-equation models are based on a subsequent hypothesis by Prandtl [10]. In his development, Prandtl hypothesized that the eddy viscosity was proportional to the product of the square root of the turbulent kinetic energy per unit mass and a characteristic length scale. Prandtl used a modeled version of the turbulent-kinetic-energy transport equation and related the eddy viscosity to the turbulent kinetic energy algebraically. The length scale was also calculated algebraically from the mean flow. The distinguishing factors of most one-equation models is that they model the turbulent kinetic energy per unit mass by a differential equation, they express the eddy viscosity as a function of the turbulent kinetic energy per unit mass and they calculate some type of length scale from the mean fluid flow. Many one-equation models have been proposed including those by Emmons [31], Glushko [32] and Wolfshtein [33]. Feriss and Atwell

[34] also proposed a one-equation model based on the turbulent kinetic energy but do not use the Boussinesq's hypothesis. Other one-equation models that are based on some transport property other than turbulent kinetic energy include models by Nee and Kovasznay [35], Sekundov [36], Baldwin and Barth [18], Spalart and Allmaras [17] and Menter [37].

Widely used turbulence models are two-equation models based on an approach to turbulence modeling that was originally proposed by Kolmogorov [8]. Kolmogorov employed a modeled differential equation for the transport of turbulent kinetic energy and also developed a modeled differential transport equation for the second scalar turbulence variable, for which he used a symbol ω. This turbulence variable is a frequency characteristic of the turbulent-kinetic-energy dissipation process and it is sometimes said to be proportional to the root-mean-square magnitude of the fluctuating turbulent vorticity. Researchers have since built upon Kolmogorov's two-equation approach, refining the modeled transport equations and solving for other turbulence parameters such as the turbulent-kinetic-energy dissipation rate. The feature common to all two-equation turbulence models is that the eddy viscosity is modeled as a function of two scalar turbulence variables, which are both computed from differential transport equations. Many two-equation models can be classified as either k-ω or k-ε models, where k is the turbulent kinetic energy per unit mass, ω is a characteristic turbulence frequency and ε is the turbulent-kinetic-energy dissipation rate per unit mass. Two-equation models do not require an algebraic relation that describes any turbulence variable in a prescribed manner from the mean flow. Two-equation models that are based on transport properties other than ε or ω include models suggested by Rotta [38, 39], Zeierman and Wolfshtein [40] and Speziale et al. [20].

B. Reynolds Stress Transport

Traditional turbulence models are based on a differential transport equation for the Reynolds stress tensor. The equations of motion for turbulent flow, comprising the continuity equation and the three components of the momentum equation, can be expressed in terms of the three components of the mean velocity vector, the mean pressure and the Reynolds stress tensor. To complete this formulation, we need some means of relating the six independent components of the symmetric Reynolds stress tensor to the mean velocity and pressure fields. A differential transport equation for the Reynolds stress tensor can be developed from the Navier-Stokes equation. Assuming constant viscosity, we have

$$\nabla \cdot \left(2\mu \,\bar{\bar{\mathbf{S}}}(\mathbf{V})\right) = 2\mu \,\nabla \cdot \left(\bar{\bar{\mathbf{S}}}(\mathbf{V})\right) \tag{1.43}$$

The mathematical identity (A.2) from Appendix A gives

$$2\mu \,\nabla \cdot \left(\bar{\bar{\mathbf{S}}}(\mathbf{V})\right) = 2\mu \left(\nabla^2 \mathbf{V} + \nabla(\nabla \cdot \mathbf{V})\right) \tag{1.44}$$

With constant viscosity, the Navier Stokes equation given in Eq. (1.6) expressed using the geopotential altitude for the gravitational and the two above equations give

$$\rho\left[\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla)\mathbf{V}\right] = -\nabla\left(p + g_0 \rho Z - \frac{1}{3}\mu \nabla \cdot \mathbf{V}\right) + \mu \nabla^2 \mathbf{V} + g_0 Z \nabla \rho \tag{1.45}$$

Multiplying Eq. (1.45) by the fluctuating components of the velocity $\tilde{\mathbf{V}}$, applying Eqs. (1.17)–(1.18) and taking the ensemble average yield

$$\begin{aligned} &\rho\left(\bar{\tilde{\mathbf{V}}}\frac{\partial \bar{\mathbf{V}}}{\partial t} + \overline{\tilde{\mathbf{V}}\frac{\partial \tilde{\mathbf{V}}}{\partial t}} + \bar{\tilde{\mathbf{V}}}\left[(\bar{\mathbf{V}} \cdot \nabla)\bar{\mathbf{V}}\right] + \bar{\tilde{\mathbf{V}}}\left[(\bar{\mathbf{V}} \cdot \nabla)\tilde{\mathbf{V}}\right] + \bar{\tilde{\mathbf{V}}}\left[(\tilde{\mathbf{V}} \cdot \nabla)\bar{\mathbf{V}}\right] + \bar{\tilde{\mathbf{V}}}\left[(\tilde{\mathbf{V}} \cdot \nabla)\tilde{\mathbf{V}}\right]\right) = \\ &-\bar{\tilde{\mathbf{V}}}\nabla\left(\bar{p} + g_0 \rho Z - \frac{1}{3}\mu \nabla \cdot \bar{\mathbf{V}}\right) - \overline{\tilde{\mathbf{V}}\nabla\left(\tilde{p} - \frac{1}{3}\mu \nabla \cdot \tilde{\mathbf{V}}\right)} + \mu \bar{\tilde{\mathbf{V}}}\nabla^2 \bar{\mathbf{V}} + \mu \overline{\tilde{\mathbf{V}}\nabla^2 \tilde{\mathbf{V}}} + \bar{\tilde{\mathbf{V}}} g_0 Z \nabla \rho \end{aligned} \tag{1.46}$$

Equation (1.46) can be reduced using the equality $\bar{\tilde{\mathbf{V}}} = 0$ given in Eq. (1.20)

$$\rho\left(\overline{\tilde{\mathbf{V}}\frac{\partial \tilde{\mathbf{V}}}{\partial t}} + \bar{\tilde{\mathbf{V}}}(\bar{\mathbf{V}} \cdot \nabla)\bar{\mathbf{V}} + \overline{\tilde{\mathbf{V}}(\bar{\mathbf{V}} \cdot \nabla)\tilde{\mathbf{V}}} + \overline{\tilde{\mathbf{V}}(\tilde{\mathbf{V}} \cdot \nabla)\tilde{\mathbf{V}}}\right) = -\overline{\tilde{\mathbf{V}}\nabla\left(\tilde{p} - \frac{1}{3}\mu \nabla \cdot \tilde{\mathbf{V}}\right)} + \mu \overline{\tilde{\mathbf{V}}\nabla^2 \tilde{\mathbf{V}}} \tag{1.47}$$

Each term in Eq. (1.47) is a second order tensor containing nine scalar components. Applying the following mathematical identities

$$\overline{\tilde{\mathbf{V}}\frac{\partial \tilde{\mathbf{V}}}{\partial t}} + \left(\overline{\tilde{\mathbf{V}}\frac{\partial \tilde{\mathbf{V}}}{\partial t}}\right)^T = \frac{\partial \overline{\tilde{\mathbf{V}}\tilde{\mathbf{V}}}}{\partial t} \tag{1.48}$$

$$\overline{\widetilde{\mathbf{V}}\left(\overline{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}}+\left(\overline{\widetilde{\mathbf{V}}\left(\overline{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}}\right)^{T}=\left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}} \tag{1.49}$$

$$\overline{\widetilde{\mathbf{V}}\left[\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}\right]}+\left(\overline{\widetilde{\mathbf{V}}\left[\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}\right]}\right)^{T}=\bar{\bar{\boldsymbol{\chi}}}(\widetilde{\mathbf{V}})-\left(\nabla\cdot\overline{\widetilde{\mathbf{V}}}\right)\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}} \tag{1.50}$$

$$\overline{\widetilde{\mathbf{V}}\nabla^{2}\widetilde{\mathbf{V}}}+\left(\overline{\widetilde{\mathbf{V}}\nabla^{2}\widetilde{\mathbf{V}}}\right)^{T}=\nabla^{2}\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}-\bar{\bar{\boldsymbol{\varepsilon}}}(\widetilde{\mathbf{V}}) \tag{1.51}$$

The Reynolds stress transport equation given in Eq. (1.47) can be rewritten as

$$\begin{aligned}&\rho\left(\frac{\partial\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}}{\partial t}+\left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}-\bar{\bar{\boldsymbol{\gamma}}}(\overline{\mathbf{V}},\widetilde{\mathbf{V}})+\bar{\bar{\boldsymbol{\chi}}}(\widetilde{\mathbf{V}})+\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}\right)\\&=-\overline{\widetilde{\mathbf{V}}\nabla\widehat{\widetilde{p}}}-\left(\overline{\widetilde{\mathbf{V}}\nabla\widehat{\widetilde{p}}}\right)^{T}+\mu\nabla^{2}\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}-\mu\bar{\bar{\boldsymbol{\varepsilon}}}(\widetilde{\mathbf{V}})\end{aligned} \tag{1.52}$$

where

$$\widehat{\widetilde{p}}=\widetilde{p}-\frac{1}{3}\mu\,\nabla\cdot\widetilde{\mathbf{V}} \tag{1.53}$$

and the following definition are used

$$\bar{\bar{\boldsymbol{\gamma}}}(\overline{\mathbf{V}},\widetilde{\mathbf{V}})=\overline{\widetilde{\mathbf{V}}\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}}-\left(\overline{\widetilde{\mathbf{V}}\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}}\right)^{T} \tag{1.54}$$

$$\bar{\bar{\boldsymbol{\chi}}}(\widetilde{\mathbf{V}})=\begin{bmatrix}\nabla\cdot\left(\overline{\widetilde{V}_x\widetilde{V}_x\widetilde{\mathbf{V}}}\right) & \nabla\cdot\left(\overline{\widetilde{V}_x\widetilde{V}_y\widetilde{\mathbf{V}}}\right) & \nabla\cdot\left(\overline{\widetilde{V}_x\widetilde{V}_z\widetilde{\mathbf{V}}}\right)\\ \nabla\cdot\left(\overline{\widetilde{V}_y\widetilde{V}_x\widetilde{\mathbf{V}}}\right) & \nabla\cdot\left(\overline{\widetilde{V}_y\widetilde{V}_y\widetilde{\mathbf{V}}}\right) & \nabla\cdot\left(\overline{\widetilde{V}_y\widetilde{V}_z\widetilde{\mathbf{V}}}\right)\\ \nabla\cdot\left(\overline{\widetilde{V}_z\widetilde{V}_x\widetilde{\mathbf{V}}}\right) & \nabla\cdot\left(\overline{\widetilde{V}_z\widetilde{V}_y\widetilde{\mathbf{V}}}\right) & \nabla\cdot\left(\overline{\widetilde{V}_z\widetilde{V}_z\widetilde{\mathbf{V}}}\right)\end{bmatrix} \tag{1.55}$$

$$\bar{\bar{\boldsymbol{\varepsilon}}}(\widetilde{\mathbf{V}})=2\begin{bmatrix}\overline{\nabla\widetilde{V}_x\cdot\nabla\widetilde{V}_x} & \overline{\nabla\widetilde{V}_x\cdot\nabla\widetilde{V}_y} & \overline{\nabla\widetilde{V}_x\cdot\nabla\widetilde{V}_z}\\ \overline{\nabla\widetilde{V}_y\cdot\nabla\widetilde{V}_x} & \overline{\nabla\widetilde{V}_y\cdot\nabla\widetilde{V}_y} & \overline{\nabla\widetilde{V}_y\cdot\nabla\widetilde{V}_z}\\ \overline{\nabla\widetilde{V}_z\cdot\nabla\widetilde{V}_x} & \overline{\nabla\widetilde{V}_z\cdot\nabla\widetilde{V}_y} & \overline{\nabla\widetilde{V}_z\cdot\nabla\widetilde{V}_z}\end{bmatrix} \tag{1.56}$$

$$\overline{\overline{\boldsymbol{\pi}}}(\widetilde{\mathbf{V}},\widetilde{p}) = \frac{1}{\rho}\overline{\widetilde{\mathbf{V}}\nabla\widetilde{p}} + \frac{1}{\rho}\left(\overline{\widetilde{\mathbf{V}}\nabla\widetilde{p}}\right)^{T} + \overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}} \tag{1.57}$$

Notice that the tensor $\overline{\overline{\boldsymbol{\gamma}}}$ depends only on the components of $\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}$ and the spatial derivatives of the mean velocity field. On the other hand, the tensors $\overline{\overline{\boldsymbol{\varepsilon}}}$, $\overline{\overline{\boldsymbol{\pi}}}$ and $\overline{\overline{\boldsymbol{\chi}}}$ involve other unknown correlations of the velocity and pressure fluctuations. After rearranging Eq. (1.52) and using Eqs. (1.54)–(1.57), we obtain the Reynolds-stress-transport equation

$$\frac{\partial\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}}{\partial t} + \left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}} = \overline{\overline{\boldsymbol{\gamma}}}(\overline{\mathbf{V}},\widetilde{\mathbf{V}}) - \nu\,\overline{\overline{\boldsymbol{\varepsilon}}}(\widetilde{\mathbf{V}}) + \nu\nabla^{2}\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}} - \overline{\overline{\boldsymbol{\chi}}}(\widetilde{\mathbf{V}}) - \overline{\overline{\boldsymbol{\pi}}}(\widetilde{\mathbf{V}},\widetilde{p}) \tag{1.58}$$

where ν is the kinematic viscosity $\nu = \mu/\rho$. The tensor $\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}$ is the negative of the specific Reynolds stress tensor, as given in Eq. (1.35)

$$\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}} = -\frac{\overline{\overline{\boldsymbol{\tau}}}}{\rho} \tag{1.59}$$

The tensor equation given by Eq. (1.59) provides scalar transport equations for the components of the specific Reynolds stress tensor. Because this tensor is symmetric, there are only six independent scalar equations. However, many of the scalar terms that make up the components of the tensors on the right-hand side of Eq. (1.58) are new unknowns. Counting these unknowns reveals that we have added more unknowns than equations. This is the essence of the turbulence closure problem. We could develop additional transport equations for the new unknowns in Eq. (1.58) by taking other moments of the Navier-Stokes equations. However, each time we add new equations in this manner, we also add more unknowns. At no point will this procedure produce a number of equations equal to the number of unknowns.

The most commonly used method for closing the Reynolds-averaged formulation is to use some type of approximation to model the unknown terms on the right-hand side of Eq. (1.59). One possible approach is to model the unknown terms on the right-hand side of Eq. (1.58) and then use the resulting Reynolds-stress-transport model to evaluate the Reynolds stresses. On the other hand, we could return to the Reynolds-averaged Navier-Stokes equations given by Eq. (1.37) and make some approximations that

would allow us to model the Reynolds stress tensor directly. The Boussinesq hypothesis is commonly used for this purpose.

However, Boussinesq's hypothesis alone does not close the Reynolds-averaged formulation. To complete any Boussinesq-based turbulence model, we must develop additional equations that allow us to relate the eddy viscosity and the turbulence kinetic energy per unit mass to the mean flow. Although the Reynolds-stress-transport equation is not required for Boussinesq-based models, Eq. (1.58) can provide significant insight into how the turbulence kinetic energy per unit mass depends on the mean flow.

C. Traditional Kinetic Energy Transport Equation

A transport differential equation for the turbulent kinetic energy is developed to account for the turbulent transport of momentum. The turbulent kinetic energy associated with turbulent velocity fluctuations is generated directly from velocity gradients in the mean flow, i.e. fluid shear and friction. However, turbulent kinetic energy is also dissipated with time as a result of molecular viscosity and is transported from one location to another by the mean fluid motion, the turbulent velocity fluctuations and the molecular diffusion. Thus, the specific turbulent kinetic energy is not expected to be a function of only local gradients in the mean flow. Instead, the level of kinetic energy associated with the turbulent fluctuations at any point in space and time must depend on the history of flow, i.e. the specific turbulent kinetic energy at any point in the flow field depends on where the fluid has been as well as the local gradients in the mean motion. To account for such effects, the specific turbulent kinetic energy must be related to the mean flow through a differential transport equation.

The transport of turbulent kinetic energy results from the transfer down the turbulence energy cascade. The total specific turbulent kinetic energy associated with the turbulence was defined in Eq. (1.39) to be one-half of the mean squared magnitude of the turbulent fluctuations. This specific turbulent kinetic energy can also be expressed in terms of one half of the trace of the negative of the specific Reynolds stress tensor defined in Eq. (1.35)

$$k = \frac{1}{2}\overline{\widetilde{V}^{\,2}} = \frac{1}{2}\left(\overline{\widetilde{V}_x^{\,2}} + \overline{\widetilde{V}_y^{\,2}} + \overline{\widetilde{V}_z^{\,2}}\right) = \frac{1}{2}\,\text{trace}\left(-\frac{\overline{\overline{\tau}}}{\rho}\right) = \frac{1}{2}\,\text{trace}\left(\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}\right) \tag{1.60}$$

For a Newtonian fluid, a transport equation for turbulent kinetic energy per unit mass is typically developed directly from the trace of the Reynolds-stress-transport equation given in Eq. (1.58)

$$\frac{\partial k}{\partial t} + \left(\overline{\mathbf{V}} \cdot \nabla\right)k = \gamma_k - \nu\varepsilon_k + \nu\nabla^2 k - \chi_k - \pi_k \tag{1.61}$$

where

$$\gamma_k = \frac{1}{2}\,\mathrm{trace}\left(\bar{\bar{\gamma}}(\overline{\mathbf{V}}, \widetilde{\mathbf{V}})\right) \tag{1.62}$$

$$\varepsilon_k = \frac{1}{2}\,\mathrm{trace}\left(\bar{\bar{\varepsilon}}(\widetilde{\mathbf{V}})\right) \tag{1.63}$$

$$\chi_k = \frac{1}{2}\,\mathrm{trace}\left(\bar{\bar{\chi}}(\widetilde{\mathbf{V}})\right) \tag{1.64}$$

$$\pi_k = \frac{1}{2}\,\mathrm{trace}\left(\bar{\bar{\pi}}(\widetilde{\mathbf{V}}, \widetilde{p})\right) \tag{1.65}$$

with

$$\gamma_k = \frac{\bar{\bar{\boldsymbol{\tau}}}}{\rho} : \bar{\bar{\mathbf{J}}}(\overline{\mathbf{V}}) \tag{1.66}$$

where the Jacobian tensor $\bar{\bar{\mathbf{J}}}(\overline{\mathbf{V}})$ can be expanded in Cartesian coordinates as

$$\bar{\bar{\mathbf{J}}}(\overline{\mathbf{V}}) = \begin{bmatrix} \dfrac{\partial V_x}{\partial x} & \dfrac{\partial V_x}{\partial y} & \dfrac{\partial V_x}{\partial z} \\ \dfrac{\partial V_y}{\partial x} & \dfrac{\partial V_y}{\partial y} & \dfrac{\partial V_y}{\partial z} \\ \dfrac{\partial V_z}{\partial x} & \dfrac{\partial V_z}{\partial y} & \dfrac{\partial V_z}{\partial z} \end{bmatrix} \tag{1.67}$$

Equation (1.63) can be expressed as

$$\varepsilon_k = \overline{\tilde{\bar{\bar{\mathbf{J}}}}(\tilde{\mathbf{V}}) : \tilde{\bar{\bar{\mathbf{J}}}}(\tilde{\mathbf{V}})} \tag{1.68}$$

This last equation can be rewritten in several other forms

$$\varepsilon_k = \overline{\tilde{\bar{\bar{\mathbf{S}}}}(\tilde{\mathbf{V}}) : \tilde{\bar{\bar{\mathbf{S}}}}(\tilde{\mathbf{V}})} + \overline{\tilde{\bar{\bar{\Omega}}}(\tilde{\mathbf{V}}) : \tilde{\bar{\bar{\Omega}}}(\tilde{\mathbf{V}})} \tag{1.69}$$

$$\varepsilon_k = \overline{\tilde{\bar{\bar{\mathbf{S}}}}(\tilde{\mathbf{V}}) : \tilde{\bar{\bar{\mathbf{S}}}}(\tilde{\mathbf{V}})} + \frac{1}{2}\overline{\left(\nabla \times \tilde{\mathbf{V}}\right) \cdot \left(\nabla \times \tilde{\mathbf{V}}\right)} \tag{1.70}$$

$$\varepsilon_k = 2\overline{\tilde{\bar{\bar{\mathbf{S}}}}(\tilde{\mathbf{V}}) : \tilde{\bar{\bar{\mathbf{S}}}}(\tilde{\mathbf{V}})} - \overline{\tilde{\bar{\bar{\mathbf{J}}}}(\tilde{\mathbf{V}}) : \nabla\tilde{\mathbf{V}}} \tag{1.71}$$

$$\varepsilon_k = 2\overline{\tilde{\bar{\bar{\mathbf{S}}}}(\tilde{\mathbf{V}}) : \tilde{\bar{\bar{\mathbf{S}}}}(\tilde{\mathbf{V}})} - \nabla \cdot \left[\overline{\tilde{\bar{\bar{\mathbf{J}}}}(\tilde{\mathbf{V}}) \cdot \tilde{\mathbf{V}}}\right] + \overline{\tilde{\mathbf{V}} \cdot \left[\nabla \cdot \tilde{\bar{\bar{\mathbf{J}}}}(\tilde{\mathbf{V}})\right]} \tag{1.72}$$

$$\varepsilon_k = 2\overline{\tilde{\bar{\bar{\mathbf{S}}}}(\tilde{\mathbf{V}}) : \tilde{\bar{\bar{\mathbf{S}}}}(\tilde{\mathbf{V}})} - \nabla \cdot \left[\overline{\left(\tilde{\mathbf{V}} \cdot \nabla\right)\tilde{\mathbf{V}}}\right] + \overline{\tilde{\mathbf{V}} \cdot \left[\nabla\left(\nabla \cdot \tilde{\mathbf{V}}\right)\right]} \tag{1.73}$$

where $\tilde{\bar{\bar{\Omega}}}(\tilde{\mathbf{V}})$ is the rotation tensor. Using the continuity equation for incompressible flows, the last terms of Eq. (1.72) is zero. The relations for ε_k, which are given in Eqs. (1.68)–(1.73), involve three tensors that are important in many aspect of turbulence modeling. These are the strain-rate tensor, the Jacobian tensor and the rotation tensor. These three tensors are related through the important mathematical identity

$$\tilde{\bar{\bar{\mathbf{J}}}}(\mathbf{V}) = \tilde{\bar{\bar{\mathbf{S}}}}(\mathbf{V}) + \tilde{\bar{\bar{\Omega}}}(\mathbf{V}) \tag{1.74}$$

Equation (1.64) can be expanded in Cartesian coordinates

$$\chi_k = \frac{1}{2}\,\mathrm{trace}\left(\tilde{\bar{\bar{\chi}}}(\tilde{\mathbf{V}})\right) = \frac{1}{2}\nabla \cdot \left[\left(\tilde{V}_x{}^2 + \tilde{V}_y{}^2 + \tilde{V}_z{}^2\right)\tilde{\mathbf{V}}\right] = \frac{1}{2}\nabla \cdot \overline{\tilde{V}^{\,2}\tilde{\mathbf{V}}} \tag{1.75}$$

Equation (1.65) can be rewritten as

$$\pi_k = \frac{1}{\rho}\left[\nabla\cdot\overline{\tilde{\tilde{p}}\tilde{\tilde{\mathbf{V}}}} - \overline{\tilde{\tilde{p}}\left(\nabla\cdot\tilde{\mathbf{V}}\right)} + \frac{1}{2}\overline{\tilde{V}^2\tilde{\mathbf{V}}}\cdot\nabla\rho\right] \tag{1.76}$$

Using Eqs. (1.66) for γ_k, Eq. (1.68) for ε_k, Eq. (1.75) for χ_k, Eq. (1.76) for π_k and the assumption of constant viscosity in Eq. (1.59), which represents the turbulent-kinetic-energy transport equation for a Newtonian fluid yields

$$\begin{aligned}\frac{\partial k}{\partial t} + \left(\overline{\mathbf{V}}\cdot\nabla\right)k = {} & \frac{\bar{\bar{\boldsymbol{\tau}}}}{\rho} : \bar{\bar{\mathbf{J}}}(\overline{\mathbf{V}}) - \nu\overline{\bar{\bar{\mathbf{J}}}(\tilde{\mathbf{V}}) : \bar{\bar{\mathbf{J}}}(\tilde{\mathbf{V}})} + \frac{1}{\rho}\overline{\tilde{p}\left(\nabla\cdot\tilde{\mathbf{V}}\right)} - \frac{1}{3}\nu\overline{\left(\nabla\cdot\tilde{\mathbf{V}}\right)^2} \\ & + \nu\nabla^2 k - \frac{1}{\rho}\nabla\cdot\left(\frac{1}{2}\rho\overline{\tilde{V}^2\tilde{\mathbf{V}}} + \overline{\tilde{p}\tilde{\mathbf{V}}} - \frac{1}{3}\overline{\left(\mu\nabla\cdot\tilde{\mathbf{V}}\right)\tilde{\mathbf{V}}}\right)\end{aligned} \tag{1.77}$$

The left-hand side of Eq. (1.77) represents the mean substantial derivative of k, which is the time rate of change of the specific turbulent kinetic energy for a fluid element as it moves with the mean flow. The term $(1/\rho)\bar{\bar{\boldsymbol{\tau}}} : \bar{\bar{\mathbf{J}}}(\overline{\mathbf{V}})$ represents the production because it is the rate at which specific kinetic energy is transferred from the fluid flow to the turbulent fluctuations. The term $-\nu\overline{\bar{\bar{\mathbf{J}}}(\tilde{\mathbf{V}}) : \bar{\bar{\mathbf{J}}}(\tilde{\mathbf{V}})}$ is commonly referred to as the dissipation per unit mass, because it is usually approximated as being the rate at which the specific kinetic energy is converted to thermal energy through viscous dissipation. The term $(1/\rho)\overline{\tilde{p}\left(\nabla\cdot\tilde{\mathbf{V}}\right)} - (1/3)\nu\overline{\left(\nabla\cdot\tilde{\mathbf{V}}\right)^2}$ is called dilatation because it accounts for interchange between turbulent kinetic energy and thermal energy resulting from fluid expansion or compression. The term $\nu\nabla^2 k$ arises from molecular diffusion, which is the transport of specific turbulent kinetic energy resulting from the molecular motions within the fluid. The term $-(1/\rho)\nabla\cdot\left((1/2)\rho\overline{\tilde{V}^2\tilde{\mathbf{V}}} + \overline{\tilde{p}\tilde{\mathbf{V}}} - (1/3)\overline{(\mu\nabla\cdot\tilde{\mathbf{V}})\tilde{\mathbf{V}}}\right)$ is usually called the turbulent transport term because it includes the transport of specific turbulent kinetic energy that results from the turbulent fluctuations.

Equation (1.77) provides an additional differential equation for the unknown specific turbulent kinetic energy. However, as was the case with the Reynolds-stress-transport equation, Eq. (1.77) also introduces additional unknowns associated with the turbulent fluctuations. Thus, turbulence models that use Eq. (1.77) must also include closure approximation for those unknown terms. The production term

$(1/\rho)\bar{\bar{\boldsymbol{\tau}}} : \bar{\bar{\mathbf{J}}}(\overline{\mathbf{V}})$ in Eq. (1.77) can be expressed in terms of the eddy viscosity, the mean velocity and specific turbulent kinetic energy by using the Boussinesq hypothesis given in Eq. (1.38), which yields

$$\frac{1}{\rho}\bar{\bar{\boldsymbol{\tau}}} : \bar{\bar{\mathbf{J}}}(\overline{\mathbf{V}}) = 2\frac{\mu_t}{\rho}\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) : \bar{\bar{\mathbf{J}}}(\overline{\mathbf{V}}) - \frac{2}{3}\left(k + \frac{\mu_t}{\rho}\nabla\cdot\overline{\mathbf{V}}\right)\bar{\bar{\boldsymbol{\delta}}} : \bar{\bar{\mathbf{J}}}(\overline{\mathbf{V}}) \tag{1.78}$$

The kinematic eddy viscosity ν_t is defined as

$$\nu_t = \frac{\mu_t}{\rho} \tag{1.79}$$

Using the mathematical identity (A.5) from Appendix A and the kinematic eddy viscosity ν_t given in Eq. (1.77), we can rewrite the production term given in Eq. (1.78) as

$$\begin{aligned}\frac{1}{\rho}\bar{\bar{\boldsymbol{\tau}}} : \bar{\bar{\mathbf{J}}}(\overline{\mathbf{V}}) &= 2\nu_t\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) : \bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) - \frac{2}{3}\left(k + \nu_t\nabla\cdot\overline{\mathbf{V}}\right)\bar{\bar{\boldsymbol{\delta}}} : \bar{\bar{\mathbf{J}}}(\overline{\mathbf{V}}) \\ &= 2\nu_t\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) : \bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) - \frac{2}{3}\left(k + \nu_t\nabla\cdot\overline{\mathbf{V}}\right)\nabla\cdot\overline{\mathbf{V}}\end{aligned} \tag{1.80}$$

The term $\nabla\cdot\overline{\mathbf{V}}$ is identically 0 for incompressible flow and is sometimes neglected even for compressible flow. The approximate dissipation term $-\nu\overline{\bar{\bar{\mathbf{J}}}(\widetilde{\mathbf{V}}) : \bar{\bar{\mathbf{J}}}(\widetilde{\mathbf{V}})}$ in Eq. (1.77) has been modeled in many different ways depending on the turbulence model selected. In fact, the primary difference between the turbulent energy equation models in common use today is the manner in which the second term on the right-hand side of Eq. (1.77) is determined. This term is still an unknown function of the turbulent velocity fluctuations and is typically denoted as ε. Using Eq.(1.68), the approximate dissipation term ε can be written as

$$\varepsilon = \nu\overline{\bar{\bar{\mathbf{J}}}(\widetilde{\mathbf{V}}) : \bar{\bar{\mathbf{J}}}(\widetilde{\mathbf{V}})} = \nu\varepsilon_k \tag{1.81}$$

The dilatation term $(1/\rho)\overline{\widetilde{p}\left(\nabla\cdot\widetilde{\mathbf{V}}\right)} - (1/3)\nu\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2}$ of Eq. (1.77) is zero for incompressible flow, and even for compressible flow they are commonly assumed to be negligible for the case of flows with high supersonic

mean Mach numbers [41]. Accordingly, the third and fourth terms on the right-hand side of Eq. (1.77) are usually neglected for turbulence models in common use today.

$$\frac{1}{\rho}\overline{\tilde{p}\left(\nabla\cdot\tilde{\mathbf{V}}\right)}-\frac{1}{3}\nu\overline{\left(\nabla\cdot\tilde{\mathbf{V}}\right)^2}=0 \tag{1.82}$$

The turbulent transport term $-(1/\rho)\nabla\cdot\left((1/2)\rho\overline{\tilde{V}^{\,2}\tilde{\mathbf{V}}}+\overline{\tilde{p}\tilde{\mathbf{V}}}-(1/3)\overline{\left(\mu\nabla\cdot\tilde{\mathbf{V}}\right)\tilde{\mathbf{V}}}\right)$ in Eq. (1.75) is typically combined and modeled as a pure gradient-diffusion process, which is analogous to the molecular diffusion term in Eq. (1.77). The traditional modeled version of the turbulent-kinetic-energy-transport equation is

$$\begin{aligned}\frac{\partial k}{\partial t}+\left(\overline{\mathbf{V}}\cdot\nabla\right)k &= 2\nu_t\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}):\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})-\frac{2}{3}\left(k+\nu_t\nabla\cdot\overline{\mathbf{V}}\right)\nabla\cdot\overline{\mathbf{V}}-\varepsilon\\ &\quad+\nu\nabla^2k+\frac{1}{\rho}\nabla\cdot\left(\frac{\mu_t}{\sigma_k}\nabla k\right)\end{aligned} \tag{1.83}$$

Equation (1.83) is the turbulence-energy-transport equation that is used with most turbulent-kinetic-energy models in common use today [13, 17, 18, 19]. When this last equation is combined with the mean continuity equation and the three components of the Boussinesq approximation for the Reynolds-averaged Navier-Stokes equations, those five scalar differential equations involve several unknowns. For the case of incompressible flow with density and molecular viscosity known, these five transport equations involve seven unknown scalar variables; the three components of the mean velocity vector, the mean pressure and the three turbulence variables (eddy viscosity, turbulence kinetic energy and dissipation). To close this formulation, two additional independent equations relating these seven variables are required. The additional required equations could be either algebraic relations or differential transport equations. For one-equation turbulence models, both of the additional relations are algebraic. Most two-equation turbulence models provide one additional transport equation together with an algebraic equation relating the eddy viscosity to the other turbulence variables.

Assuming a constant molecular viscosity, as it was the case to generate the Reynolds-stress-transport equation, the traditional modeled version of the turbulent-kinetic-energy transport equation expressed in Eq. (1.83) can be rearranged as

$$\frac{\partial k}{\partial t}+\left(\overline{\mathbf{V}}\cdot\nabla\right)k=2\nu_t\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}):\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})-\frac{2}{3}\left(k+\nu_t\nabla\cdot\overline{\mathbf{V}}\right)\nabla\cdot\overline{\mathbf{V}}-\varepsilon+\nabla\cdot\left[\left(\nu+\frac{\nu_t}{\sigma_k}\right)\nabla k\right] \tag{1.84}$$

Equation (1.84) is the traditional modeled version of the turbulent-kinetic-energy transport equation used in most of the turbulence models, such as Jones and Launder [30], Wilcox [16] and Menter [15]. It assumes the fluid is Newtonian with constant density and constant viscosity. The unknowns are the three components of the mean velocity vector, the mean pressure, and the three turbulence variables (eddy viscosity, turbulent kinetic energy and turbulent dissipation rate).

One of the main concerns in the development of the turbulent kinetic energy equation is the assumption of constant viscosity. In the derivation of the *k*-*ε* model, it was assumed that the flow is fully turbulent, and the effects of molecular viscosity are negligible. The traditional *k*-*ε* model is therefore valid only for fully turbulent flows. Phillips [24] suggested an alternative equation for the turbulent-kinetic-energy transport equation that does not assume constant viscosity. The turbulent-energy dissipation per unit mass used in the development of the traditional turbulent energy transport equation is not equal to the true dissipation of turbulent kinetic energy per unit mass as Wilcox [14] mentioned. Phillips [24] showed that a part of the molecular transport was neglected in the development of the traditional turbulent-kinetic-energy transport equation. This resulted in neglecting a portion of the turbulent transport term when applying Boussinesq's analogy between molecular and turbulent transport. Phillips [24] proposed an alternate model for the turbulent transport energy equation that does not neglect part of the molecular transport term and does not assume a constant viscosity at any place. Phillips's turbulence kinetic energy transport equation is based on a rigorous application of the Boussinesq approximation.

D. Traditional Energy-Dissipation Turbulence Models

Most turbulence models use an algebraic relation for the eddy viscosity, obtained from dimensional analysis. By using the Boussinesq hypothesis, the Reynolds stress tensor was expressed in terms of the mean velocity vector and two additional scalar variables, the turbulent eddy viscosity and the specific turbulent kinetic energy. However, both ν_t and *k* are unknowns functions of space and time. The turbulent-energy-transport equation provides one additional equation. However, this transport equation also introduces at least one additional unknown. In the forms of the equation considered to this point, the

additional unknown introduced by the turbulent-energy-transport equation is one of the following: the approximate dissipation, the exact dissipation or the RMS fluctuating vorticity. To close the formulation from this point, one of these unknown scalar functions has to be related to the other turbulence variables and the mean flow. This is accomplished by observing the length and velocity scales of turbulence.

The algebraic relation for the eddy viscosity is based on dimensional analysis first suggested by Prandtl [5, 10]. From the kinetic theory of gases, the molecular kinematic viscosity is found to be proportional to the product of molecular mean free path and the square root of the total specific molecular kinetic energy. In a similar manner, Prandtl [5] hypothesized that the kinematic eddy viscosity was proportional to the product of a characteristic length ℓ, called the mixing length, and some suitable characteristic velocity. In a later development, Prandtl [10] proposed a more direct analogy between turbulence and kinetic theory, assuming that the kinematic eddy viscosity was proportional to the product of a turbulence mixing length ℓ and the square root of the total specific kinetic energy associated with the fluctuating velocity field.

$$\nu_t \propto \ell\sqrt{k} \tag{1.85}$$

The turbulence energy give information about the strength of the turbulent eddies. However it says nothing about their size. Predicting turbulent transport requires the knowledge of the characteristic length associated with the turbulent fluctuations as well as their energy. Turbulence consists of the superposition of various sized eddies, all having kinetic energy determined by the intensity of their velocity fluctuations. The distribution of kinetic energy among turbulent eddies of various size is called the turbulence energy spectrum. Taylor [42] was the first to present a mathematical relationship between the energy spectrum and the velocity fluctuations. An extensive discussion of their energy spectrum was presented by Hinze [43]. The characteristic length associated with the turbulent transport should be some kind of weighted average of the characteristic lengths for all eddies that make up the energy spectrum.

The kinetic energy results from superimposed periodic motions of various frequencies. Such a motion can be characterized in two different ways, either in terms of a characteristic velocity V_c and a

characteristic frequency ω_c, or in terms of a characteristic velocity and a characteristic wavelength. A wavelength for turbulent motion is commonly called a length scale and is usually denoted by ℓ.

$$\ell \equiv \frac{V_c}{\omega_c} \tag{1.86}$$

The exact physical interpretation of the characteristic angular velocity has been a matter of some controversy. The characteristic angular velocity is often taken to be the proportional to the approximate dissipation divided by the specific kinetic energy.

$$\omega \propto \frac{\varepsilon}{k} \tag{1.87}$$

This definition links the characteristic length of the turbulent flow field to the characteristic length of turbulent dissipation, not to the characteristic length of the eddies in which the most energy is found. Substituting Eqs. (1.166) and (1.86) in Eq. (1.85) yield an expression for the turbulent eddy viscosity

$$\nu_t = C_\mu \frac{k^2}{\varepsilon} \tag{1.88}$$

where C_μ is a closure coefficient. This relationship provides one algebraic relation for the *k*-*ε* turbulence model.

The modeled transport equation for the turbulent dissipation *ε* is usually developed by analogy with Eq. (1.84) and can be written for incompressible flows as

$$\frac{\partial \varepsilon}{\partial t} + \left(\overline{\mathbf{V}} \cdot \nabla\right)\varepsilon = 2C_{\varepsilon 1}\nu_t \frac{\varepsilon}{k}\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}) : \overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}) - C_{\varepsilon 2}\frac{\varepsilon^2}{k} + \nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\nabla\varepsilon\right] \tag{1.89}$$

where $C_{\varepsilon 1}$, $C_{\varepsilon 2}$ and σ_ε are closure constants. Note that the turbulent dissipation equation is traditionally constructed by dimensional analysis and is not developed rigorously from the Navier-Stokes equations. Wilcox [21] makes mention of this concern and gives what he terms an “exact” equation for the turbulent dissipation by taking a moment of the Navier-Stokes equations using the definition of the turbulent

dissipation. However, the resulting equation is extremely complicated and Wilcox submits that closure coefficients for the resulting differential equation are all but impossible to measure at this point. Thus the modeled version of the dissipation transport equation is traditionally used for lack of a useful version of a more rigorously derived equation.

The traditional k-ε model is a semi-empirical model based on transport equations for the turbulence kinetic energy and its dissipation rate. The model transport equation for k is derived from the Navier-Stokes equations, while the model transport equation for ε was obtained using physical reasoning and bears little resemblance to its counterpart. In the derivation of the model, it was assumed that the flow is fully-turbulent and the effects of molecular viscosity are negligible. The traditional k-ε model is given in Eqs. (1.84), (1.88) and (1.89).

The k-ϵ model can be reparametrized with a simple change of variable to eliminate the so-called dissipation per unit mass, ϵ, in favor of the dissipation frequency, ω, which is defined by Eq. (1.87). From this definition, ϵ is proportional to the k-ω product and choosing the proportionality constant to be C_μ yields the change of variables

$$\omega \equiv \frac{\varepsilon}{C_\mu k} \tag{1.90}$$

Applying Eq. (1.90) to Eqs. (1.84), (1.88) and (1.89) produces the k-ω turbulence model for incompressible flow. The algebraic equation for the kinematic eddy viscosity is given by

$$\nu_t = \frac{k}{\omega} \tag{1.91}$$

The turbulent-energy-transport equation is given by

$$\frac{\partial k}{\partial t} + \left(\overline{\mathbf{V}} \cdot \nabla\right) k = 2\nu_t \bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) : \bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) - C_\mu k\omega + \nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\nabla k\right] \tag{1.92}$$

The dissipation-frequency-transport equation is obtained by analogy with the transport equation for the turbulent kinetic energy

$$\frac{\partial \omega}{\partial t} + \left(\overline{\mathbf{V}} \cdot \nabla\right)\omega = 2C_{\omega 1}\frac{\omega}{k}\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) : \bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) - C_{\omega 2}\omega^2 + \nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_\omega}\right)\nabla\omega\right] \tag{1.93}$$

where C_μ, $C_{\omega 1}$, $C_{\omega 2}$, σ_k and σ_ω are closure coefficients.

V. Phillips' Turbulent-Kinetic-Energy Equation

A. Main Concerns with the Traditional Turbulence Models

At this point, several concerns with the traditional k-ε or k-ω turbulence models can be identified. The dissipation length scale is that associated to the smaller turbulent eddies having the highest strain rates per unit kinetic energy, However, the larger turbulent eddies carry more energy and are primarily responsible for the transport of momentum energy in a fluid. The transport equations for the second turbulence variables were obtained simply from dimensional analysis and analogy with the turbulent-energy-transport equation. They were not developed in a rigorous manner from the Navier-Stokes equations. The so-called turbulent-energy dissipation per unit mass is not equal to the true dissipation of turbulent kinetic energy per unit mass. Because the approximate turbulent-energy dissipation per unit mass that is used in the traditional k-ε or k-ω turbulence models includes a portion of the total molecular transport, the so-called molecular transport terms do not include the total molecular transport of turbulent kinetic energy per unit mass. Because a part of the molecular transport was neglected, subsequent application of Boussinesq's analog between molecular and turbulent transport also results in neglecting a portion of the turbulent transport of turbulent kinetic energy per unit mass. Using the dissipation length scale to define the eddy viscosity predicts a Reynolds stress tensor that is inversely proportional to the molecular viscosity, whereas the Reynolds stress tensor should not depend directly on molecular viscosity.

The dissipation rate transport equation remains the most uncertain part of turbulence modeling. The transport equation for the second turbulence variable is traditionally constructed by dimensional analysis and analogy with the turbulent-energy transport equation. It is not developed in a rigorous manner from the Navier-Stokes equations. This transport equation has no basis in physics. Wilcox [14] mentions this concern and gives what he terms an exact equation for the turbulent dissipation by taking a moment of the Navier-Stokes equation using the definition of the turbulent dissipation. However, the resulting equation is

extremely complicated and Wilcox submits that closure coefficients for the resulting differential equation are all but impossible to measure at this point. Thus a simplified dissipation equation modeled based on the transport equation of the turbulent-kinetic energy is traditionally used for lack of a useful version of a more rigorously derived equation.

In the original development of the *k*-ε turbulence model and in many subsequent presentations of the turbulent-energy-transport equation, the parameter ε is presented as being exactly the turbulent energy dissipation per unit mass. With this misinterpretation, the molecular diffusion term $\nu\nabla^2 k$ is commonly presented as being the total molecular transport of turbulent energy per unit mass. Although it is now generally recognized that ε is not precisely the turbulent-energy dissipation per unit mass, its continued use is typically justified on the grounds that the additional terms are small compared with the turbulent transport terms. In fact for most turbulent flows, all molecular transport can be neglected in comparison with the turbulent transport. The most significant concern with the traditional *k*-ε or *k*-ω turbulence models is not the lack of precision in defining the dissipation or the molecular transport. A more significant concern is that associated with the application of the Boussinesq analogy between molecular and turbulent transport to a molecular transport term that has been less than rigorously developed.

The traditional *k*-ε or *k*-ω turbulence models are all based on approximating the turbulent transport of turbulent kinetic energy as pure gradient diffusion. For the case of incompressible flow, these models all assume a turbulent kinetic energy flux given by $(\nu_t / \sigma_k)\nabla k$. A more rigorous development suggests that the improved results for incompressible flow might be obtained by using a turbulent kinetic energy flux that is specified by $(\nu_t / \sigma_k)\left\{(5/3)\nabla k - 2\nabla\cdot\left[\nu_t \bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})\right]\right\}$. This is based on a more direct analogy between turbulent and molecular transport.

Perhaps the greatest concern with traditional turbulence models is that they all fail to exhibit proper dependence on molecular viscosity. From the definition of the Reynolds stress tensor, we see that the Reynolds stresses depend only on the fluid density and turbulent velocity fluctuations. They are independent of the other natural fluid properties such as the molecular viscosity. Hence if the Boussinesq analogy between turbulent and molecular transport is strictly followed, the dynamic eddy viscosity should be related to only the fluid density and turbulent velocity fluctuations. The eddy viscosity should not depend directly on molecular viscosity.

B. Phillips Turbulent-kinetic-energy transport equation

Phillips's turbulent-kinetic-energy transport equation [24] does not originate from the trace of the Reynolds stress transport equation as the traditional model does. Phillips bases his transport equation on the direct application of the Navier-Stokes and RANS equations. Phillips uses a rigorous application of the Boussinesq's hypothesis and defines an alternate algebraic relation for the turbulent eddy viscosity.

1. Dot Product of the Navier-Stokes Equations with the Velocity Vector

The turbulent-kinetic-energy transport equation can be developed directly from the mechanical energy equation, which is obtained by taking the dot product of the fluid velocity vector with the Navier-Stokes equations. The Navier Stokes equations expanded using the geopotential altitude for the gravity force as given in Eq. (1.8) is

$$\rho\left[\frac{\partial \mathbf{V}}{\partial t}+(\mathbf{V}\cdot\nabla)\mathbf{V}\right]=\nabla\cdot\left[2\mu\,\bar{\bar{\mathbf{S}}}(\mathbf{V})\right]-\nabla\hat{p}+g_0Z\nabla(\rho) \tag{1.94}$$

where $\hat{p}$ represents the total hydrostatic pressure, and was define in Eq. (1.7). Taking the dot product of Eq. (1.94) with the fluid velocity vector yields

$$\rho\mathbf{V}\cdot\left[\frac{\partial \mathbf{V}}{\partial t}+(\mathbf{V}\cdot\nabla)\mathbf{V}\right]=\mathbf{V}\cdot\left[\nabla\cdot\left[2\mu\bar{\bar{\mathbf{S}}}(\mathbf{V})\right]-\nabla\hat{p}+g_0Z\nabla(\rho)\right] \tag{1.95}$$

Using the mathematical identity (A.4) from Appendix A in the left-hand side of Eq. (1.95) yields

$$\rho\left[\frac{\partial \frac{1}{2}V^2}{\partial t}+\mathbf{V}\cdot\nabla\left(\tfrac{1}{2}V^2\right)\right]=\mathbf{V}\cdot\left[\nabla\cdot\left[2\mu\bar{\bar{\mathbf{S}}}(\mathbf{V})\right]-\nabla\hat{p}+g_0Z\nabla(\rho)\right] \tag{1.96}$$

The first term of Eq. (1.96) can be expanded as given in the mathematical identity proven in (A.7)

$$\mathbf{V}\cdot\left[\nabla\cdot\left(2\mu\bar{\bar{\mathbf{S}}}(\mathbf{V})\right)\right]=\nabla\cdot\left(\mu\left[\nabla\left(\tfrac{1}{2}V^2\right)+(\mathbf{V}\cdot\nabla)\mathbf{V}\right]\right)-2\mu\bar{\bar{\mathbf{S}}}(\mathbf{V}):\bar{\bar{\mathbf{S}}}(\mathbf{V}) \tag{1.97}$$

Hence, the dot product of the Navier-Stokes equations with the fluid velocity vector, which represents the mechanical energy equation for a Newtonian fluid is

$$\rho\left[\frac{\partial \frac{1}{2}V^2}{\partial t}+\mathbf{V}\cdot\nabla\left(\tfrac{1}{2}V^2\right)\right]=\nabla\cdot\left(\mu\left[\nabla\left(\tfrac{1}{2}V^2\right)+(\mathbf{V}\cdot\nabla)\mathbf{V}\right]\right)$$
$$-\mathbf{V}\cdot\left[\nabla\hat{p}-g_0Z\nabla(\rho)\right]-2\mu\bar{\bar{\mathbf{S}}}(\mathbf{V}):\bar{\bar{\mathbf{S}}}(\mathbf{V}) \tag{1.98}$$

The term $\nabla\cdot\left(\mu\left[\nabla\left(\frac{1}{2}V^2\right)+(\mathbf{V}\cdot\nabla)\mathbf{V}\right]\right)$ in Eq. (1.98) accounts for the molecular transport of mechanical energy. The term $-\mathbf{V}\cdot\left[\nabla\hat{p}-g_0Z\nabla(\rho)\right]$ in Eq. (1.98) describes the rate at which the mechanical energy is transported from one point to another within the fluid as a result of molecular motion. The term $-2\mu\bar{\bar{\mathbf{S}}}(\mathbf{V}):\bar{\bar{\mathbf{S}}}(\mathbf{V})$ in Eq. (1.98) is the viscous dissipation of mechanical energy per unit volume. It represents the volumetric rate at which mechanical energy is converted into thermal energy through the process of viscous dissipation. Using the velocity Reynolds decomposition given in Eq. (1.17) yields

$$\begin{aligned}V^2=\mathbf{V}\cdot\mathbf{V}&=\left(\overline{\mathbf{V}}+\widetilde{\mathbf{V}}\right)\cdot\left(\overline{\mathbf{V}}+\widetilde{\mathbf{V}}\right)\\&=\overline{\mathbf{V}}\cdot\overline{\mathbf{V}}+2\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}}+\widetilde{\mathbf{V}}\cdot\widetilde{\mathbf{V}}\\&=\overline{\mathbf{V}}^2+2\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}}+\widetilde{\mathbf{V}}^2\end{aligned} \tag{1.99}$$

Using Eq. (1.99) and (1.17)–(1.18) in Eq. (1.98) gives

$$\begin{aligned}&\rho\left[\frac{\partial}{\partial t}\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}}+\tfrac{1}{2}\widetilde{\mathbf{V}}^2\right)+\overline{\mathbf{V}}\cdot\nabla\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}}+\tfrac{1}{2}\widetilde{\mathbf{V}}^2\right)+\widetilde{\mathbf{V}}\cdot\nabla\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}}+\tfrac{1}{2}\widetilde{\mathbf{V}}^2\right)\right]\\&=\nabla\cdot\left(\mu\left[\nabla\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}}+\tfrac{1}{2}\widetilde{\mathbf{V}}^2\right)+(\overline{\mathbf{V}}\cdot\nabla)\overline{\mathbf{V}}+(\widetilde{\mathbf{V}}\cdot\nabla)\overline{\mathbf{V}}+(\overline{\mathbf{V}}\cdot\nabla)\widetilde{\mathbf{V}}+(\widetilde{\mathbf{V}}\cdot\nabla)\widetilde{\mathbf{V}}\right]\right)\\&\quad-\overline{\mathbf{V}}\cdot\left[\nabla\bar{\hat{p}}+\nabla\tilde{\hat{p}}-g_0Z\nabla(\rho)\right]-\widetilde{\mathbf{V}}\cdot\left[\nabla\bar{\hat{p}}+\nabla\tilde{\hat{p}}-g_0Z\nabla(\rho)\right]\\&\quad-2\mu\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}):\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})-4\mu\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}):\bar{\bar{\mathbf{S}}}(\widetilde{\mathbf{V}})-2\mu\bar{\bar{\mathbf{S}}}(\widetilde{\mathbf{V}}):\bar{\bar{\mathbf{S}}}(\widetilde{\mathbf{V}})\end{aligned} \tag{1.100}$$

Taking the ensemble average of the right-hand side of Eq. (1.100) yields

$$\begin{aligned}&\rho\left[\overline{\frac{\partial}{\partial t}\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}}+\tfrac{1}{2}\widetilde{\mathbf{V}}^2\right)+\overline{\mathbf{V}}\cdot\nabla\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}}+\tfrac{1}{2}\widetilde{\mathbf{V}}^2\right)+\widetilde{\mathbf{V}}\cdot\nabla\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}}+\tfrac{1}{2}\widetilde{\mathbf{V}}^2\right)}\right]\\&=\rho\left[\frac{\partial}{\partial t}\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+\tfrac{1}{2}\overline{\widetilde{\mathbf{V}}^2}\right)+\overline{\mathbf{V}}\cdot\nabla\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+\tfrac{1}{2}\overline{\widetilde{\mathbf{V}}^2}\right)+\overline{\widetilde{\mathbf{V}}\cdot\nabla\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}}+\tfrac{1}{2}\widetilde{\mathbf{V}}^2\right)}\right]\\&=\rho\left[\frac{\partial}{\partial t}\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+k\right)+\overline{\mathbf{V}}\cdot\nabla\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+k\right)+\overline{\widetilde{\mathbf{V}}\cdot\nabla\left(\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}}\right)}+\overline{\widetilde{\mathbf{V}}\cdot\nabla\left(\tfrac{1}{2}\widetilde{\mathbf{V}}^2\right)}\right]\end{aligned} \tag{1.101}$$

where k is the turbulent kinetic energy defined in Eq. (1.39). Taking the ensemble average of the first term on the left-hand side of Eq. (1.100) gives

$$\begin{aligned}&\overline{\nabla\cdot\left(\mu\left[\nabla\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}}+\tfrac{1}{2}\widetilde{\mathbf{V}}^2\right)+\left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\mathbf{V}}+\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\overline{\mathbf{V}}+\left(\overline{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}+\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}\right]\right)}\\&=\nabla\cdot\left(\mu\left[\nabla\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+k\right)+\left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\mathbf{V}}+\overline{\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}}\right]\right)\end{aligned}\tag{1.102}$$

Taking the ensemble average of the last two terms on the left-hand side of Eq. (1.89) gives

$$\begin{aligned}&\overline{-\overline{\mathbf{V}}\cdot\left[\nabla\overline{\hat{p}}+\nabla\widetilde{\hat{p}}-g_0Z\nabla(\rho)\right]-\widetilde{\mathbf{V}}\cdot\left[\nabla\overline{\hat{p}}+\nabla\widetilde{\hat{p}}-g_0Z\nabla(\rho)\right]}\\&\overline{-2\mu\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}})-4\mu\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}})-2\mu\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}})}}\\&=-\overline{\mathbf{V}}\cdot\left[\nabla\overline{\hat{p}}-g_0Z\nabla(\rho)\right]-\overline{\widetilde{\mathbf{V}}\cdot\nabla\widetilde{\hat{p}}}-2\mu\left(\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}})+\overline{\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}})}\right)\end{aligned}\tag{1.103}$$

where $\overline{\hat{p}}$ represents the mean value of the hydrostatic pressure $\hat{p}$ defined in Eq. (1.7).

$$\overline{\hat{p}}=\overline{p}+g_0\rho Z+\frac{2}{3}\mu\nabla\cdot\overline{\mathbf{V}}\tag{1.104}$$

Similarly $\widetilde{\hat{p}}$ represents the fluctuating component of the hydrostatic pressure $\hat{p}$.

$$\widetilde{\hat{p}}=\widetilde{p}+\frac{2}{3}\mu\nabla\cdot\widetilde{\mathbf{V}}\tag{1.105}$$

Substituting Eqs.(1.101)–(1.102) in Eq. (1.100) gives the mechanical energy equation

$$\begin{aligned}&\rho\left[\frac{\partial}{\partial t}\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+k\right)+\overline{\mathbf{V}}\cdot\nabla\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+k\right)+\overline{\widetilde{\mathbf{V}}\cdot\nabla\left(\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}}\right)}+\overline{\widetilde{\mathbf{V}}\cdot\nabla\left(\tfrac{1}{2}\widetilde{\mathbf{V}}^2\right)}\right]\\&=\nabla\cdot\left(\mu\left[\nabla\left(\tfrac{1}{2}\overline{\mathbf{V}}^2+k\right)+\left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\mathbf{V}}+\overline{\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}}\right]\right)\\&-\overline{\mathbf{V}}\cdot\left[\nabla\overline{\hat{p}}-g_0Z\nabla(\rho)\right]-\overline{\widetilde{\mathbf{V}}\cdot\nabla\widetilde{\hat{p}}}-2\mu\left(\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}})+\overline{\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}})}\right)\end{aligned}\tag{1.106}$$

2. Dot Product of the RANS Equations with the Mean Velocity Vector

The Reynolds-Averaged Navier-Stokes equations can be written from Eqs. (1.36)–(1.37) using the definition of $\bar{\tilde{p}}$ given in Eq. (1.104) as

$$\rho\left[\frac{\partial\overline{\mathbf{V}}}{\partial t}+\left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\mathbf{V}}+\overline{\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}}\right]=\nabla\cdot\left[2\mu\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})\right]-\nabla\bar{\tilde{p}}+g_0Z\nabla\rho \tag{1.107}$$

Taking the dot product of this equation with the mean velocity vector gives the mean mechanical energy equation

$$\rho\left[\frac{\partial\overline{\mathbf{V}}\cdot\overline{\mathbf{V}}}{\partial t}+\overline{\mathbf{V}}\cdot\left[\left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\mathbf{V}}\right]+\overline{\mathbf{V}}\cdot\overline{\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}}\right]=\overline{\mathbf{V}}\cdot\left(\nabla\cdot\left[2\mu\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})\right]\right)-\overline{\mathbf{V}}\cdot\left(\nabla\bar{\tilde{p}}-g_0Z\nabla\rho\right) \tag{1.108}$$

Using the mathematical identity given in Eq. (A.7) expanded in terms of $\overline{\mathbf{V}}$ in place of $\mathbf{V}$ yields

$$\overline{\mathbf{V}}\cdot\left[\nabla\cdot\left(2\mu\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})\right)\right]=\nabla\cdot\left(\mu\left[\nabla\left(\tfrac{1}{2}\overline{V}^2\right)+\left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\mathbf{V}}\right]\right)-2\mu\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}):\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) \tag{1.109}$$

Using the mathematical identity given in Eq. (1.109) in Eq. (1.108) gives

$$\begin{aligned}&\rho\left[\frac{\partial\overline{\mathbf{V}}\cdot\overline{\mathbf{V}}}{\partial t}+\overline{\mathbf{V}}\cdot\left[\left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\mathbf{V}}\right]+\overline{\mathbf{V}}\cdot\overline{\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}}\right]=\nabla\cdot\left(\mu\left[\nabla\left(\tfrac{1}{2}\overline{V}^2\right)+\left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\mathbf{V}}\right]\right)\\&-2\mu\,\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}):\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})-\overline{\mathbf{V}}\cdot\left(\nabla\bar{\tilde{p}}-g_0Z\nabla\rho\right)\end{aligned} \tag{1.110}$$

Using the mathematical identity (A.4) yields

$$\begin{aligned}&\rho\left[\frac{\partial\,\frac{1}{2}\overline{V}^2}{\partial t}+(\overline{\mathbf{V}}\cdot\nabla)(\tfrac{1}{2}\overline{V}^2)+\overline{\mathbf{V}}\cdot\overline{\left(\widetilde{\mathbf{V}}\cdot\nabla\right)\widetilde{\mathbf{V}}}\right]=\nabla\cdot\left(\mu\left[\nabla\left(\tfrac{1}{2}\overline{V}^2\right)+\left(\overline{\mathbf{V}}\cdot\nabla\right)\overline{\mathbf{V}}\right]\right)\\&-\overline{\mathbf{V}}\cdot\left(\nabla\bar{\tilde{p}}-g_0Z\nabla\rho\right)-2\mu\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}):\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})\end{aligned} \tag{1.111}$$

After applying the mathematical identity (A.8) to Eq. (1.111) yields

$$\rho\left[\frac{\partial \frac{1}{2}\overline{V}^2}{\partial t}+(\overline{\mathbf{V}}\cdot\nabla)(\tfrac{1}{2}\overline{V}^2)+\overline{\mathbf{V}}\cdot\left(\nabla\cdot\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}\right)-\overline{\mathbf{V}}\cdot\left(\overline{\widetilde{\mathbf{V}}(\nabla\cdot\widetilde{\mathbf{V}})}\right)\right]$$
$$=\nabla\cdot\left(\mu\left[\nabla\left(\tfrac{1}{2}\overline{V}^2\right)+(\overline{\mathbf{V}}\cdot\nabla)\overline{\mathbf{V}}\right]\right)-\overline{\mathbf{V}}\cdot\left(\nabla\overline{\hat{p}}-g_0Z\nabla\rho\right)-2\mu\,\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}) \tag{1.112}$$

Taking the ensemble average of the mathematical identity (A.9) yields

$$\overline{\mathbf{V}}\cdot\left[\nabla\cdot\left(\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}\right)\right]=\overline{\widetilde{\mathbf{V}}\cdot\nabla(\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}})}+\overline{(\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}})(\nabla\cdot\widetilde{\mathbf{V}})}-\left(\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}\right):\overline{\overline{\mathbf{J}}}(\overline{\mathbf{V}})$$
$$=\overline{\widetilde{\mathbf{V}}\cdot\nabla(\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}})}+\overline{\mathbf{V}}\cdot\left(\overline{\widetilde{\mathbf{V}}(\nabla\cdot\widetilde{\mathbf{V}})}\right)-\left(\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}\right):\overline{\overline{\mathbf{J}}}(\overline{\mathbf{V}}) \tag{1.113}$$

After applying the Eq. (1.113) to Eq. (1.112) gives

$$\rho\left[\frac{\partial \frac{1}{2}\overline{V}^2}{\partial t}+(\overline{\mathbf{V}}\cdot\nabla)(\tfrac{1}{2}\overline{V}^2)+\overline{\widetilde{\mathbf{V}}\cdot\nabla(\overline{\mathbf{V}}\cdot\widetilde{\mathbf{V}})}-\left(\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}\right):\overline{\overline{\mathbf{J}}}(\overline{\mathbf{V}})\right]$$
$$=\nabla\cdot\left(\mu\left[\nabla\left(\tfrac{1}{2}\overline{V}^2\right)+(\overline{\mathbf{V}}\cdot\nabla)\overline{\mathbf{V}}\right]\right)-\overline{\mathbf{V}}\cdot\left(\nabla\overline{\hat{p}}-g_0Z\nabla\rho\right)-2\mu\,\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}) \tag{1.114}$$

This last equation, Eq. (1.114), represents the mean mechanical energy equation.

3. Turbulent-kinetic-energy transport equation

The turbulent-kinetic-energy transport equation is based on the difference between the ensemble average of the dot product of the Navier-Stokes equations with the velocity vector and the dot product of the RANS equations with the mean velocity vector. Subtracting the mean mechanical energy equation given in Eq. (1.114) from the mechanical energy equation given in Eq. (1.106) yields

$$\rho\left[\frac{\partial k}{\partial t}+\overline{\mathbf{V}}\cdot\nabla k+\left(\overline{\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}}\right):\overline{\overline{\mathbf{J}}}(\overline{\mathbf{V}})+\overline{\widetilde{\mathbf{V}}\cdot\nabla\left(\tfrac{1}{2}\widetilde{\mathbf{V}}^2\right)}\right]$$
$$=\nabla\cdot\left(\mu\left[\nabla k+\overline{(\widetilde{\mathbf{V}}\cdot\nabla)\widetilde{\mathbf{V}}}\right]\right)-\overline{\widetilde{\mathbf{V}}\cdot\nabla\widetilde{\hat{p}}}-2\mu\left(\overline{\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}})}\right) \tag{1.115}$$

Applying the continuity equation for the fluctuations given in Eq. (1.24) to the mathematical identity (A.10) and taking the ensemble average yields

$$\rho\overline{\widetilde{\mathbf{V}}\cdot\nabla\left(\frac{1}{2}\widetilde{\mathbf{V}}^2\right)}=\nabla\cdot\left(\frac{1}{2}\rho\overline{\widetilde{V}^2\widetilde{\mathbf{V}}}\right) \tag{1.116}$$

Taking the ensemble average of the mathematical identity (A.11) yields

$$\overline{\widetilde{\mathbf{V}}\cdot\nabla\widetilde{\hat{p}}}=\nabla\cdot\left(\overline{\widetilde{\hat{p}}\widetilde{\mathbf{V}}}\right)-\overline{\widetilde{\hat{p}}\left(\nabla\cdot\widetilde{\mathbf{V}}\right)} \tag{1.117}$$

Applying the definition of the turbulent stress tensor given in Eq. (1.35) and the turbulent momentum transport term expressed in Eq. (1.36) to Eq. (1.115) gives

$$\begin{aligned}\rho\left[\frac{\partial k}{\partial t}+\overline{\mathbf{V}}\cdot\nabla k\right]&=\overline{\overline{\boldsymbol{\tau}}}:\overline{\overline{\mathbf{J}}}\left(\overline{\mathbf{V}}\right)-2\mu\left(\overline{\overline{\widetilde{\mathbf{S}}}(\widetilde{\mathbf{V}}):\overline{\widetilde{\mathbf{S}}}(\widetilde{\mathbf{V}})}\right)+\overline{\widetilde{\hat{p}}\left(\nabla\cdot\widetilde{\mathbf{V}}\right)}\\&+\nabla\cdot\left(\mu\nabla k-\nu\nabla\cdot\overline{\overline{\boldsymbol{\tau}}}\right)-\nabla\cdot\left(\frac{1}{2}\rho\overline{\widetilde{V}^2\widetilde{\mathbf{V}}}+\overline{\widetilde{\hat{p}}\widetilde{\mathbf{V}}}\right)\end{aligned} \tag{1.118}$$

Applying the definition of the fluctuating hydrostatic pressure given in Eq. (1.105) to Eq. (1.118) yields

$$\begin{aligned}\rho\left[\frac{\partial k}{\partial t}+\overline{\mathbf{V}}\cdot\nabla k\right]&=\overline{\overline{\boldsymbol{\tau}}}:\overline{\overline{\mathbf{J}}}\left(\overline{\mathbf{V}}\right)-2\mu\left(\overline{\overline{\widetilde{\mathbf{S}}}(\widetilde{\mathbf{V}}):\overline{\widetilde{\mathbf{S}}}(\widetilde{\mathbf{V}})}\right)+\overline{\left(\widetilde{p}+\frac{2}{3}\mu\nabla\cdot\widetilde{\mathbf{V}}\right)\left(\nabla\cdot\widetilde{\mathbf{V}}\right)}\\&+\nabla\cdot\left(\mu\nabla k-\nu\nabla\cdot\overline{\overline{\boldsymbol{\tau}}}\right)-\nabla\cdot\left(\frac{1}{2}\rho\overline{\widetilde{V}^2\widetilde{\mathbf{V}}}+\overline{\left(\widetilde{p}+\frac{2}{3}\mu\nabla\cdot\widetilde{\mathbf{V}}\right)\widetilde{\mathbf{V}}}\right)\end{aligned} \tag{1.119}$$

After simplifying Eq. (1.119) yields the turbulent energy transport equation

$$\begin{aligned}\rho\left[\frac{\partial k}{\partial t}+\overline{\mathbf{V}}\cdot\nabla k\right]&=\overline{\overline{\boldsymbol{\tau}}}:\overline{\overline{\mathbf{J}}}\left(\overline{\mathbf{V}}\right)-2\mu\left(\overline{\overline{\widetilde{\mathbf{S}}}(\widetilde{\mathbf{V}}):\overline{\widetilde{\mathbf{S}}}(\widetilde{\mathbf{V}})}-\frac{1}{3}\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2}\right)+\overline{\widetilde{p}\left(\nabla\cdot\widetilde{\mathbf{V}}\right)}\\&+\nabla\cdot\left(\mu\nabla k-\nu\nabla\cdot\overline{\overline{\tau}}\right)-\nabla\cdot\left(\frac{1}{2}\rho\overline{\widetilde{V}^2\widetilde{\mathbf{V}}}+\overline{\widetilde{p}\widetilde{\mathbf{V}}}+\frac{2}{3}\mu\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)\widetilde{\mathbf{V}}}\right)\end{aligned} \tag{1.120}$$

The term $\overline{\overline{\tau}}:\overline{\overline{\mathbf{J}}}\left(\overline{\mathbf{V}}\right)$ represents the production. The term $-2\mu\left(\overline{\overline{\widetilde{\mathbf{S}}}(\widetilde{\mathbf{V}}):\overline{\widetilde{\mathbf{S}}}(\widetilde{\mathbf{V}})}-\frac{1}{3}\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2}\right)$ is the viscous dissipation. The term $\overline{\widetilde{\hat{p}}\left(\nabla\cdot\widetilde{\mathbf{V}}\right)}$ is the pressure dilatation. The term $\nabla\cdot\left(\mu\nabla k-\nu\nabla\cdot\overline{\overline{\tau}}\right)$is the molecular transport of turbulent kinetic energy per unit volume. The term $-\nabla\cdot\left(\frac{1}{2}\rho\overline{\widetilde{V}^2\widetilde{\mathbf{V}}}+\overline{\widetilde{\hat{p}}\widetilde{\mathbf{V}}}\right)+\frac{2}{3}\mu\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)\widetilde{\mathbf{V}}}$

accounts for the volumetric turbulent transport of kinetic energy. The only approximation that was made in the development of Eq. (1.120) is that of a Newtonian fluid.

The molecular transport of turbulent kinetic energy is not a simple gradient diffusion process. The term $\mu \nabla k$ is gradient diffusion. However, the contribution from $\nu \nabla \cdot \bar{\bar{\tau}}$ is not necessarily gradient diffusion. Accordingly, even if we accept the Boussinesq analogy between molecular and turbulent transport, we should not expect turbulent transport of kinetic energy to be a simple gradient diffusion process in general.

4. Boussinesq's Approximation

Applying the Boussinesq analogy between molecular and turbulent transport to the turbulent transport term $\frac{1}{2}\rho\overline{\widetilde{V}^2\widetilde{\mathbf{V}}} + \overline{\widetilde{p}\widetilde{\mathbf{V}}} + \frac{2}{3}\mu\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)\widetilde{\mathbf{V}}}$ in Eq. (1.120) suggests

$$\frac{1}{2}\rho\overline{\widetilde{V}^2\widetilde{\mathbf{V}}} + \overline{\widetilde{p}\widetilde{\mathbf{V}}} + \frac{2}{3}\mu\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)\widetilde{\mathbf{V}}} = \frac{\nu_t}{\sigma_k}\left(\rho\nabla k - \nabla\cdot\bar{\bar{\tau}}\right) \tag{1.121}$$

For this Boussinesq model, the Reynolds stress tensor is given by Eq. (1.38)

$$\nabla\cdot\bar{\bar{\tau}} = 2\nabla\cdot\left[\mu_t\,\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})\right] - \frac{2}{3}\left(\rho k + \mu_t\nabla\cdot\overline{\mathbf{V}}\right) \tag{1.122}$$

Using Eqs. (1.121)–(1.122) the new transport kinetic energy equation given in Eq. (1.120) becomes

$$\begin{aligned}\rho\left[\frac{\partial k}{\partial t} + \overline{\mathbf{V}}\cdot\nabla k\right] &= \bar{\bar{\tau}} : \bar{\bar{\mathbf{J}}}\left(\overline{\mathbf{V}}\right) - 2\mu\left(\overline{\bar{\bar{\mathbf{S}}}(\widetilde{\mathbf{V}}) : \bar{\bar{\mathbf{S}}}(\widetilde{\mathbf{V}})} - \tfrac{1}{3}\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2}\right) + \overline{\widetilde{p}\left(\nabla\cdot\widetilde{\mathbf{V}}\right)} \\ &+ \nabla\cdot\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\left(\rho\nabla k - 2\nabla\cdot\left[\mu_t\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})\right] + \tfrac{2}{3}\left(\rho k + \mu_t\nabla\cdot\overline{\mathbf{V}}\right)\right)\right]\end{aligned} \tag{1.123}$$

The production term in this new model is the same as the production term in the traditional model, which was given in Eq. (1.79). Applying Eq. (1.79) to Eq. (1.123), the Boussinesq based turbulent energy transport equation is

$$\begin{aligned}\rho\left[\frac{\partial k}{\partial t}+\overline{\mathbf{V}}\cdot\nabla k\right] &= 2\mu_t\,\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}})-\tfrac{2}{3}\left(\rho k+\mu_t\nabla\cdot\overline{\mathbf{V}}\right)\nabla\cdot\overline{\mathbf{V}}\\ &-2\mu\left(\overline{\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}})}-\tfrac{1}{3}\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2}\right)+\overline{\widetilde{p}\left(\nabla\cdot\widetilde{\mathbf{V}}\right)}\\ &+\nabla\cdot\left[\left(\nu+\frac{\nu_t}{\sigma_k}\right)\left(\rho\nabla k+\tfrac{2}{3}\nabla\left(\rho k+\mu_t\nabla\cdot\overline{\mathbf{V}}\right)-2\nabla\cdot\left[\mu_t\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}})\right]\right)\right]\end{aligned} \tag{1.124}$$

This last equation can be simplified by denoting $\overline{S}^2$ and $\widetilde{S}^2$ the magnitude of the mean and fluctuating components of the strain rate tensor, respectively

$$\overline{S}^2=\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}}) \tag{1.125}$$

$$\widetilde{S}^2=\overline{\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}})} \tag{1.126}$$

Using Eqs. (1.125)–(1.126) in Eq. (1.124) yields

$$\begin{aligned}\rho\left[\frac{\partial k}{\partial t}+\overline{\mathbf{V}}\cdot\nabla k\right] &= 2\mu_t\overline{S}^2-\tfrac{2}{3}\left(\rho k+\mu_t\nabla\cdot\overline{\mathbf{V}}\right)\nabla\cdot\overline{\mathbf{V}}-2\mu\left(\widetilde{S}^2-\tfrac{1}{3}\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2}\right)\\ &+\overline{\widetilde{p}\left(\nabla\cdot\widetilde{\mathbf{V}}\right)}+\nabla\cdot\left[\left(\nu+\frac{\nu_t}{\sigma_k}\right)\left(\rho\nabla k+\tfrac{2}{3}\nabla\left(\rho k+\mu_t\nabla\cdot\overline{\mathbf{V}}\right)-2\nabla\cdot\left[\mu_t\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}})\right]\right)\right]\end{aligned} \tag{1.127}$$

For incompressible flow $\nabla(\rho k)=\rho\nabla k$ and the Boussinesq based turbulent energy transport equation given in Eq. (1.127) reduces to

$$\rho\left[\frac{\partial k}{\partial t}+\overline{\mathbf{V}}\cdot\nabla k\right]=2\mu_t\overline{S}^2-2\mu\widetilde{S}^2+\nabla\cdot\left[\left(\nu+\frac{\nu_t}{\sigma_k}\right)\left(\frac{5}{3}\rho\nabla k-2\nabla\cdot\left[\mu_t\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}})\right]\right)\right] \tag{1.128}$$

The term $-2\mu\widetilde{S}^2$ in Eq. (1.128) for incompressible flow or the term $-2\mu\left(\widetilde{S}^2-\tfrac{1}{3}\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2}\right)$ for compressible flow in Eq. (1.127) represents the exact volumetric dissipation of turbulent kinetic energy. It is an unknown function of the turbulent velocity fluctuations. To close the formulation, there needs to be an additional equation to relate this dissipation to the other turbulence parameters and the mean flow. If

Eq. (1.127) is to be used as part of the conventional *k*-ε turbulence model, this volumetric dissipation can be written in terms of the unknown dissipation per unit mass

$$\widetilde{\varepsilon} = 2\nu\left(\widetilde{S}^2 - \frac{1}{3}\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2}\right) = 2\nu\left(\overline{\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}}):\overline{\overline{\mathbf{S}}}(\widetilde{\mathbf{V}})} - \frac{1}{3}\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2}\right) \tag{1.129}$$

Using Eq. (1.129) to model the dissipation in Eq. (1.127) and neglecting the pressure dilatation term, as it was done in the development of the traditional model, yields the turbulent-energy transport equation that can be used to replace the traditional turbulent-energy transport equation given in Eq. (1.84)

$$\begin{aligned}\rho\left[\frac{\partial k}{\partial t} + \overline{\mathbf{V}}\cdot\nabla k\right] &= 2\mu_t\overline{S}^2 - \frac{2}{3}\left(\rho k + \mu_t\nabla\cdot\overline{\mathbf{V}}\right)\nabla\cdot\overline{\mathbf{V}} - \rho\widetilde{\varepsilon} \\ &+ \nabla\cdot\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\left(\rho\nabla k + \frac{2}{3}\nabla\left(\rho k + \mu_t\nabla\cdot\overline{\mathbf{V}}\right) - 2\nabla\cdot\left[\mu_t\overline{\overline{\mathbf{S}}}(\overline{\mathbf{V}})\right]\right)\right]\end{aligned} \tag{1.130}$$

The symbol $\widetilde{\varepsilon}$ in Eq. (1.130) is simply used to indicate that this is the exact dissipation per unit mass, as defined in Eq. (1.129) and not the approximate dissipation per unit mass, defined in Eq. (1.80). Nevertheless, conventional models are based on the assumption that ε is the dissipation per unit mass. Thus, ε and $\widetilde{\varepsilon}$ can be interchanged in conventional *k*-ε and *k*-ω turbulence models.

The difference between Eq. (1.84) and (1.130) resides only in the last term on the right-hand side, which accounts for the molecular and turbulent transport of kinetic energy. Eq. (1.84) is based on the approximation that the turbulence parameter ε, defined in Eq. (1.80), includes only the viscous dissipation. This approximation results in neglecting a portion of the molecular transport. Although the neglected molecular transport term is small, subsequent application of the Boussinesq analogy between molecular and turbulent transport results in also neglecting a portion of the turbulent transport, which may not be insignificant. The more rigorous development used to obtain Eq. (1.130) produces a molecular transport term that is based only on the assumption of a Newtonian fluid. Thus, one might expect the Boussinesq analogy applied to this molecular transport term to be more realistic. For the traditional term used in Eq. (1.84), both the molecular and turbulent transports are modeled as pure gradient diffusion. However, the more rigorous development of Eq. (1.130) shows that, in general, this transport is not pure gradient diffusion.

The exact dissipation term defined in Eq. (1.129) can be written as

$$\begin{aligned}\widetilde{\varepsilon} &= 2\nu\left(\widetilde{S}^2 - \tfrac{1}{3}\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2}\right)\\ &= \nu\left\{\widetilde{\omega}^2 + \frac{4}{3}\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2} - 2\nabla\cdot\left[\frac{1}{\rho}\left(\nabla\cdot\bar{\bar{\tau}}\right) + \overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)\widetilde{\mathbf{V}}}\right]\right\}\end{aligned} \tag{1.131}$$

where $\widetilde{\omega}$ is the root-mean-square fluctuating vorticity.

$$\widetilde{\omega}^2 = \overline{\left(\nabla\times\widetilde{\mathbf{V}}\right)\cdot\left(\nabla\times\widetilde{\mathbf{V}}\right)} \tag{1.132}$$

Equation (1.120) can be rewritten in terms of the turbulent vorticity

$$\begin{aligned}\rho\left[\frac{\partial k}{\partial t} + \overline{\mathbf{V}}\cdot\nabla k\right] &= \bar{\bar{\tau}} : \bar{\bar{\mathbf{J}}}\left(\overline{\mathbf{V}}\right) - \mu\left\{\widetilde{\omega}^2 + \frac{4}{3}\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2} - 2\nabla\cdot\left[\frac{1}{\rho}\left(\nabla\cdot\bar{\bar{\tau}}\right) + \overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)\widetilde{\mathbf{V}}}\right]\right\}\\ &+ \overline{\widetilde{p}\left(\nabla\cdot\widetilde{\mathbf{V}}\right)} + \nabla\cdot\left(\mu\nabla k - \nu\nabla\cdot\bar{\bar{\tau}}\right) - \nabla\cdot\left(\tfrac{1}{2}\rho\overline{\widetilde{V}^2\widetilde{\mathbf{V}}} + \overline{\widetilde{p}\widetilde{\mathbf{V}}} + \tfrac{2}{3}\mu\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)\widetilde{\mathbf{V}}}\right)\end{aligned} \tag{1.133}$$

The only approximation that was made in the development of Eq. (1.133) is that of a Newtonian fluid. Applying Boussinesq's analogy between molecular and turbulent transport gives

$$\begin{aligned}\rho\left[\frac{\partial k}{\partial t} + \overline{\mathbf{V}}\cdot\nabla k\right] &= \bar{\bar{\tau}} : \bar{\bar{\mathbf{J}}}\left(\overline{\mathbf{V}}\right) - \mu\widetilde{\omega}^2 - \mu\frac{4}{3}\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2} + \mu\left\{2\nabla\cdot\left[\frac{1}{\rho}\left(\nabla\cdot\bar{\bar{\tau}}\right) + \overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)\widetilde{\mathbf{V}}}\right]\right\}\\ &+ \overline{\widetilde{p}\left(\nabla\cdot\widetilde{\mathbf{V}}\right)} + \nabla\cdot\left[\left(\nu + \tfrac{\nu_t}{\sigma_k}\right)\left(\rho\nabla k - \nabla\cdot\bar{\bar{\tau}}\right)\right]\end{aligned} \tag{1.134}$$

Using Eq. (1.122) for the Reynolds stress tensor and Eq. (1.38) for the Reynolds stress tensor in Eq. (1.134) produces the Boussinesq transport equation

$$\begin{aligned}\rho\left[\frac{\partial k}{\partial t} + \overline{\mathbf{V}}\cdot\nabla k\right] &= 2\mu_t\bar{S}^2 - \frac{2}{3}\left(\rho k + \mu_t\nabla\cdot\overline{\mathbf{V}}\right)\nabla\cdot\overline{\mathbf{V}} - \mu\widetilde{\omega}^2 - \frac{4}{3}\mu\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)^2}\\ &- 4\mu\nabla\cdot\left[\frac{1}{\rho}\left\{\frac{1}{3}\nabla\left(\rho k + \mu_t\nabla\cdot\overline{\mathbf{V}}\right) - \nabla\cdot\left[\mu_t\bar{\bar{\mathbf{S}}}\left(\overline{\mathbf{V}}\right)\right]\right\} - \frac{1}{2}\overline{\left(\nabla\cdot\widetilde{\mathbf{V}}\right)\widetilde{\mathbf{V}}}\right]\\ &+ \overline{\widetilde{p}\left(\nabla\cdot\widetilde{\mathbf{V}}\right)} + \nabla\cdot\left[\left(\nu + \tfrac{\nu_t}{\sigma_k}\right)\left(\rho\nabla k + \frac{2}{3}\nabla\left(\rho k + \mu_t\nabla\cdot\overline{\mathbf{V}}\right) - 2\nabla\cdot\left[\mu_t\bar{\bar{\mathbf{S}}}\left(\overline{\mathbf{V}}\right)\right]\right)\right]\end{aligned} \tag{1.135}$$

Huang et al. [41] have shown that the dilatational dissipation rate and other terms involving the divergence of the fluctuating velocity are negligibly small, at least up to supersonic mean Mach 3. Hence Eq. (1.135) is closely approximated as

$$\begin{aligned}\rho\left[\frac{\partial k}{\partial t}+\overline{\mathbf{V}}\cdot\nabla k\right] &= 2\mu_t\overline{S}^2-\frac{2}{3}\left(\rho k+\mu_t\nabla\cdot\overline{\mathbf{V}}\right)\nabla\cdot\overline{\mathbf{V}} \\ &\quad -\mu\,\widetilde{\omega}^2-4\mu\nabla\cdot\left[\frac{1}{\rho}\left\{\frac{1}{3}\nabla\left(\rho k+\mu_t\nabla\cdot\overline{\mathbf{V}}\right)-\nabla\cdot\left[\mu_t\overline{\overline{\mathbf{S}}}\left(\overline{\mathbf{V}}\right)\right]\right\}\right] \\ &\quad +\nabla\cdot\left[\left(\nu+\frac{\nu_t}{\sigma_k}\right)\left(\rho\nabla k+\frac{2}{3}\nabla\left(\rho k+\mu_t\nabla\cdot\overline{\mathbf{V}}\right)-2\nabla\cdot\left[\mu_t\overline{\overline{\mathbf{S}}}\left(\overline{\mathbf{V}}\right)\right]\right)\right]\end{aligned} \tag{1.136}$$

The three lines on the right-hand side of Eq. (1.136) are production, dissipation and the combination of molecular and turbulent transport, respectively. For steady-state incompressible flow, the continuity equation defined in Eq. (1.25) substituted in Eq. (1.136) yields

$$\begin{aligned}\overline{\mathbf{V}}\cdot\nabla k &= 2\nu_t\overline{S}^2-\nu\left(\widetilde{\omega}^2+4\nabla\cdot\left\{\frac{1}{3}\nabla k-\nabla\cdot\left[\nu_t\overline{\overline{\mathbf{S}}}\left(\overline{\mathbf{V}}\right)\right]\right\}\right) \\ &\quad +\nabla\cdot\left[\left(\nu+\frac{\nu_t}{\sigma_k}\right)\left(\frac{5}{3}\nabla k-2\nabla\cdot\left[\nu_t\overline{\overline{\mathbf{S}}}\left(\overline{\mathbf{V}}\right)\right]\right)\right]\end{aligned} \tag{1.137}$$

Based on the dimensional analysis given in Eq. (1.86), the mean vortex wavelength, λ, is defined in terms of the ratio of the mean square magnitude of the fluctuating velocity to the mean square magnitude of the fluctuating vorticity.

$$\lambda^2=C_\lambda\frac{\overline{\widetilde{\mathbf{V}}\cdot\widetilde{\mathbf{V}}}}{2\overline{\left(\nabla\times\widetilde{\mathbf{V}}\right)\cdot\left(\nabla\times\widetilde{\mathbf{V}}\right)}} \tag{1.138}$$

In terms of the turbulent kinetic energy defined in Eq. (1.39) and the root-mean-square fluctuating vorticity $\widetilde{\omega}^2$, the mean vortex wavelength is

$$\lambda^2=C_\lambda\frac{k}{\widetilde{\omega}^2} \tag{1.139}$$

Substituting Eq. (1.166) in Eq. (1.137) yields

$$\overline{\mathbf{V}}\cdot\nabla k = 2\nu_t \overline{S}^2 - \nu\left(C_\lambda \frac{k}{\lambda^2} + 4\nabla\cdot\left\{\frac{1}{3}\nabla k - \nabla\cdot\left[\nu_t \bar{\bar{\mathbf{S}}}\left(\overline{\mathbf{V}}\right)\right]\right\}\right) + \nabla\cdot\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\left(\frac{5}{3}\nabla k - 2\nabla\cdot\left[\nu_t \bar{\bar{\mathbf{S}}}\left(\overline{\mathbf{V}}\right)\right]\right)\right] \tag{1.140}$$

The algebraic relation for the turbulent eddy viscosity is

$$\nu_t = \lambda k^{1/2} \tag{1.141}$$

The continuity equation defined in Eq. (1.25), the Boussinesq-RANS equations defined in Eq. (1.42), the algebraic equation for the kinematic eddy viscosity given in Eq. (1.143) with the turbulent-energy-transport equation from Eq. (1.142) and a closing equation for the mean vortex wavelength λ define a system of equations for an incompressible steady-state fluid. Eventually, the model will derive a transport equation for the second turbulence variable. To assess the accuracy of the new turbulent model, the resulting velocity, turbulent kinetic energy and vorticity will be compared to reference distributions. Because of the high frequencies of the turbulent fluctuations, it is not possible to measure the vorticity at high Reynoldds numbers. Therefore, the vorticity has to be compared to an algebraic relation. This dissertation is aimed at developing a reference distribution for the second turbulence variable at fully rough flows.

C. Fully Developed Flow in a Circular Pipe

Fully developed flow in a pipe has historically been the foundational case and has been studied in great detail. This flow scenario is one of the easiest cases to evaluate the closure coefficients.

For fully developed axisymmetric flow in a circular pipe of radius *R*, there can be no θ-velocity component, no change in the mean velocity or pseudo mean pressure with respect to θ, and no change in the mean velocity with respect to *z*,

$$\overline{V}_\theta = 0, \quad \frac{\partial \overline{\mathbf{V}}}{\partial \theta} = 0, \quad \frac{\partial \hat{\overline{p}}}{\partial \theta} = 0, \quad \frac{\partial \overline{\mathbf{V}}}{\partial z} = 0 \tag{1.142}$$

The mean strain-rate tensor is in cylindrical coordinates

$$\bar{\bar{\mathbf{S}}}\left(\overline{\mathbf{V}}\right)=\frac{1}{2}\begin{bmatrix} 2\dfrac{\partial \overline{V}_r}{\partial r} & \left(r\dfrac{\partial \overline{V}_\theta/r}{\partial r}+\dfrac{1}{r}\dfrac{\partial \overline{V}_r}{\partial \theta}\right) & \left(\dfrac{\partial \overline{V}_z}{\partial r}+\dfrac{\partial \overline{V}_r}{\partial z}\right) \\ \left(\dfrac{1}{r}\dfrac{\partial \overline{V}_r}{\partial \theta}+r\dfrac{\partial \overline{V}_\theta/r}{\partial r}\right) & 2\left(\dfrac{1}{r}\dfrac{\partial \overline{V}_\theta}{\partial \theta}+\dfrac{\overline{V}_r}{r}\right) & \left(\dfrac{1}{r}\dfrac{\partial \overline{V}_z}{\partial \theta}+\dfrac{\partial \overline{V}_\theta}{\partial z}\right) \\ \left(\dfrac{\partial \overline{V}_r}{\partial z}+\dfrac{\partial \overline{V}_z}{\partial r}\right) & \left(\dfrac{\partial \overline{V}_\theta}{\partial z}+\dfrac{1}{r}\dfrac{\partial \overline{V}_z}{\partial \theta}\right) & 2\dfrac{\partial \overline{V}_z}{\partial z} \end{bmatrix} \tag{1.143}$$

1. Continuity Equation

For steady incompressible flow, the ensemble averaged continuity equation expressed in Eq. (1.22) becomes

$$\nabla\cdot\overline{\mathbf{V}}=0 \tag{1.144}$$

Using Eqs. (1.142), the continuity equation becomes

$$\frac{1}{r}\frac{\partial\left(r\overline{V}_r\right)}{\partial r}=0 \tag{1.145}$$

The no-slip boundary condition at the pipe wall requires

$$\overline{V}_r\left(R\right)=0 \tag{1.146}$$

Integrating Eq. (1.145) subject to Eq. (1.146) results in

$$\overline{V}_r\left(r\right)=0 \tag{1.147}$$

2. Boussinesq RANS Equations

Applying Eqs. (1.142) and Eq. (1.147) to the steady-state, incompressible Boussinesq RANS equations given in Eq. (1.42) results in

$$0=-\frac{1}{\rho}\frac{\partial\hat{\overline{p}}}{\partial r}\mathbf{i}_r-\frac{1}{\rho}\frac{\partial\hat{\overline{p}}}{\partial z}\mathbf{i}_z+\frac{1}{r}\frac{\partial}{\partial r}\left[\left(\nu+\nu_t\right)r\frac{\partial\overline{V}_z}{\partial r}\right]\mathbf{i}_z \tag{1.148}$$

The *r*-component of Eq. (1.148) yields

$$\frac{\partial \hat{\overline{p}}}{\partial r} = 0\,, \quad \hat{\overline{p}}(r,\theta,z) = \hat{\overline{p}}(z) \tag{1.149}$$

The pseudo mean pressure depends only on the axial coordinate z and the mean velocity depends only on the radial coordinate r. Thus, the z component of Eq. (1.148) can be written as an ordinary differential equation.

$$\frac{1}{r}\frac{d}{dr}\left[(\nu + \nu_t) r \frac{d\overline{V}_z}{dr}\right] = \frac{1}{\rho}\frac{d\hat{\overline{p}}}{dz} \tag{1.150}$$

The symmetry boundary condition at the pipe centerline and the no-slip boundary condition at the pipe wall require

$$\frac{d\overline{V}_z}{dr}(r=0) = 0 \tag{1.151}$$

$$\overline{V}_z(r=R) = 0 \tag{1.152}$$

The right-hand side of Eq. (1.150) is a function of only z, whereas the left-hand side is a function of only r. Thus both sides of Eq. (1.150) must be constant. Integrating this equation once with respect to r and applying Eq. (1.151) gives

$$(\nu + \nu_t)\frac{d\overline{V}_z}{dr} = \frac{1}{2}\frac{1}{\rho}\frac{d\hat{\overline{p}}}{dz} r \tag{1.153}$$

The left-hand side of Eq. (1.153) evaluated at the wall is the negative of the shear velocity squared. Thus,

$$\left[(\nu + \nu_t)\frac{d\overline{V}_z}{dr}\right]_{r=R} = -\frac{\tau_w}{\rho} = -u_\tau^2 = \frac{1}{2}\frac{1}{\rho}\frac{d\hat{\overline{p}}}{dz} R \quad \text{or} \quad u_\tau = \sqrt{-\frac{1}{2}\frac{R}{\rho}\frac{d\hat{\overline{p}}}{dz}} \tag{1.154}$$

Hence, Eq. (1.153) can be written as

$$(\nu + \nu_t)\frac{d\overline{V}_z}{dr} = -u_\tau^2 \frac{r}{R} \tag{1.155}$$

3. Turbulent-Kinetic-Energy Equation

For fully developed axisymmetric flow, there can be no change in the turbulent energy with respect to either θ or z

$$\nabla k = \frac{dk}{dr}\mathbf{i}_r + 0\mathbf{i}_\theta + 0\mathbf{i}_z \tag{1.156}$$

$$\nabla^2 k = \frac{1}{r}\frac{d}{dr}\left(r\frac{dk}{dr}\right) \tag{1.157}$$

$$\nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\nabla k\right] = \frac{1}{r}\frac{d}{dr}\left(r\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{dk}{dr}\right) \tag{1.158}$$

Using Eqs. (1.142)–(1.143) and (1.156)–(1.158) in Eq. (1.140) yields

$$-\frac{5}{3r}\frac{d}{dr}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)r\frac{dk}{dr}\right] = \nu_t\left(\frac{d\overline{V}_z}{dr}\right)^2 - \nu\left[C_\lambda \frac{k}{\lambda^2} + \frac{4}{3r}\frac{d}{dr}\left(r\frac{dk}{dr}\right)\right] \tag{1.159}$$

The left-hand side of Eq. (1.159) is the net outflow of turbulent kinetic energy per unit volume resulting from the combined effects of molecular and turbulent diffusion. The two terms on the right-hand side of Eq. (1.159) are the generation and the dissipation, respectively. If the molecular viscosity is constant, a portion of the dissipation term can be combined with the molecular diffusion to yield

$$-\frac{1}{r}\frac{d}{dr}\left[\left(\frac{\nu}{3} + \frac{5\nu_t}{3\sigma_k}\right)r\frac{dk}{dr}\right] = \nu_t\left(\frac{d\overline{V}_z}{dr}\right)^2 - \nu\, C_\lambda \frac{k}{\lambda^2} \tag{1.160}$$

4. Nondimensional Form

The formulation can be nondimensionalized using the following dimensionless parameters

$$u^+ \equiv \frac{\overline{V}_z}{u_\tau}, \quad k^+ \equiv \frac{k}{u_\tau^2}, \quad \hat{\lambda} \equiv \frac{\lambda}{R}, \quad \hat{r} \equiv \frac{r}{R}, \quad \hat{\nu} \equiv \frac{\nu}{u_\tau R}, \quad \hat{\nu}_t \equiv \frac{\nu_t}{u_\tau R} \tag{1.161}$$

The once integrated Boussinesq RANS equation and the no-slip boundary condition becomes

$$\frac{du^+}{d\hat{r}} = -\frac{\hat{r}}{\hat{\nu} + \hat{\nu}_t}, \quad u^+\left(\hat{r}=1\right) = 0 \tag{1.162}$$

The algebraic equation for the kinematic eddy viscosity becomes

$$\hat{\nu}_t = \hat{\lambda}\sqrt{k^+} \tag{1.163}$$

Substituting Eqs. (1.161) in Eq. (1.160), the turbulent energy transport equation is non-dimensionalized as

$$-\frac{d}{d\hat{r}}\left[\left(\frac{\hat{\nu}}{3} + \frac{5\hat{\nu}_t}{3\sigma_k}\right)\hat{r}\frac{dk^+}{d\hat{r}}\right] = \hat{\nu}_t\left(\frac{du^+}{d\hat{r}}\right)^2 \hat{r} - C_\lambda \frac{\hat{\nu}{k^+}^2}{\hat{\nu}_t{}^2}\hat{r} \tag{1.164}$$

Defining

$$\hat{Q} \equiv -\left(\frac{\hat{\nu}}{3} + \frac{5\hat{\nu}_t}{3\sigma_k}\right)\hat{r}\frac{dk^+}{d\hat{r}} \tag{1.165}$$

and using this definition along with the Boussinesq-RANS relation for the velocity gradient, the second-order *k*-transport equation can be written as two first order transport equations with associated boundary conditions

$$\begin{aligned} \frac{dk^+}{d\hat{r}} &= -\frac{\hat{Q}}{\left(\frac{\hat{\nu}}{3} + \frac{5\hat{\nu}_t}{3\sigma_k}\right)\hat{r}}, \quad k^+\Big|_{\hat{r}=1} = k^+_{\text{wall}} \\ \frac{d\hat{Q}}{d\hat{r}} &= \hat{\nu}_t\left(\frac{du^+}{d\hat{r}}\right)^2 \hat{r} - C_\lambda \frac{\hat{\nu}{k^+}^2}{\hat{\nu}_t{}^2}\hat{r}, \quad \hat{Q}\Big|_{\hat{r}=0} = 0 \end{aligned} \tag{1.166}$$

This gives a system of equations for the turbulent transport equation based on a fourth order Runga-Kutta algorithm, along with the two boundary conditions. However, the equation for the second

turbulence variable is still missing and the value of the turbulent kinetic energy at the wall is unknown. An algebraic relation for the second turbulence variable will be proposed in the following chapter.

D. Closing the Phillips Energy-Vorticity Turbulence Model

A closure problem arises in the Phillips energy-vorticity turbulence model because the second turbulence variable is unknown. The second turbulence variable could be represented by the root mean square fluctuating vorticity or the mean vortex wavelength. A relation between these two variables exists and is given in Eq. (1.139). Two-equation turbulence models give a differential equation for the second turbulence variable. This differential equation is traditionally obtained by analogy to the turbulent kinetic energy and dimensional analysis. This differential equation is dimensionally correct and has no physical basis.

Hunsaker [44] proposed several closure methods for the energy vorticity turbulence model in the traditional two-equation method. However, all these attempts were unsuccessful. Hunsaker mainly investigated a transport equation for the root mean square fluctuating vorticity to close the model. Several versions were given for the transport equation of the root mean square fluctuating vorticity featuring different closure coefficients. However, the closure coefficients could not be found to match experimental data.

VI. Summary and Conclusions

The turbulent-energy-transport equation is usually developed directly from the trace of the Reynolds-stress-transport equation. Subsequently, the turbulent-energy dissipation per unit mass is modeled using some approximations, although it is often referred to incorrectly as the exact dissipation. This modeling of the dissipation has led to omitting some elements in the molecular transport term. The molecular transport term is not a pure gradient diffusion process as is currently assumed with traditional models.

The dissipation of turbulent energy has been studied in great detail and it is now generally recognized that the traditional definition for ε is not precisely the turbulent-energy dissipation per unit mass. However, the continued use of this traditional definition for ε has been justified on the grounds that the additional terms are small [41]. It is not surprising that these terms have been shown to be small, because they actually arise from the fluctuating compressibility effects and molecular transport, not from viscous dissipation.

Because compressibility effects are negligible for Mach numbers less than 0.3 and turbulent fluctuations are typically on the order of 10 percent of the mean flow, fluctuating compressibility effects are usually negligible for Mach numbers less than three. Furthermore, for most turbulent flows, molecular transport is negligible compared to turbulent transport. Thus, the most significant concern with the traditional k-ε and k-ω turbulence models is not in the lack of precision in defining the energy dissipation.

Molecular transport of turbulent energy has not been studied in such detail. There has been little or no discussion in the literature of the fact that the molecular diffusion term, $\nu\nabla^2 k$, is not the total molecular transport of turbulent energy per unit mass. Little interest has been shown in molecular transport of turbulent kinetic energy because molecular transport is typically recognized as being negligible compared to turbulent transport. What seems to have been overlooked in literature is the fact that we cannot adequately apply the Boussinesq analogy between molecular and turbulent transport without a thorough understanding of molecular transport. The alternate turbulent-kinetic-energy transport equation, which is developed from moments of the Navier-Stokes equations, provides this requisite understanding. The most important contribution in the development of the proposed turbulent-kinetic-energy transport equation is the inclusion of the last term in Eq. (1.140), which has not been included in any other turbulence models.

Based on fewer assumptions, Phillips developed a turbulent-kinetic-energy transport equation that could significantly improve the accuracy of turbulence modeling and its understanding. The continuity equation defined in Eq. (1.25), the Boussinesq-RANS equations defined in Eq. (1.42), the algebraic equation for the kinematic eddy viscosity given in Eq. (1.143) and the Phillips turbulent-energy-transport equation given in Eq. (1.142) define a system of equations for an incompressible steady-state fluid. However, the equation describing the transport of the second turbulent variable is missing.

With traditional two-equation energy-dissipation turbulence models like the k-ϵ and k-ω models, the transport equations for the second turbulence variable were obtained simply from dimensional analysis and analogy with the turbulent-kinetic-energy transport equation. They were not developed in a rigorous manner from the Navier-Stokes equations. These transport equations have no basis in physics. About all that can be said concerning the validity of these dissipation transport equations is that they are dimensionally correct. Nevertheless, some success has been achieved using this approach. The Phillips energy-vorticity turbulence model could be closed in a similar manner. However, Hunsaker [44] has

examined this approach in considerable detail with no success. It now seems apparent that a more rigorous approach to the development of a turbulent-vorticity transport equation must be taken.

Eventually, the Phillips energy-vorticity model will encompass a transport equation for the root-mean-square fluctuating vorticity. Once this transport equation is developed, the model must be compared to experimental data or established relations for the mean velocity, turbulent kinetic energy, and fluctuating vorticity distributions. However, the fluctuating vorticity cannot be measured directly, and at high Reynolds numbers, the fluctuating components of velocity include frequencies too high to be measured accurately. Because experimental data for the fluctuating variables are unobtainable for fully rough flow and there are currently no established relations for the distributions of the turbulence variables, a relation for the second turbulent variable needs to be developed. This algebraic relation has to be derived such that the mean velocity distribution and the friction factor obtained from the resulting model match the well established relations obtained for fully rough flow. The following chapter will present an algebraic relation for the mean turbulent wavelength, which has been developed for fully rough pipe flow. The mean-fluctuating-vorticity distribution can be obtained directly from this mean-turbulent-wavelength distribution and the turbulent-kinetic-energy transport equation by applying the algebraic relation given in Eq. (1.139).

CHAPTER 2

EMPIRICAL RELATION FOR THE MEAN VORTEX WAVELENGTH

I. Introduction

Modeling turbulent behavior near perfectly smooth walls can present significant difficulties. All available experimental data have been taken near walls with some degrees of roughness. Because at high roughness Reynolds numbers, certain flow properties become independent of roughness Reynolds numbers, the argument has been made that developing a model for fully rough flow may be more straightforward than developing a model that exhibits the correct behavior near a perfectly smooth wall. Such a rough-wall model could then be extended to lower roughness Reynolds numbers and eventually to the hydraulically smooth surface, which is simply a low-roughness-Reynolds-number asymptote. The first step in this process is to develop a turbulence model that is consistent for high roughness Reynolds numbers.

In most practical applications, the surface over which flow occurs is significantly rough, and in spite of extensive studies, there is still much to be learned. The significant effects created by surface roughness on skin friction and velocity distributions have led many investigators to study this problem. Data on velocity distributions for wall-bounded flow over hydraulically smooth surfaces are available and laws controlling the resistance or friction factor are well established and widely accepted, see the work of Prandtl [45] and Nikuradse [46]. However, similar experimental data obtained from rough surfaces are sparser. The most widely accepted correlations for pipe flow are all fundamentally based on the work of Nikuradse [47]. Nikuradse proposed an empirical formula based on thorough experimental investigations to predict the friction factor over a wide range of Reynolds numbers covering both laminar and turbulent flows having either hydraulically smooth or rough surfaces.

Traditionally turbulence models have been developed for the case of hydraulically smooth walls and a select few have been subsequently modified to model rough walls. At present, the rough wall option is only a secondary feature with ω-based turbulence models, such as k-ω [13, 48], shear stress transport [15, 49] and ω-Reynolds stress models [50] or for the case of laminar flow. For rough walls, the logarithmic velocity profile still exists but moves closer to the wall. Roughness effects are accounted for by modifying the wall functions based on an equivalent sand-grain roughness. In fact, the roughness height specified in

turbulence models is not exactly equal to the roughness height of the surface under consideration. Wall friction depends not only on roughness height but also on the type of roughness such as shape and distribution. Guidance can be obtained from White [51] and Schlichting [52, 53]. The k-ω model [14] has been accepted as the most capable of modeling rough-wall effects without implementing wall functions. The effects of surface roughness are commonly incorporated into the k-ω model by simply altering the surface boundary condition on ω.

Many two-equation turbulence models have been proposed on the basis of two transport equations, one for the turbulent kinetic energy and one for the dissipation. Although the k-ϵ and k-ω models are often thought to be fundamentally different, both models are based on a transport equation for a dissipation parameter, either ϵ or ω, which can be related through a change of variables. Robinson et al. [54] developed a k-ζ model based on a transport equation for the enstrophy. The model was further developed by Robinson and Hasen [55]. The enstrophy is defined as the mean square magnitude of the fluctuating vorticity. However, the approximate turbulent-energy dissipation term is defined in their model through a change of variables, which contradicts their definition for the enstrophy.

The Phillips energy vorticity turbulence model is intended to be a two-equation model, which will combine the turbulent-kinetic-energy transport equation given in Eq. (1.140) with a turbulent-vorticity transport equation. To evaluate closure coefficients and assess the model, the distribution of the turbulence variables resulting from the model must be compared to reference distributions. For fully rough pipe flow, there are no accurate experimental data available which describe the fluctuating velocity components. Therefore, the distributions of the turbulent kinetic energy and the fluctuating vorticity are not available.

This chapter presents an overview of some of the fundamental work on rough pipe flow and develops an algebraic relation for the mean vortex wavelength, which is one possible choice for the second turbulence variable. Another possible choice for the second turbulence variable is the RMS fluctuating vorticity, which can be determined directly from the mean vortex wavelength and the turbulent kinetic energy using the algebraic relation given in Eq. (1.139). The development presented in this chapter is based primarily on the work of Nikuradse [47], who performed a series of experiments on rough pipes that will be used to derive algebraic relations for the velocity profiles and the turbulent eddy viscosity in rough pipes.

II. Fully Developed Rough Pipe Flow

A. Fundamental Relations and Definitions

Fully developed pipe flow has been investigated by many researchers over the past centuries and is a common case study to evaluate a new turbulence model. When the fluid enters a pipe, its velocity will often be uniform across the pipe cross-section. Near the entrance, the fluid near the center of the pipe is not affected by the friction between fluid and pipe walls, but as the flow proceeds down the pipe, the effect of the wall friction moves in toward the pipe center until the velocity profile becomes constant. At the end of the entrance length, the flow enters the fully developed region and the governing equations of motion can be greatly simplified. This section presents a few of the most relevant relations for fully developed flow in a circular pipe.

For fully developed axisymmetric flow in a circular pipe, gradients in the flow properties with respect to the flow direction disappear and the profiles of flow properties become only dependent on the coordinate normal to the wall. The Boussinesq-RANS can be reduced to a one-dimensional problem.

$$\left(\nu + \nu_t\right)\frac{d\overline{V}_z}{dr} = -{u_\tau}^2\,\frac{r}{R}, \qquad \overline{V}_z\Big|_{r=R} = 0 \tag{2.1}$$

The left-hand side of Eq. (2.1) evaluated at the wall is the negative of the shear velocity u_τ squared. Thus, the wall shear stress is related to the pressure drop according to

$$\tau_w \equiv -\mu\left.\frac{d\overline{V}_z}{dr}\right|_{r=R} = -\frac{R}{2}\frac{d\hat{\overline{p}}}{dz} \tag{2.2}$$

There are many ways to nondimensionalize the Boussinesq-RANS equations: wall-scaled variables and relative variables, or pipe-scaled variables.

$$
\begin{aligned}
&u^+ = \frac{\overline{V}_z}{u_\tau}, \quad y^+ \equiv \frac{u_\tau y}{\nu}, \quad \nu^+ \equiv \frac{\nu_t}{\nu} \\
&\hat{r} = \frac{r}{R}, \quad \hat{\nu} \equiv \frac{\nu}{u_\tau R}, \quad \hat{\nu}_t \equiv \frac{\nu_t}{u_\tau R}, \quad \hat{u} = \frac{\overline{V}_z}{V_m} \\
&y \equiv R - r, \quad R_\tau \equiv \frac{u_\tau R}{\nu} = \frac{1}{\hat{\nu}}, \quad \hat{y} = \frac{y}{R}, \quad y^+ \equiv R_\tau \hat{y}
\end{aligned}
\tag{2.3}
$$

The parameter R_τ is commonly referred to as the shear Reynolds number and is the reciprocal of the nondimensional molecular viscosity $\hat{\nu}$. Using these definitions, the wall-scaled Boussinesq-RANS formulation for fully developed axisymmetric flow in a circular pipe including the no-slip boundary condition can be written in nondimensional form as

$$
\left(1 + \nu^+\right)\frac{du^+}{dy^+} = 1 - \frac{y^+}{R_\tau}, \qquad u^+\Big|_{y^+=0} = 0 \tag{2.4}
$$

Equation (2.4) applies over the domain $0 \leq y^+ \leq R_\tau$. Near the wall, the ratio y^+/R_τ is small compared to unity, so the left-hand side of Eq. (2.4) remains approximately constant in the near-wall region. Hence, this near-wall region is commonly referred to as the constant stress layer or the Couette flow region. Neglecting the term y^+/R_τ , Eq. (2.4) can be closely approximated near the pipe wall as

$$
\left(1 + \nu^+\right)\frac{du^+}{dy^+} = 1, \qquad u^+\Big|_{y^+=0} = 0 \tag{2.5}
$$

In the case of fully rough flow, the molecular viscosity is much less than the turbulent eddy viscosity throughout the flow field, so the non-dimensional eddy viscosity ν^+ is large compared to unity. Hence in the near-wall fully rough limit, Eq. (2.5) can be closely approximated as

$$
\nu^+ \frac{du^+}{dy^+} = 1, \qquad u^+\Big|_{y^+=0} = 0 \tag{2.6}
$$

The mean wall shear stress for fully developed flow in a pipe is traditionally characterized in terms of the Darcy friction factor, which is four times the Fanning friction factor C_f often called the skin-friction coefficient. The Darcy friction factor is a very important parameter in pipe study as it provides a means of estimating the head loss. The pressure drop can be easily measured along the length of the pipe and then the

wall shear stress can be directly computed. The wall shear stress is generally expressed in terms of the Darcy friction factor

$$\text{Darcy Friction Factor} \equiv 4C_f \equiv 4\frac{\tau_w}{\frac{1}{2}\rho V_m^2} = \frac{8}{\left(V_m / u_\tau\right)^2} \tag{2.7}$$

Equation (2.7) may be rewritten in terms of the pressure drop per unit length $\left(-d\hat{\bar{p}}/dz\right)$, the pipe diameter D and the dynamic pressure of the average flow $(1/2)\rho V_m^2$ all of which are readily available from experimental data. Substituting Eq. (2.3) in Eq. (2.7) yields

$$4C_f \equiv \frac{D}{\frac{1}{2}\rho V_m^2}\left(-\frac{d\hat{\bar{p}}}{dz}\right) \tag{2.8}$$

The characteristic velocity V_m used to define the Darcy friction factor is the spatial-mean velocity, which is also commonly called the bulk velocity,

$$u_m^+ \equiv \frac{V_m}{u_\tau} = \frac{\int_{r=0}^{R}\overline{V}_z r dr}{u_\tau\int_{r=0}^{R} r dr} = \frac{\int_{\hat{r}=0}^{1}u^+\hat{r}d\hat{r}}{\int_{r=0}^{R}\hat{r}d\hat{r}} = 2\int_{\hat{r}=0}^{1}u^+\hat{r}d\hat{r} = \frac{2}{R_\tau^2}\int_{y^+=0}^{R_\tau}u^+\left(R_\tau - y^+\right)dy^+ \tag{2.9}$$

The Darcy friction factor and Reynolds number can be expressed in terms of the bulk velocity as

$$\text{Re} \equiv \frac{V_m(2R)}{\nu} = \frac{2u_m^+}{\hat{\nu}}, \qquad 4C_f = \frac{8}{u_m^{+2}} \tag{2.10}$$

Because of the frequent occurrence of rough pipes in practice, the study of rough pipes is just as important as the study of flow along smooth surfaces. Darcy [56] made comprehensive and careful tests on 21 pipes of cast iron, lead, wrought iron, asphalt covered cast iron and glass. He noticed that the friction factor was dependent upon the type of surface and upon the diameter of the pipe. The type of surface is

commonly described in terms of a wall roughness k_r, which is characterized in terms of the ratio of the equivalent sand-grain roughness k_s to the pipe diameter

$$k_r \equiv \frac{k_s}{2R} \tag{2.11}$$

The roughness Reynolds number is defined as

$$k_s^+ \equiv \frac{u_\tau k_s}{\nu} = \frac{2k_r}{\hat{\nu}} = 2k_r R_\tau \tag{2.12}$$

For convenience, we define the ratio

$$\hat{R}_s = \frac{R}{\gamma k_s} \tag{2.13}$$

By combining Eqs. (2.9) and (2.12), the roughness Reynolds number can be determined from the bulk Reynolds number, the roughness height and the friction factor

$$k_s^+ = \frac{\mathrm{Re}\, k_r \sqrt{4C_f}}{\sqrt{8}} \tag{2.14}$$

The surface roughness has an effect on friction resistance. For laminar flow, this effect is negligible. However, a turbulent flow is strongly affected by roughness. In fact, three regions can be characterized by their roughness Reynolds number: hydraulically smooth walls, transitional roughness and fully rough flow. Darcy [56] noticed that for certain roughness at some high Reynolds numbers, his experimental data indicate that the friction factor is independent of the Reynolds number. He noticed that at a constant Reynolds number, the friction factor increases markedly for an increasing relative roughness. Nikuradse's [47] experiments on rough pipes confirmed those statements.

B. The Nikuradse Number

Nikuradse [47] studied the effect of coarse and fine roughness for a wide range of Reynolds numbers to determine the friction factor laws for six different degrees of relative roughness. Through careful

measurements, Nikuradse determined the dynamic pressure by means of Pitot tubes. The diameter of the pipe was determined from the weight of the water enclosed in the pipe. The static pressure gradients and resulting pressure drop were measured using a hooked tube placed at half the radius of the pipe. Nikuradse used sand grains of uniform roughness. The sand grains were sifted and spread on a flat plate. The diameters of the individual grains were measured by sliding the plate and the arithmetical average was computed. The pipes were first coated with a lacquer and then artificially roughened.

Nikuradse computed the friction factor from Eq. (2.8) for a wide range of Reynolds numbers and for six different degrees of roughness ratios. Nikuradse observed that three different ranges could be considered, depending upon the Reynolds number, namely laminar, transition and turbulent flows. Nikuradse's experimental friction factors obtained from experiments on hydraulically smooth-wall pipes [46] and on rough pipes [47] are shown in Fig. 2.1.

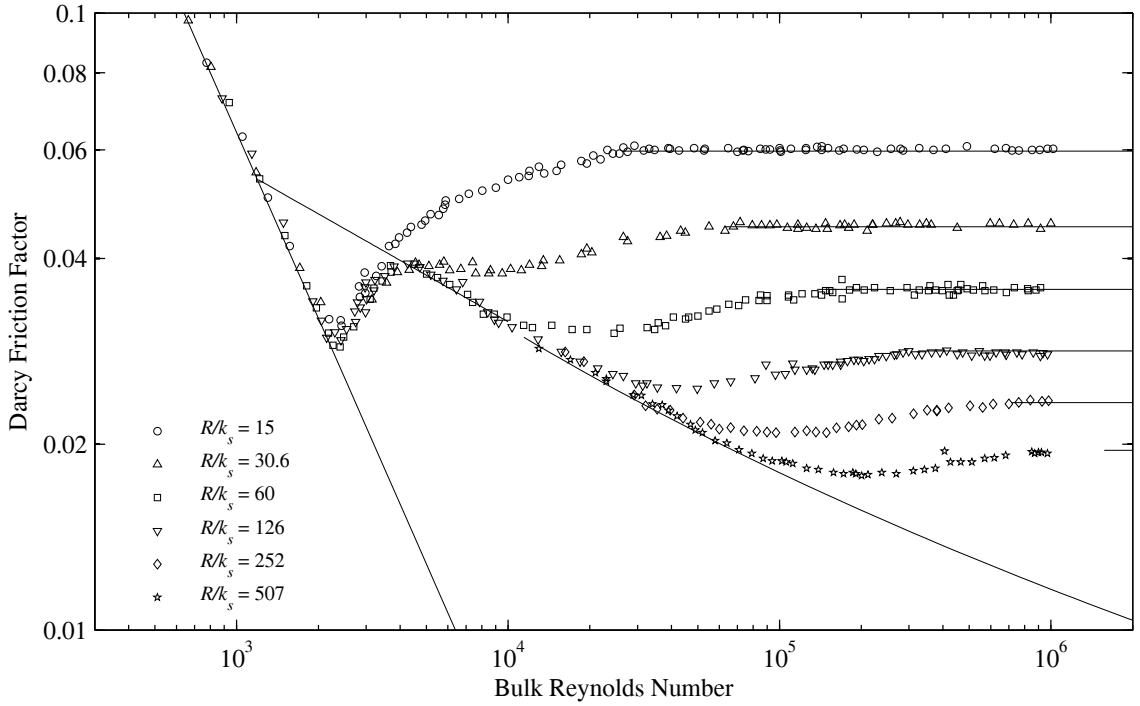

Fig. 2.1 Nikuradse's experimental friction factor.

Within the first range, that of low Reynolds number, the roughness has no effect on the resistance (or friction factor), and for all values of inverse roughness ratio R/k_s, the friction factor is the same as for a hydraulically smooth-wall pipe. In this first range, the thickness of the laminar sublayer is still larger than

the average roughness element. Therefore energy losses due to roughness are no greater than those for a smooth-wall pipe. In this range, Blasius [57] represented the friction factor by the law

$$4C_f = \frac{64}{\mathrm{Re}} \tag{2.15}$$

For setting up this friction factor formula, Blasius used the experiments of Saph and Schoder [58] who worked with water and Nusselt [59] who worked with air. This formula was later verified by Ombeck [60], Stanton and Pannell [61], Lees [62] and Jakob and Erk [63]. They all found insignificant deviations from Eq. (2.15)

The critical Reynolds number, separating laminar from turbulent flows, occurs at about the same position for all degrees of roughness ratios as for the smooth-wall pipe for a Reynolds number between 2150 and 2500. Within the first portion of turbulent flow in smooth pipes for a Reynolds number of up to $\mathrm{Re}=10^5$, the Blasius resistance law holds, as shown in Fig. 2.1.

$$4C_f = \frac{0.316}{\mathrm{Re}^{1/4}} \tag{2.16}$$

Within the second range, called the transition range, the influence of roughness becomes more noticeable. The friction factor increases with an increase in Reynolds number. The resistance factor depends on the Reynolds number as well as the roughness ratio. In this transition range, the thickness of the laminar sublayer is of the same magnitude as the average roughness element. Individual roughness elements extend through the laminar sublayer and cause vortices, which produce an additional loss of energy. As the Reynolds number increases, an increasing number of projections pass through the laminar sublayer because of the reduction in thickness. The additional energy loss becomes greater as the Reynolds number increases. This is expressed by the rise of the friction factor. For a given roughness type, the behavior in the transitional roughness regime is known only from experiment, see for example the work of Jimenez [64].

Within the third range, the friction factor is independent of the Reynolds number and depends only on the roughness ratio, so that the friction factor curves become parallel to the horizontal axis. In this range, the thickness of the laminar sublayer has become so small that all the roughness elements extend through it.

The energy loss due to the vortices has now attained a constant value and an increase in the Reynolds number no longer increases the resistance. In this range, Nikuradse [47] expressed the friction factor as

$$4C_f = \frac{1}{\left[1.74 + 2\log_{10}\left(\frac{R}{k_s}\right)\right]^2} \tag{2.17}$$

Equation (2.17) provided a key result in the development of our current capability to predict the pressure losses of turbulent flow through rough pipes, and it will be referred to herein as the Nikuradse equation. Equation (2.17) was the starting point for the development of the Colebrook equation [65] and the associated Moody chart [66]. Hence, both the Colebrook equation and the Moody chart assume the validity of the Nikuradse equation.

The Nikuradse equation is often presented in a form that differs slightly from Eq. (2.17). In the original work by Nikuradse [47], and its subsequent presentation by Schlichting [52], the pipe roughness was characterized using the dimensionless inverse roughness ratio R/k_s. Using Eq. (2.11), the Nikuradse equation given in Eq. (2.17) might be rearranged as

$$\frac{1}{\sqrt{4C_f}} = 2\log_{10}\left(10^{1.74/2}\right) + 2\log_{10}\left(\frac{1}{2k_r}\right) \tag{2.18}$$

Represented to two significant digits, Nikuradse's fully rough limit empirical equation is given by

$$\frac{1}{\sqrt{4C_f}} = 2.0\log_{10}\left(\frac{3.7}{k_r}\right) \tag{2.19}$$

In terms of the relative roughness k_s/D, where D is the pipe diameter, the most widely accepted form of the Nikuradse equation becomes

$$4C_f = \left[2.0\log_{10}\left(\frac{3.7}{k_s/D}\right)\right]^{-2} \tag{2.20}$$

Equations (2.15)–(2.17) are shown on Fig. 2.1 along with Nikuradse's experimental friction factors. Nikuradse, in his 1932 paper [46], plotted the parameter $-2.0\log_{10}(3.7/k_r)+1/(4C_f)^{1/2}$ versus the roughness Reynolds number. Although Nikuradse did not give a name to this parameter, a possible name that would give him the credit for his work could be the Nikuradse number. A possible definition for the Nikuradse number can be

$$\mathrm{Ni} \equiv 2.0\log_{10}\left(\frac{3.7}{k_r}\right)-\frac{1}{\sqrt{4C_f}} \tag{2.21}$$

Nikuradse [47] noticed that the entire field of Reynolds numbers from his experiments could be covered by plotting Eq. (2.21) as a function of the roughness Reynolds number k_s^+, as shown in Fig. 2.2. Schlichting [67] suggested that the data of Nikuradse deviate from the smooth-wall asymptote at about $k_s^+ \approx 5$ and from the fully rough asymptote at about $k_s^+ \approx 70$. Thus, flows with roughness Reynolds numbers greater than 70 are traditionally termed "fully rough" flows because the friction factor appears independent of the roughness Reynolds number in this region. Flows with roughness Reynolds numbers smaller than 5 have traditionally been termed "hydraulically smooth" because roughness effects appear to be negligible in this region. The question of when the roughness effects first become important was discussed by Perry and Abell [68], McKeon et al. [69, 70]. This discussion hinges on the influence of roughness in the transitional regime. It also touches on how the characteristic roughness height can be determined for an arbitrary roughness distribution, and whether a single length scale is an adequate description. The limit of 5 for the roughness Reynolds number for a hydraulically smooth-wall pipe is supported by Zaragola and Smits [71] and Schockling et al. [72].

The Nikuradse number for the fully rough limit is obtained from Eqs. (2.19)–(2.21).

$$\mathrm{Ni} \equiv 2.0\log_{10}\left(\frac{3.7}{k_r}\right)-\frac{1}{\sqrt{4C_f}} = 0 \tag{2.22}$$

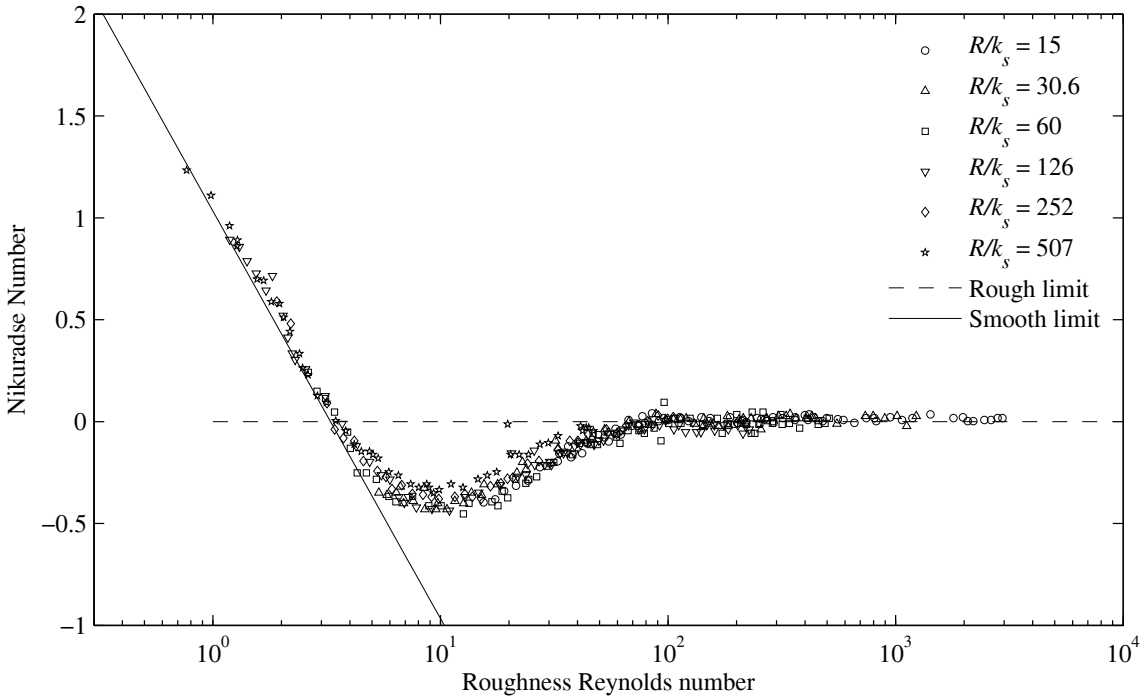

Fig. 2.2 Nikuradse number as a function of roughness Reynolds number for rough pipe flow.

In the limit of a smooth-wall pipe that is for low roughness Reynolds number, the friction factor is unaffected by the roughness such that all pipes have a behavior similar to that of a smooth-wall pipe. Based on the experimental data for rough pipes, Nikuradse [47] suggested the following expression

$$\frac{1}{\sqrt{4C_f}} - 2.0\log_{10}\left(\frac{R}{k_s}\right) = 0.8 + 2.0\log_{10}\left(k_s^+\right) \tag{2.23}$$

Using Eqs. (2.11) and (2.14) in Eq. (2.23), Eq. (2.23) may be written equivalently as

$$\frac{1}{\sqrt{4C_f}} = 2.0\log_{10}\left(\mathrm{Re}\sqrt{4C_f}\right) - 0.71 \tag{2.24}$$

Closer agreement is obtained if the constants are slightly modified to

$$\frac{1}{\sqrt{4C_f}} = 2.0\log_{10}\left(\mathrm{Re}\sqrt{4C_f}\right) - 0.80 \tag{2.25}$$

It can be seen from Eq. (2.25) that the friction factor for the smooth-wall limit is independent of the relative roughness because the roughness elements are within the laminar sublayer. Therefore, the friction factor depends on the Reynolds number only. Equation (2.25) may be written in the form

$$\frac{1}{\sqrt{4C_f}} = -2.0\log_{10}\left(\frac{2.51}{\mathrm{Re}\sqrt{4C_f}}\right) \tag{2.26}$$

In terms of the Nikuradse number, the smooth-wall limit given in Eq. (2.26) becomes

$$\mathrm{Ni} = 2.0\log_{10}\left(\frac{9.3}{\sqrt{8}k_s^+}\right) \tag{2.27}$$

Only Nikuradse's rough pipe flow experimental data [47] were used in Fig. 2.2. All six inverse roughness ratios in the range 15–507 are included. The figure includes the asymptotes for fully rough and smooth-wall limits as given in Eqs. (2.22) and (2.27), respectively.

Three different regions can be seen from Fig. 2.2, smooth-wall, transition and fully rough regions. Schlichting [52] suggested using $k_s^+ \le 5$ as the region of pure laminar friction. In this range, the contribution from turbulent friction may be neglected when compared to laminar friction. At low velocities in the laminar flow region, the laminar layer is much thicker than the depth of the irregularities. Therefore the actual texture has no effect on the nature of the flow. The flow is comparable to laminar flow and is said to be hydraulically smooth. Schlichting [53] proposed the range of roughness Reynolds number $k_s^+ \ge 70$ as the limit of pure turbulent friction. In this region where the laminar contribution is negligible compared to the turbulent friction, the friction factor is independent of the Reynolds number and the flow is said to be hydraulically rough.

It should be emphasized that Nikuradse based all of his experiments on pipes artificially roughened with sand grains of uniform roughness. Commercial pipes have an irregular texture but are still characterized by using an equivalent sand-grain roughness, which represents the average value for the irregularity heights.

C. The Colebrook Equation and Moody Diagram

Colebrook [65] carried out experiments on commercial pipes of different size and made of various materials including drawn brass, galvanized-coated, cast and wrought iron, bitumen-lined pipes and concrete-lined pipes and determined the friction factor for a wide range of Reynolds numbers. The surface of a commercial pipe differs from that of an artificially roughened pipe and cannot be described in terms of a single roughness size. An average equivalent particle size together with a shape factor and a size distribution should be used to characterize the surface accurately. Colebrook determined an equivalent roughness size for each of the commercial pipes he used during his experiments by comparing the friction factor at high Reynolds numbers and fully rough flow, to those obtained from his experiments with the Nikuradse friction factor obtained using pipes coated with uniform roughness elements.

Colebrook proposed an empirical correlation to determine the friction factor of commercial pipes, which covers the hydraulically smooth, transition and rough flow regions. Nikuradse [47] suggested empirical correlations for the cases of perfectly smooth-wall pipes given by Eq. (2.23) and for rough pipes where the flow is fully turbulent given by Eq. (2.19). No empirical correlations based on Nikuradse's experimental data were satisfactorily developed for the transition zone, between hydraulically smooth and rough pipes, which would match the friction factor obtained on commercial pipes. Colebrook [65] proposed an empirical correlation for the transition zone based on his commercial pipe experiments.

$$4C_f = \left[-2.0 \log_{10} \left(\frac{k_r}{3.7} + \frac{2.51}{\mathrm{Re}\sqrt{4C_f}} \right) \right]^{-2} \tag{2.28}$$

This equation is known as the Colebrook formula and gives results very close to experimental values for transitional behavior when using the effective roughness for commercial pipes as calculated by Colebrook. This equation is implicit, i.e., if the bulk Reynolds number and the roughness ratio are known, one must iterate to evaluate the Darcy friction factor. By setting the roughness Reynolds number k_r to 0, Eq. (2.28) reduces to Eq. (2.26), which represents the smooth-wall limit. Similarly, in the fully rough limit, as the Reynolds number increases, Eq. (2.28) reduces to Eq. (2.19).

The Colebrook equation given in Eq. (2.28) might be rearranged in terms of the Nikuradse number to yield

$$\mathrm{Ni} = 2.0 \log_{10}\left(1 + \frac{9.3}{\sqrt{8}k_s^+}\right) \tag{2.29}$$

Colebrook plotted experimental friction factor data in the same form as Fig. 2.2 as shown in Fig. 2.3. From Fig. 2.3, it is obvious that the data taken by Nikuradse do not match the data obtained from Colebrook in the transition region. Commercial pipes have a roughness that is uneven both in size and spacing and do not correspond to the pipes artificially roughened by Nikuradse.

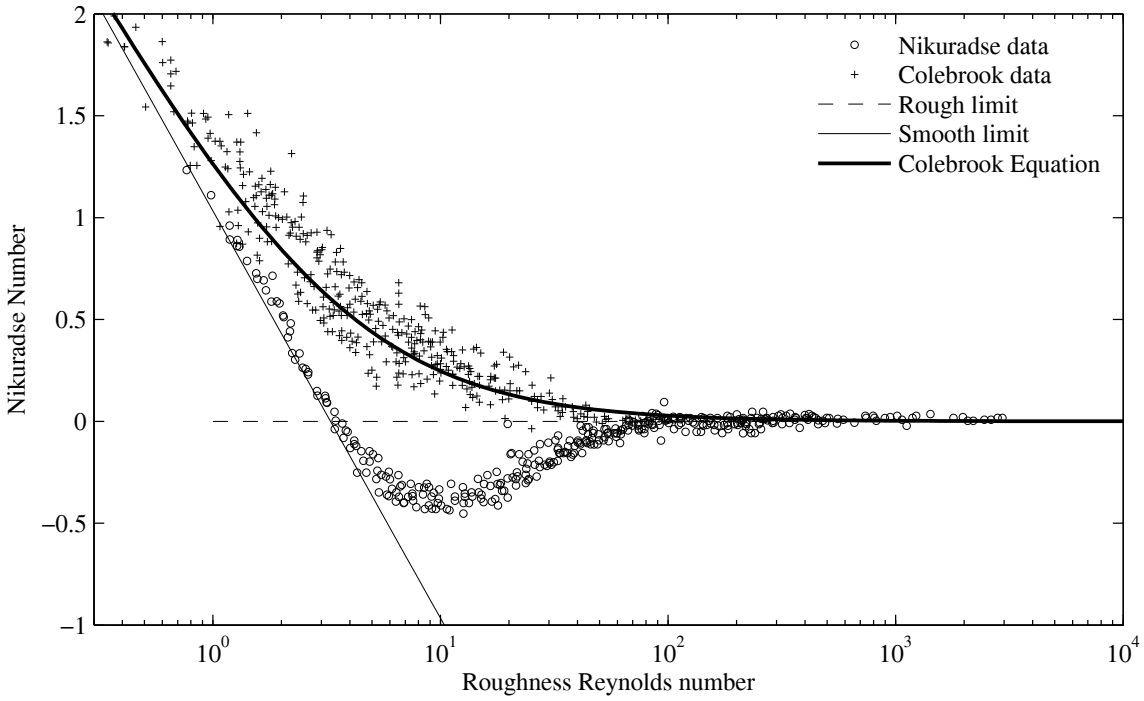

Fig. 2.3 Nikuradse number for commercial and artificially roughened pipes.

Initially Colebrook carried out experiments to determine the effect of non-uniform roughness as found on commercial pipe. Colebrook showed that the transition from hydraulically smooth to rough was in fact gradual. An artificially roughened surface is characterized by a very low standard deviation, which indicates that the roughness height of each individual element tends to be very close to the mean roughness height, whereas a commercial pipe is expected to have a much higher standard deviation, indicating that the

roughness height is spread out over a wider range. In the transition region, the thickness of the laminar sublayer is of the order of the roughness element height. Elements that extend through the laminar sublayer will tend to trigger the transition from hydraulically smooth to rough at a lower roughness Reynolds number for a surface with a high standard deviation than for a surface with a low standard deviation. Nikuradse defined the transition region as $5.0 \le k_s^+ \le 70.0$. For commercial pipes, this region should be adjusted to $0.2 \le k_s^+ \le 100.0$.

Moody [66] presented a diagram to evaluate the friction factor of commercial pipe based on the Colebrook equation given in Eq. (2.28) which has been extensively used for practical applications. The Moody diagram is the graphical solution of the Colebrook equation. Because of Moody's work and the demonstrated applicability of the Colebrook equation over a wide range of Reynolds numbers and relative roughness, Eq. (2.28) has become the accepted standard for calculating the friction factors. It suffers however from being an implicit equation and thus requires an iterative solution. A very accurate initial estimate can be obtained from the explicit relation

$$4C_f = \left\{ -1.8 \log_{10} \left[\left(\frac{k_r}{3.7} \right)^{1.11} + \frac{6.9}{\mathrm{Re}} \right] \right\}^{-2} \tag{2.30}$$

Moody [66] used the relation developed by Colebrook [65] to generate the well known Moody chart; he used the symbol ε to denote Nikuradse's sand-grain roughness. In this dissertation, we will continue to use k_s to signify the equivalent sand-grain roughness. The Moody chart is presented in Fig. 2.4 along with Nikurdase's experimental data on hydraulically smooth and rough pipes [47] as well as Shockling's experimental data [72]. Shockling et al. [72] carried out experiments for fully developed turbulent pipe flow over a wide range of Reynolds numbers where the flow exhibits hydraulically smooth, transitionally rough and fully rough behaviors. The surface of the pipe was prepared with a honing tool. As can be seen on Fig. 2.4 in the transitionally rough regime, the friction factor follows the same inflection point pattern as Nikuradse [47] observed, rather than the monotonic pattern Colebrook obtained from his commercial pipe experiments [65]. More recent experiments carried on prepared pipes support this inflection point pattern. See for example the work of Ligrani and Moffat [73] and Perry et al. [74].

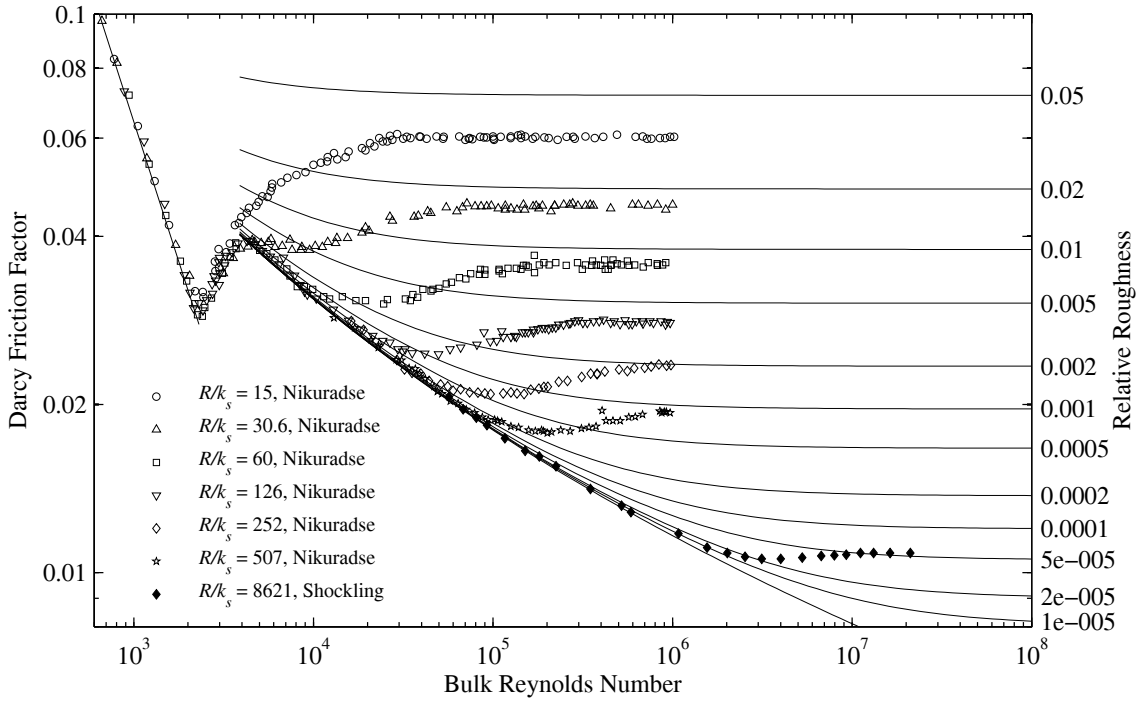

Fig. 2.4 The Moody diagram compared with experimental data from Nikuradse and Shockling.

In the transition region, the Moody diagram should be used with caution and the uniformity of the surface should be taken in consideration when evaluating the friction factor. Part of the difficulty in comparing roughness functions in the transitionally rough regime is that the roughness height is not well defined. It seems obvious that an arbitrary surface would need more than one characteristic scale to describe its effects on the near-wall flow (such as at least an equivalent sand-grain roughness and a standard deviation). Nevertheless, in order to compare different type of surfaces, it is usual to prescribe an equivalent sand-grain roughness that relates the root mean square roughness height for a given surface to a particular sand-grain roughness height. The equivalent sand-grain roughness is found by comparing the friction factor of the surface in question to Nikuradse's sand grain data in the fully rough regime, independent from the particular form of the roughness function in the transitional rough regime. As an example, Hama [75] suggests that for a machined surface with an approximately Gaussian distribution of roughness elements the equivalent sand grain roughness is five times the root mean square roughness height.

The most widely accepted correlation for the friction factor in pipes is based on Nikuradse and Colebrook empirical correlations. Nikuradse's work on fully rough limit has set a reference for estimating

the equivalent sand-grain roughness that characterized the pipe geometry. The validity of Nikuradse's fully rough limit given in Eq. (2.19) is so widely accepted that it has become the definition of the surface roughness. Any turbulence model capable of predicting rough flow behavior should yield a friction factor coefficient that matches the experimental data obtained by either Colebrook [65], Nikuradse [47] or Shockling et al. [72] in the fully rough region.

D. Mixing-Length Theory

Prandtl [5] recognized that the turbulent eddy viscosity has units of length times velocity and suggested that the turbulent eddy viscosity was proportional to the product of a distance, referred to as the mixing length ℓ, and a turbulent vertical velocity scale. Analogous to the mean free path of a molecule, Prandtl assumed that the fluid element conserves its properties over the characteristic length ℓ before being mixed with the surrounding fluid. Prandtl suggested that turbulent vertical motions were caused by the collision of fluid elements moving horizontally at different speeds. This results in a turbulent vertical velocity being proportional to a turbulent horizontal velocity. Prandtl suggested to use $\ell\left|\frac{d\overline{V}_z}{dy}\right|$ as a turbulent velocity scale. From Prandtl's mixing length theory, the turbulent eddy viscosity is equal to the product of the mixing length squared and the absolute value of the velocity gradient.

$$\nu_t = \ell^2\left|\frac{d\overline{V}_z}{dy}\right| \tag{2.31}$$

The turbulent eddy viscosity given in Eq. (2.31) can be non-dimensionalized in terms of the wall-scaled dimensionless variables u^+ and y^+ given in Eq. (2.3) to yield

$$\nu^+ = \left(\frac{u_\tau \ell}{\nu}\right)^2\left|\frac{du^+}{dy^+}\right| \tag{2.32}$$

Substituting the dimensionless form of the eddy viscosity given in Eq. (2.32) into the wall-scaled Boussinesq-RANS formulation formulated in Eq. (2.4) yields

$$\left(1+\left(\frac{u_\tau \ell}{\nu}\right)^2\left|\frac{du^+}{dy^+}\right|\right)\frac{du^+}{dy^+}=1-\frac{y^+}{R_\tau}\ , \qquad u^+\Big|_{y^+=0}=0 \tag{2.33}$$

An exact analytical solution to Eq. (2.33) is not available, and accurate numerical solutions were not practical when the foundation for modeling rough wall turbulent flow was laid. Van Driest [76] proposed a near-wall empirical relation for the mixing length that matches experimental data very well. Using the non-dimensional turbulent eddy viscosity given in Eq. (2.32) in the near-wall fully rough limit formulation given in Eq. (2.6) yields an important nondimensional relation between Prandtl's mixing length and the mean velocity gradient, which applies to the near-wall fully rough limit

$$\frac{u_\tau \ell}{\nu}\frac{du^+}{dy^+}=1\,, \qquad u^+\Big|_{y^+=0}=0 \tag{2.34}$$

Prandtl's mixing length cannot be measured directly. Hence, empirical correlations for the mixing length must be inferred from measurements of the mean velocity profile and the mean wall shear stress or pressure gradient. From measured mean velocity profiles, Nikuradse [47] used this procedure to calculate and plot the variation of mixing length along the wall coordinate *y/R*. In the fully rough limit near the pipe wall, the mixing-length profiles obtained by Nikuradse were found to vary linearly with the distance from the pipe wall. Furthermore, the proportionality constant between the mixing length ℓ and the wall coordinate y was shown to be independent of both the Reynolds number and the surface roughness. In this near-wall region, the results obtained by Nikuradse for the fully rough limit are in excellent agreement with the empirical correlation

$$\ell=\kappa\left(y+\gamma\, k_s\right) \tag{2.35}$$

where κ and γ are two empirical constants. The coefficient k_s is known as the roughness height first introduced by Nikuradse [47] who sifted sand into groups of known diameters and artificially roughened hydraulically smooth-wall pipes to determine the effect of surface roughness on the pressure drop of the fluid flow through the pipe. The constant κ is traditionally called the Von Kármán constant after Ludwig

Prandtl's student Theodore Von Kármán [77]. In a similar manner, the constant γ will be referred to as the Nikuradse constant.

E. Fully Rough Flow Velocity Profile

A sensitive indicator for the effect of roughness in a pipe is given by the behavior of the velocity profile. Nikuradse proposed a very simple law for the velocity distribution in rough pipes based on his experiments [47] that is similar to the well-known law of the wall first proposed by Von Kármán [77].

$$u^+ = 5.75 \log_{10}\left(\frac{y^+}{k_s^+}\right) + 8.48 \tag{2.36}$$

Using only two significant digits and the natural logarithm, Eq. (2.36) can be rewritten as

$$u^+ = 2.5 \ln\left(\frac{y^+}{k_s^+}\right) + 8.5 \tag{2.37}$$

Obviously, Eq. (2.38) cannot apply over the entire flow field from the wall to the centerline because it does not satisfy the no-slip boundary condition at the pipe wall or the symmetry boundary condition at the pipe centerline. However, the deviation from Eq. (2.38) was traditionally assumed to be sufficiently small so that the bulk velocity could be approximated from the integral of Eq. (2.38).

Using Eq. (2.35) in Eq. (2.34) and rearranging yields the near-wall fully rough limit based on Prandtl's mixing length theory

$$\frac{du^+}{dy^+} = \frac{1}{\kappa\left(y^+ + \gamma\, k_s^+\right)}, \qquad u^+\Big|_{y^+=0} = 0 \tag{2.38}$$

where the dimensionless parameter k_s^+, defined in Eq. (2.12) is commonly called the roughness Reynolds number. Equation (2.38) can be integrated easily subject to the no-slip wall boundary to yield

$$u^+ = \frac{1}{\kappa} \ln\left(\frac{y^+}{\gamma\, k_s^+} + 1\right) \tag{2.39}$$

The empirical constants κ and γ can be evaluated by comparing Eq. (2.37) and (2.39) in the core region. To make this comparison, we must recognize that the roughness elements are much smaller than the pipe diameter and the empirical constant γ is much less than unity. Hence, the wall coordinate y is large compared with the product $\gamma\, k_s$, except in an extremely thin layer near the pipe wall where y is on the order of k_s. With this knowledge, we can neglect the 1 in the natural log term of Eq. (2.39). The near-wall layer where this 1 is significant is much too thin to permit experimental measurements, so this layer was not captured in Nikuradse's data nor in the associated empirical correlation given by Eq. (2.37). Using the approximation $y >> \gamma\, k_s$, Eq. (2.39) can be written as

$$u^+ = \frac{1}{\kappa}\ln\left(\frac{y^+}{k_s^+}\right) + \frac{1}{\kappa}\ln\left(\frac{1}{\gamma}\right) \tag{2.40}$$

Comparing Eq. (2.37) with Eq. (2.40), we see that the velocity profile measurements of Nikuradse yield values for both the Von Kármán constant and the Nikuradse constant

$$\kappa = \frac{1}{2.5} = 0.400\,, \qquad \gamma = e^{-8.5\kappa} = 0.0334 \tag{2.41}$$

A comparison among Eq. (2.39), Eq. (2.37) and experimental data collected by Nikuradse is shown in Fig. 2.5. The dashed line is obtained from Eq. (2.39) whereas the solid line is obtained from Eq. (2.37) using κ = 0.4 and γ = 0.0334 as obtained from Eq. (2.41). Nikuradse's experimental velocities used to generate Fig. 2.5 are given in Appendix BI. Only the data with the highest roughness Reynolds number were used for each of the relative roughness presented in Fig. 2.5.

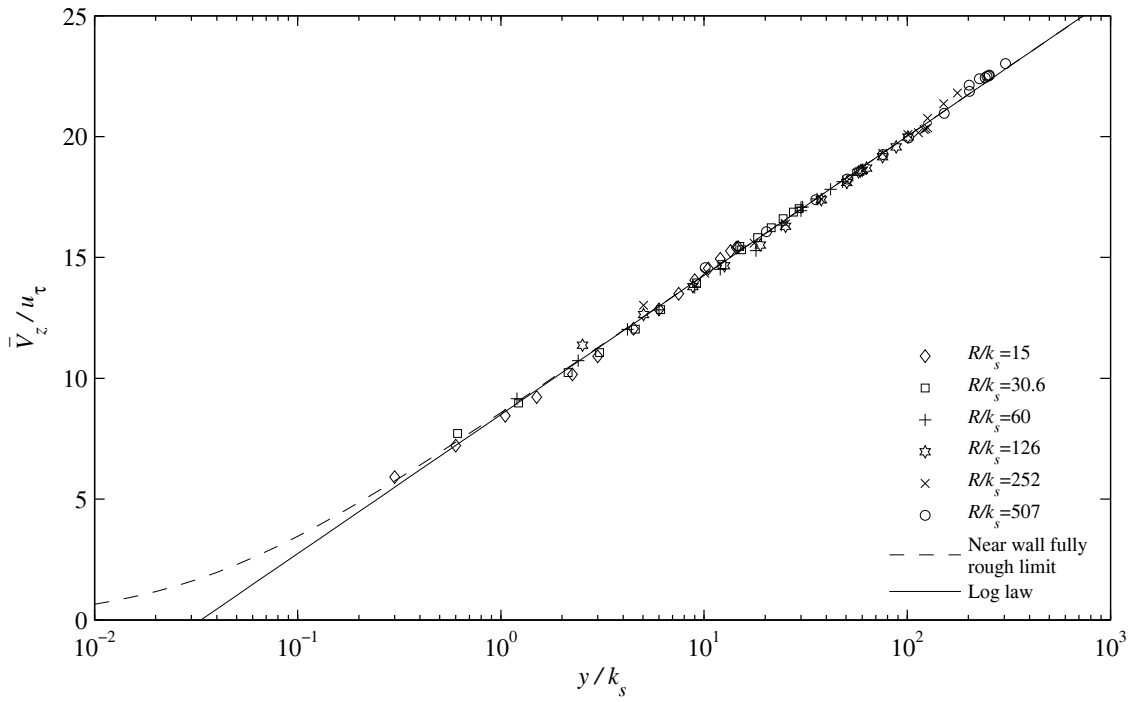

Fig. 2.5 Velocity profiles in rough pipes at high Reynolds numbers.

Using the near-wall fully rough limit for the velocity given in Eq. (2.39) in the bulk velocity defined in Eq. (2.9) results in

$$u_m^+ = \frac{2}{\kappa (U-1)^2} \int_{u=1}^{U} (U-u) \ln(u)\, du\,, \qquad u \equiv \frac{y}{\gamma k_s} + 1\,, \quad U \equiv \frac{R}{\gamma k_s} + 1 \tag{2.42}$$

After performing the indicated integration in the nondimensional bulk velocity, Eq. (2.42) yields

$$u_m^+ = \frac{1}{\kappa} \left(\frac{\gamma k_s}{R} \right)^2 \left\{ \left(\frac{R}{\gamma k_s} + 1 \right)^2 \left[\ln\left(\frac{R}{\gamma k_s} + 1 \right) - \frac{3}{2} \right] + 2 \frac{R}{\gamma k_s} + \frac{3}{2} \right\} \tag{2.43}$$

Because the pipe radius R is always very large compared with the roughness element height k_s, the sum $1 + R/(\gamma k_s)$ may be reduced to $R/(\gamma k_s)$. Schlichting [52] approximated the bulk velocity given in Eq. (2.43) to

$$u_m^+ \cong \frac{1}{\kappa}\left[\ln\left(\frac{R}{\gamma\, k_s}\right) - \frac{3}{2}\right] \quad \text{or} \quad u_m^+ \cong \frac{1}{\kappa}\ln\left(\frac{R}{k_s}\right) + \frac{1}{\kappa}\left[\ln\left(\frac{1}{\gamma}\right) - \frac{3}{2}\right] \tag{2.44}$$

Using the near-wall fully rough limit from Eq. (2.39) in the approximated bulk velocity given in Eq. (2.44), the Darcy friction factor for fully rough pipe flow is reduced to

$$4C_f = \frac{8}{\left(V_m / u_\tau\right)^2} = 8\left\{\frac{1}{\kappa}\ln\left(\frac{R}{k_s}\right) + \frac{1}{\kappa}\left[\ln\left(\frac{1}{\gamma}\right) - \frac{3}{2}\right]\right\}^{-2} \tag{2.45}$$

The relation given by Eq. (2.45) is commonly written in terms of the base-10 logarithm, i.e.,

$$4C_f = \left\{\frac{1}{8^{1/2}\kappa\log_{10}(e)}\log_{10}\left(\frac{R}{k_s}\right) + \frac{1}{8^{1/2}\kappa}\left[\ln\left(\frac{1}{\gamma}\right) - \frac{3}{2}\right]\right\}^{-2} \tag{2.46}$$

Using the values for κ and γ that are given in Eq. (2.41), the relations given in Eqs. (2.44) and (2.46) may be written as

$$u_m^+ = 2.50\ln\left(\frac{R}{k_s}\right) + 4.75 \tag{2.47}$$

and

$$4C_f = \left[2.04\log_{10}\left(\frac{R}{k_s}\right) + 1.68\right]^{-2} \tag{2.48}$$

which exhibits the expected quadratic relation between wall shear stress and bulk velocity for fully rough flow. Although based on the same experimental data, Eqs. (2.17) and (2.48) are slightly different. Proposed by Nikuradse, Eq. (2.17) was obtained graphically by plotting $1/\sqrt{4C_f}$ against $\log_{10}\left(R/k_s\right)$ and by fitting a line through the experimental data. Equation (2.48) was obtained based on the approximated bulk velocity and the definition of the friction factor.

Comparing the Nikuradse equation given in Eq. (2.19) with Eq. (2.46), the empirical constants κ and γ can be reevaluated. After rounding these constants to two significant digits, the result yields

$$\kappa = \frac{1}{2.0(8)^{1/2}\log_{10}(e)} = 0.41 \ , \qquad \gamma = \exp\left\{-2.0(8)^{1/2}\kappa\log_{10}[2(3.7)] - \frac{3}{2}\right\} = 0.030 \tag{2.49}$$

Although the values for κ and γ given in Eq. (2.49) are commonly used, it should be reemphasized that Eq. (2.46) was obtained from the approximate evaluation of the bulk velocity given by Eq. (2.44). This result is based on assuming that the near-wall velocity profile obtained from Eq. (2.39) applies over the entire flow field from the wall to the centerline. Nikuradse was forced to use this approximation because accurate solutions to the complete mixing-length formulation were not available in 1933. Today, accurate evaluation of the bulk velocity is possible using numerical integration of the complete mixing-length formulation obtained from Eq. (2.33). When the bulk velocity is evaluated in this manner and the empirical constants κ and γ are adjusted to minimize the difference between the numerical results for the fully rough limit and results obtained from Eq. (2.20), the resulting constants rounded to three significant digits are

$$\kappa = 0.403 \ , \quad \gamma = 0.0324 \tag{2.50}$$

Notice that these results are closer to those obtained from Nikuradse's velocity-profile measurements than they are to the results presented in Eq. (2.49).

Expressed equivalently in Eqs. (2.17) and (2.20), the Nikuradse equation provides an accurate means for predicting the Darcy friction factor when the Reynolds number is large enough so that the friction factor becomes independent of the molecular viscosity. However, the Nikuradse equation alone provides no information regarding how large the Reynolds number must be to make this empirical correlation valid. From Nikuradse data on artificially roughened pipes, it is commonly accepted that this correlation for fully rough flow is valid whenever the roughness Reynolds number is greater than about 70. However, by definition, the fully rough flow approximation is valid only when the molecular viscosity is negligible compared to the eddy viscosity throughout the flow field.

III. Fully Rough Flow Empirical Turbulent Eddy Viscosity

A. Turbulent Eddy Viscosity Profile Obtained from Prandtl's Mixing-Length Theory

Fully rough pipe flow is defined to be the asymptotic solution to the wall-scaled Boussinesq RANS formulation given in Eq. (2.4) in the limit as the nondimensional eddy viscosity ν^+ becomes large compared to unity throughout the flow field. Because the eddy viscosity ν^+ is smallest near the wall, the limit for application of the fully rough flow approximation can be evaluated by examining the near-wall behavior of ν^+ as predicted from the fully rough solution. From Prandtl's mixing-length theory, ν^+ is related to the wall-scaled dimensionless variables and the mixing length through Eq. (2.32). In the near-wall fully rough limit, the mixing length is given by the empirical correlation Eq. (2.35) and, based on this empirical correlation, the wall-scaled velocity gradient can be obtained from Eq. (2.38). Hence, substituting Eq. (2.35) for the mixing length and Eq. (2.38) for the velocity gradient into Eq. (2.32), we find that the ratio of the turbulent eddy viscosity ν_t to the molecular viscosity ν in the near-wall fully rough limit is given by

$$\nu^+ \equiv \frac{\nu_t}{\nu} = \kappa\left(y^+ + \gamma\, k_s^+\right) \tag{2.51}$$

The empirical constants κ and γ obtained from the Nikuradse equation as expressed in Eq. (2.20) are given in Eq. (2.50). Therefore, the near-wall eddy viscosity relation that is consistent with the Nikuradse equation is

$$\nu^+\left(y^+, k_s^+\right) \equiv \frac{\nu_t}{\nu} = 0.41\, y^+ + 0.012\, k_s^+ \tag{2.52}$$

From Eq. (2.51), we see that when the roughness Reynolds number k_s^+ is 70, the ratio of the turbulent eddy viscosity at the wall to the molecular viscosity is predicted to be $\nu^+ = 0.84$, which is obviously not large compared to unity. Clearly, we should not necessarily expect the general solution to the wall-scaled Boussinesq RANS formulation expressed in Eq. (2.4) to match the fully rough asymptote when the roughness Reynolds number k_s^+ is 70. At a roughness Reynolds number of $k_s^+ = 1000$, Eq. (2.51)

evaluated at the wall yields $\nu^+ = 12$, which is a more reasonable criterion for applying the fully rough flow approximation $\nu^+ >> 1$. The molecular viscosity is not reduced to 1 percent of the turbulent eddy viscosity at the wall, until a roughness Reynolds number k_s^+ of more than 8000 is reached.

However, it was previously shown that the molecular viscosity agrees closely with experimental data when the roughness Reynolds number k_s^+ is 70 and above. In fact, the fully rough flow approximation not only neglects the molecular viscosity ν compared with the turbulent eddy viscosity ν_t, but it also neglects the existence of a viscous sublayer. Because including the molecular viscosity ν in the turbulent portion of the flow tends to increase the friction factor C_f, and the presence of the viscous sublayer tends to decrease this friction factor C_f, the tradeoff between these two opposing effects tends to make the Nikuradse equation agree with experimental data when the roughness Reynolds number is lower than would be expected based on only a comparison of the eddy viscosity at the wall with the molecular viscosity.

A near-wall fully rough limit may be obtained for the mixing length. From Nikuradse's experimental data for the friction factor in artificially roughened pipes, it has been shown that Prandtl's mixing length in the near-wall fully rough limit is given by Eq. (2.35). Furthermore, the empirical constants κ and γ can be evaluated directly from Nikuradse's friction-factor data. Dividing Eq. (2.35) through by the pipe radius *R*, in the near-wall fully rough limit we have

$$\frac{\ell}{R} = \kappa\left(\frac{y}{R} + \gamma\frac{k_s}{R}\right) \tag{2.53}$$

This empirical correlation applies only near the pipe wall where the ratio y/R is much less than unity. Hence, it should not be used over the majority of the pipe diameter where curvature is important.

Prandtl [5] suggested an empirical correlation for the mixing length for smooth-wall pipes. Nikuradse collected data on mean-velocity profiles in both hydraulically smooth [46] and artificially roughened pipes [47]. The local mean velocity was measured in radial increments from the pipe centerline to a point near the wall where $y/R = 0.02$. From these measured velocity profiles, Nikuradse calculated and plotted

the variation of Prandtl's mixing length with the ratio y/R and found that the mixing length distribution could be represented by the same curve Prandtl suggested for smooth-wall pipes [5]

$$\frac{\ell}{R} = 0.14 - 0.08\left(1 - \frac{y}{R}\right)^2 - 0.06\left(1 - \frac{y}{R}\right)^4 \tag{2.54}$$

Within experimental uncertainty, there exists the same mixing-length distribution in rough pipes as in hydraulically smooth-wall pipes. This leads to the conclusion that the mechanics of turbulence are independent of the type of wall surface except in a very thin near-wall layer.

A more general correlation for the mixing-length can be developed to encompass both the near-wall limit given by Eq. (2.53) and the mixing-length distribution obtained in the core region and given by Eq. (2.54). A general form for the empirical correlation proposed in Eq. (2.54) is given by

$$\frac{\ell}{R} = \kappa\left[A_0 - A_2\left(\frac{r}{R}\right)^2 - A_3\left(\frac{r}{R}\right)^4\right] \tag{2.55}$$

The empirical correlation given by Eq. (2.55) is not consistent with that given by Eq. (2.53), because Eq. (2.55) shows no dependence on surface roughness in the near-wall region. However, near the pipe centerlines, Nikuradse's velocity-profile data show a mean value for ℓ/R of about 0.138 and scatter in these data indicate an uncertainty on the order of ±0.004. At the points nearest the pipe walls, Nikuradse's data show a value for ℓ/R on the order of 0.008 with an indicated uncertainty on the order of ±0.002. The roughest pipe for which Nikuradse obtained velocity-profile data had an inverse roughness ratio of $R/k_s = 15.0$. At this inverse roughness ratio, the value of the second term on the right-hand side of Eq. (2.53) is approximately $\kappa\gamma k_s/R = 0.0008$. Because this value is less than one half the uncertainty indicated by the scatter in Nikuradse's mixing-length-profile data, the second term on the right-hand side of Eq. (2.53) $\kappa\gamma k_s/R$ could not have been captured in the mixing-length-profile data obtained by Nikuradse. Combining results obtained from Nikuradse's velocity-profile data with results obtained from his friction-factor data suggests a fairly general relation for fully rough pipe flow, which could be used over the entire pipe diameter

$$\frac{\ell}{R} = \kappa\left[A_0 + \gamma\frac{k_s}{R} + A_1\left(1-\frac{r}{R}\right)^2 - A_2\left(\frac{r}{R}\right)^2 - A_3\left(\frac{r}{R}\right)^4 \right]\left(\frac{r}{R}\right)^a \tag{2.56}$$

Note that Eq. (2.55) is a special case of Eq. (2.56) for which $k_s/R = 0$, $A_1 = 0$ and $a = 0$. A near-wall expansion of Eq. (2.56) gives

$$\frac{\ell}{R} = \kappa\left\{ A_0 - A_2 - A_3 + \gamma\frac{k_s}{R} + \left[(2+a)A_2 + (4+a)A_3 - aA_0 - a\gamma\frac{k_s}{R} \right]\frac{y}{R} + \ldots \right\} \tag{2.57}$$

Comparing Eq. (2.57) with the near-wall limit given by Eq. (2.53) requires

$$A_2 = 2A_0 - 0.5\left(1 - a\gamma\frac{k_s}{R}\right) \tag{2.58}$$

and

$$A_3 = 0.5\left(1 + a\gamma\frac{k_s}{R}\right) - A_0 \tag{2.59}$$

Therefore, from Eq. (2.56), Prandtl's mixing-length profile over the entire pipe diameter could be written for fully rough pipe flow in terms of only the two empirical coefficients A_0 and a

$$\frac{\ell}{R} = \kappa\left[\begin{array}{l} A_0 + \gamma\dfrac{k_s}{R} + A_1\left(\dfrac{y}{R}\right)^2 - \left(2A_0 - \dfrac{1}{2} - \dfrac{a\gamma}{2}\dfrac{k_s}{R}\right)\left(1-\dfrac{y}{R}\right)^2 \\ -\left(\dfrac{1}{2} + \dfrac{a\gamma}{2}\dfrac{k_s}{R} - A_0\right)\left(1-\dfrac{y}{R}\right)^4 \end{array} \right]\left(1-\frac{y}{R}\right)^a \tag{2.60}$$

An empirical fit to Nikuradse's mixing-length-profile data can be obtained using $A_1 = 0$ and $a = 0$. The empirical constant A_0 is found to be approximately 0.345 to within an indicated uncertainty on the order of ±0.01. A comparison among Eq. (2.53), Eq. (2.60) and data obtained from Nikuradse is shown in Fig. 2.6. The dashed line is obtained from Eq. (2.53). The solid line is obtained from Eq. (2.60). The mixing-length empirical data were obtained from Nikuradse's tabulated velocity gradient given in Appendix B. Experimental research by Anderson et al. [78] has shown that the turbulent mixing length

varies linearly in the near-wall region of the flow. This agrees with Nikuradse's experimental data shown in Fig. 2.6.

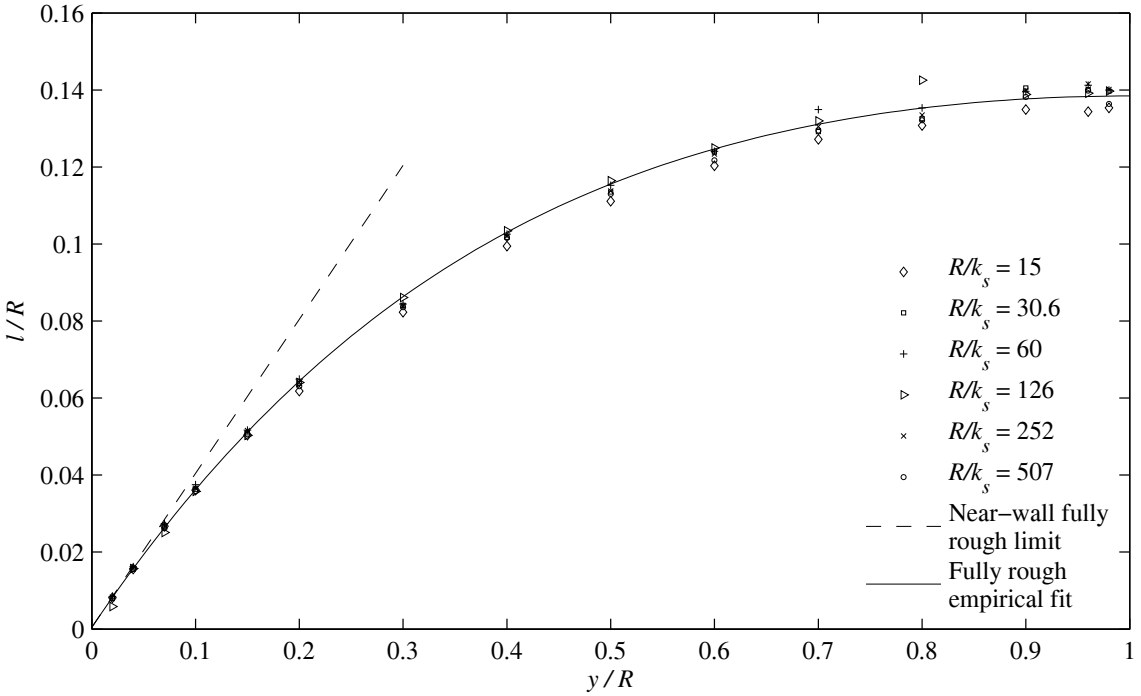

Fig. 2.6 Mixing-length profiles in rough pipes at high Reynolds numbers.

The velocity distribution may be deduced from the mixing lengths empirical correlation. Starting with the Boussinesq-RANS formulation expressed in Eq. (2.4), employing the fully rough flow approximation to eliminate the molecular viscosity ν in comparison with the turbulent eddy viscosity ν_t, and writing the result in terms of the wall coordinate y gives

$$\frac{\nu_t}{u_\tau R}\frac{du^+}{d\hat{y}} = 1 - \hat{y} \quad , \qquad u^+\Big|_{\hat{y}=0} = 0 \tag{2.61}$$

Using the turbulent eddy viscosity expressed in Eq. (2.31) in the Boussinesq-RANS equations expressed in Eq. (2.62) yields

$$\frac{\ell}{R}\frac{du^+}{d\hat{y}} = \sqrt{1-\hat{y}} \quad , \qquad u^+\Big|_{\hat{y}=0} = 0 \tag{2.62}$$

Applying Eq. (2.56) to Eq. (2.62) results in a complete mixing-length formulation for fully rough pipe flow

$$\frac{du^+}{d\hat{y}} = \frac{\left(1-\hat{y}\right)^{0.5-a}}{\kappa\left[A_0 + \gamma\frac{k_s}{R} + A_1\hat{y}^2 - A_2\left(1-\hat{y}\right)^2 . - A_3\left(1-\hat{y}\right)^4\right]}, \qquad u^+\Big|_{y^+=0} = 0 \tag{2.63}$$

Using Eq. (2.63) in Eq. (2.61), the turbulent eddy viscosity profile over the entire pipe diameter as predicted from Prandtl's mixing-length theory could be written for fully rough pipe flow as

$$\frac{\nu_t}{u_\tau R} = \kappa\left[\begin{array}{l} A_0 + \gamma\frac{k_s}{R} + A_1\left(\frac{y}{R}\right)^2 - \left(2A_0 - \frac{1}{2} - \frac{a\gamma}{2}\frac{k_s}{R}\right)\left(1-\frac{y}{R}\right)^2 \\ -\left(\frac{1}{2} + \frac{a\gamma}{2}\frac{k_s}{R} - A_0\right)\left(1-\frac{y}{R}\right)^4 \end{array}\right]\left(1-\frac{y}{R}\right)^{0.5+a} \tag{2.64}$$

A comparison between Nikuradse's experimental data and Eq. (2.64) with $A_0 = 0.345$, $A_1 = 0$ and $a = 0$ is shown in Fig. 2.7. Notice that in the limit as y/R approaches zero, Eq. (2.64) approaches the linear asymptote

$$\frac{\nu_t}{u_\tau R} = \kappa\left[\gamma\frac{k_s}{R} + \left(1 - \frac{\gamma}{2}\frac{k_s}{R}\right)\frac{y}{R}\right], \qquad \frac{y}{R} \to 0 \tag{2.65}$$

Prandtl's mixing-length theory is in excellent agreement with experimental data in the near-wall region but fails to predict the turbulent eddy viscosity accurately in the centerline region. From Eq. (2.31) combined with the symmetry boundary condition at the centerline, we see that mixing-length theory always predicts $\nu_t = 0$ at the pipe centerline, which seems implausible. In fact, the symmetry boundary condition at the pipe centerline requires that the change in ν_t with respect to the radial coordinate r must be zero at the pipe centerline, whereas Eq. (2.64) with $A_1 = 0$ and $a = 0$ predicts an infinite value for this derivative. Hence, Eq. (2.64) with $A_1 = 0$ and $a = 0$ cannot be used in the core region near the center of the pipe.

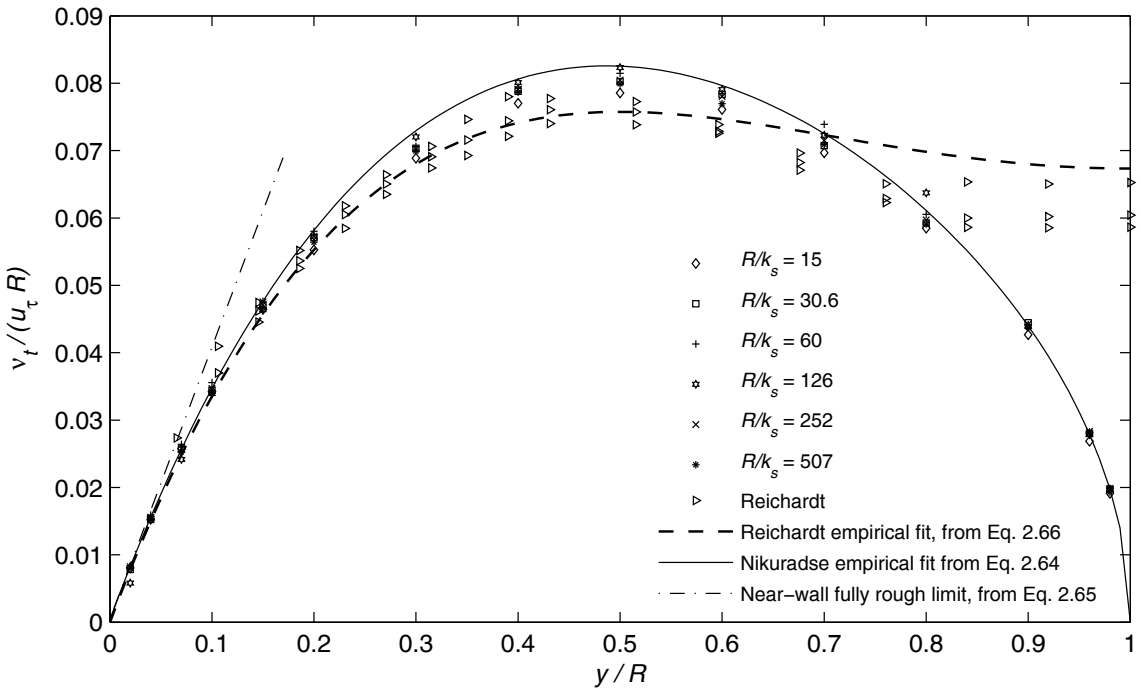

Fig. 2.7 Eddy viscosity profiles in hydraulically smooth- and rough-wall pipes at high Reynolds number.

Experimental data taken by Reichardt [79] on hydraulically smooth pipes suggest that the turbulent eddy viscosity may remain nearly constant in the central core of the pipe. From these data Reichardt [80] proposed the empirical correlation

$$\frac{\nu_t}{u_\tau R} = \frac{\kappa}{6}\frac{y}{R}\left(2-\frac{y}{R}\right)\left[1+2\left(1-\frac{y}{R}\right)^2\right] \tag{2.66}$$

which remains fairly constant from about $y/R = 0.3$ to the pipe centerline. Notice that as and $y/R \to 0$, Eq. (2.66) yields $\frac{\nu_t}{u_\tau R} \to \kappa\frac{y}{R}$, which is the same result obtained from Eq. (2.65) when $k_s/R = 0$. Reichardt's empirical correlation for the turbulent eddy viscosity in circular pipes with hydraulically smooth walls is easily modified to agree with the well established near-wall behavior for the fully rough limit, which comes from the empirical correlation expressed in Eq. (2.64)

$$\frac{\nu_t}{u_\tau R} = \frac{\kappa}{6}\left(\frac{y}{R} + \frac{\gamma k_s}{R}\right)\left(2 - \frac{y}{R}\right)\left[1 + 2\left(1 - \frac{y}{R}\right)^2\right] \tag{2.67}$$

Because $\gamma k_s / R << R$, for all practical purposes Eqs. (2.67) and (2.66) are identical, except in a thin layer near the pipe wall where y is on the order of k_s.

The velocity distribution may be obtained from the near-wall limit given in Eq. (2.39). As seen in Fig. 2.5, Eq. (2.39) provides a good fit to the mean-velocity-profile data collected by Nikuradse in the fully rough limit and it satisfies the no-slip boundary condition at the pipe wall. Reichardt's velocity-profile data are also in good agreement with this empirical correlation. Equation (2.39) is easily rearranged to give

$$u^+ = \frac{1}{\kappa}\ln\left(1 + \frac{R}{\gamma k_s}\frac{y}{R}\right) \tag{2.68}$$

Differentiating Eq. (2.67) and substituting in the Boussinesq-RANS formulation results in

$$\frac{\nu_t}{u_\tau R} = \kappa\left(\gamma\frac{k_s}{R} + \frac{y}{R}\right)\left(1 - \frac{y}{R}\right) \tag{2.69}$$

Results obtained from Eq. (2.69) are also shown in Fig. 2.7. Clearly, Eq. (2.69) cannot be applied over the entire flow field from the wall to the pipe centerline, because the symmetry boundary condition at the centerline is not satisfied. The same can be said of Eq. (2.65).

As seen in Fig. 2.5, it is important to notice that there is a very significant difference between the correlations given by Eq. (2.64) and Eq. (2.69), and it is important to recall that these two correlations came from exactly the same experimental data. For each pipe and Reynolds number, Nikuradse measured the local mean velocity in radial increments from the pipe centerline to a point near the wall where y/R=0.02. From these measurements he tabulated both the local mean velocity and the local mean-velocity gradient as obtained from the mean velocity at neighboring points. The correlation given by Eq. (2.64) was obtained directly from Nikuradse's tabulated velocity gradient profiles. On the other hand, the correlation given by Eq. (2.69) was obtained from the analytical derivative of Eq. (2.39), which is an empirical correlation

obtained from Nikuradse's tabulated velocity profiles. Clearly, the mean velocity profiles and the mean-velocity gradient profiles tabulated by Nikuradse are inconsistent.

It is not surprising to learn that there is considerable uncertainty associated Nikuradse's tabulated velocity gradients near the pipe centerline, because turbulent mean-velocity gradients are extremely small near the center of a pipe at high Reynolds numbers. For example, in Nikuradse's tabulated velocity profiles, the velocity changes over the three points nearest the pipe centerlines are on the order of 3 to 4 parts per 1000. If central difference were used to extract the velocity gradients from these measured velocities and the experimental uncertainty in the velocity data was ±0.2%, the uncertainty in the estimated velocity gradients would be on the order of ±100%. Because the eddy viscosity values obtained from Nikuradse's data and shown in Fig. 2.5 are determined directly from the estimated velocity gradients, the uncertainty in these values must be at least ±100% near the pipe centerlines. This same level of uncertainty can be attributed to the eddy-viscosity values obtained from Reichardt's data and shown in Fig. 2.5.

This experimental uncertainty could easily explain the difference between Nikuradse's and Reichardt's results shown in Fig. 2.5. About all that can be said concerning the eddy viscosity at the pipe centerline based on the experimental results shown in Fig. 2.5 is that $\nu_t / u_\tau R$ most likely falls somewhere between 0.0 and about 0.1.

To estimate the eddy viscosity near the pipe centerline more accurately, we could correlate velocity-profile data over the entire pipe diameter and estimate the eddy-viscosity profile from the derivative of the correlation equation. This eliminates amplification of experimental uncertainty that results from numerical differentiation. The empirical correlation traditionally used for this purpose is Eq. (2.37), which is compared with Nikuradse's velocity-profile data in Fig. 2.5. However, Eq. (2.37) does not satisfy the symmetry condition at the pipe centerline or the no-slip condition at the wall. As seen in Fig. 2.5, Eq. (2.39) does satisfy the no-slip condition at the pipe wall and it agrees with Nikuradse's velocity-profile data as well as Eq. (2.37). Even so, Eq. (2.39) still does not satisfy the symmetry condition at the pipe centerline.

B. Corrective Function to Nikuradse's Experimental Velocity Data

A thorough analysis of Nikuradse's experimental data shows a regular deviation from the log law. When Nikuradse's velocity-profile data are plotted with $\overline{V}_z / u_\tau$ as a function of y / k_s as shown in

Fig. 2.1, the correlation with the near-wall fully rough limit given in Eq. (2.39) appears to be quite accurate, and the deviation from Eq. (2.39) appears to be quite random. However, when exactly the same data are plotted with $\overline{V}_z / V_m$ as a function of y/R as shown in Fig. 2.8, it can be seen that the deviation from Eq. (2.39) is not entirely random. The solid lines shown in Fig. 2.8 are obtained from Eq. (2.39) by dividing through by its bulk velocity as obtained from Eq. (2.43), i.e.,

$$\frac{\overline{V}_z}{V_m} = \frac{\overline{V}_z / u_\tau}{V_m / u_\tau} = \frac{\left(\frac{R}{\gamma k_s}\right)^2 \ln\left(\frac{R}{\gamma k_s}\frac{y}{R}+1\right)}{\left(\frac{R}{\gamma k_s}+1\right)^2\left[\ln\left(\frac{R}{\gamma k_s}+1\right)-\frac{3}{2}\right]+2\frac{R}{\gamma k_s}+\frac{3}{2}} \tag{2.70}$$

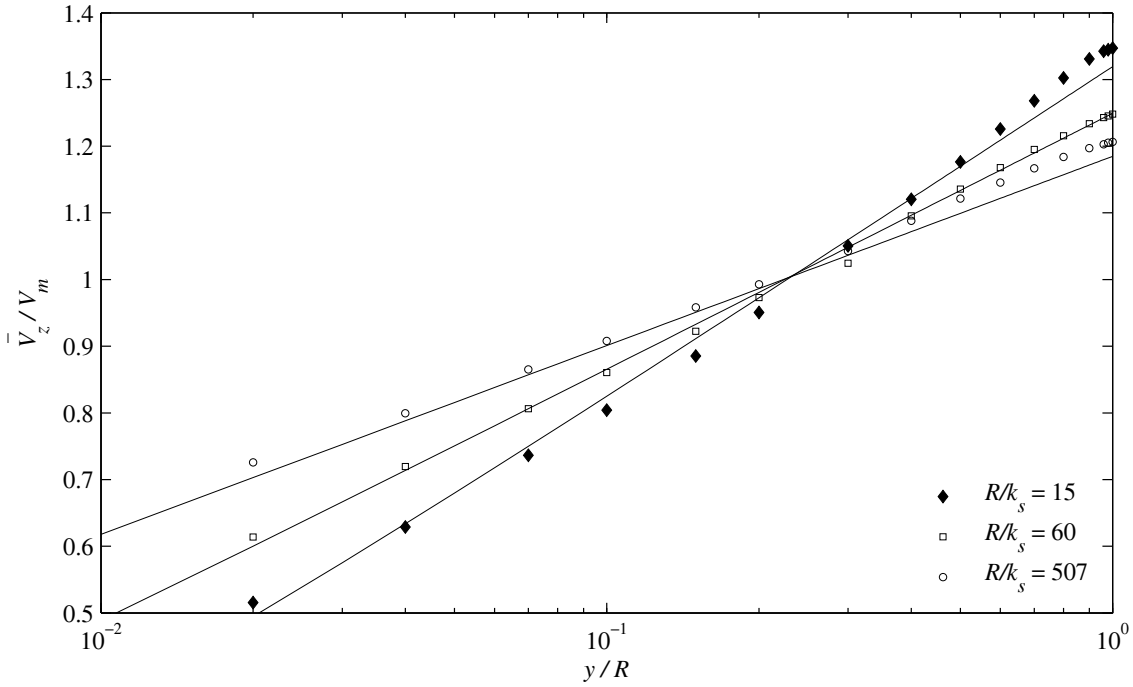

Fig. 2.8 Radial velocity profiles in rough pipes at high Reynolds numbers.

To satisfy the symmetry condition at the centerline and provide better agreement with Nikuradse's velocity profile data [47], an empirical correction function could be added to Eq. (2.70) expressed in terms of the relative coordinates defined in Eqs. (2.3) and (2.13)

$$\hat{u}\left(\hat{y},\hat{R}_s\right)=\frac{\hat{R}_s{}^2\ln\left(\hat{R}_s\hat{y}+1\right)}{\left(\hat{R}_s+1\right)^2\left[\ln\left(\hat{R}_s+1\right)-\frac{3}{2}\right]+2\hat{R}_s+\frac{3}{2}}+\delta\left(\hat{y},\hat{R}_s\right) \tag{2.71}$$

C. Constraints Imposed on the Corrective δ Function

In addition to correlating Nikuradse's velocity-profile data, mathematics and physics imposes several constraints on the empirical function δ defined in Eq. (2.72). The symmetry boundary condition at the pipe centerline requires

$$\left.\frac{d\delta}{d\hat{y}}\right|_{\hat{y}=1}=\frac{-\hat{R}_s{}^3}{\left(\hat{R}_s+1\right)\left\{\left(\hat{R}_s+1\right)^2\left[\ln\left(\hat{R}_s+1\right)-\frac{3}{2}\right]+2\hat{R}_s+\frac{3}{2}\right\}} \tag{2.72}$$

Similarly, the no-slip boundary condition at the pipe wall demands

$$\left.\delta\right|_{\hat{y}=0}=0 \tag{2.73}$$

Because Eq. (2.70) exhibits the correct near-wall behavior as expressed in Eq. (2.38), for the fully rough limit we must also enforce the relation

$$\left.\frac{d\delta}{d\hat{y}}\right|_{\hat{y}=0}=0 \tag{2.74}$$

To maintain the definition of bulk velocity, the integral of the product $\hat{u}\,\hat{r}$ from the centerline to the pipe wall must be unity. Hence, the empirical function δ defined in Eq. (28) must satisfy the relation

$$\int_{\hat{y}=0}^{1}\delta\left(1-\hat{y}\right)d\hat{y}=0 \tag{2.75}$$

A final constraint on the empirical function δ comes from applying the symmetry boundary condition on the eddy viscosity at the pipe centerline. From the fully rough Boussinesq-RANS equations, the eddy-viscosity profile is related to the velocity profile according to

$$\hat{\nu}_t = \frac{\frac{u_\tau}{V_m}(1-\hat{y})}{\frac{d\hat{u}}{d\hat{y}}} \tag{2.76}$$

and

$$\frac{d\hat{\nu}_t}{d\hat{y}} = \frac{\frac{u_\tau}{V_m}\left[-\frac{d\hat{u}}{d\hat{y}} - (1-\hat{y})\frac{d^2\hat{u}}{d\hat{y}^2}\right]}{\left(\frac{d\hat{u}}{d\hat{y}}\right)^2} \tag{2.77}$$

Because the velocity profile is also symmetric about the pipe centerline, Eq. (2.77) is indeterminate at the centerline. Applying l'Hospital's rule to Eq. (2.77) yields

$$\left.\frac{d\hat{\nu}_t}{d\hat{y}}\right|_{\hat{y}\to 1} = \frac{\frac{u_\tau}{V_m}\left[-(1-\hat{y})\frac{d^3\hat{u}}{d\hat{y}^3}\right]}{2\left(\frac{d\hat{u}}{d\hat{y}}\frac{d^2\hat{u}}{d\hat{y}^2}\right)} \tag{2.78}$$

which is also indeterminate. Applying l'Hospital's rule a second time results in

$$\left.\frac{d\hat{\nu}_t}{d\hat{y}}\right|_{\hat{y}\to 1} = \frac{\frac{u_\tau}{V_m}\left[\frac{d^3\hat{u}}{d\hat{y}^3} - (1-\hat{y})\frac{d^4\hat{u}}{d\hat{y}^4}\right]}{2\left(\frac{d^2\hat{u}}{d\hat{y}^2}\right)^2 + 2\frac{d\hat{u}}{d\hat{y}}\frac{d^3\hat{u}}{d\hat{y}^3}} \tag{2.79}$$

Because the velocity profile is symmetric about the pipe centerline, Eq. (2.79) reduces to

$$\left.\frac{d\hat{\nu}_t}{d\hat{y}}\right|_{\hat{y}\to 1} = \frac{\frac{u_\tau}{2V_m}\frac{d^3\hat{u}}{d\hat{y}^3}}{\left(\frac{d^2\hat{u}}{d\hat{y}^2}\right)^2} \tag{2.80}$$

Hence, because the eddy-viscosity profile must be symmetric about the pipe centerline, Eq. (2.80) provides another constraint on the velocity profile, i.e.,

$$\left.\frac{d^3\hat{u}}{d\hat{y}^3}\right|_{\hat{y}=1} = 0 \tag{2.81}$$

Using the corrected nondimensional velocity given in Eq. (2.71) in Eq. (2.81) provides an additional mathematical constraint on the empirical function δ,

$$\frac{d^3\hat{u}}{d\hat{y}^3} = \frac{2\hat{R}_s^{\,5}}{\left(\hat{R}_s\hat{y}+1\right)^3\left\{\left(\hat{R}_s+1\right)^2\left[\ln\left(\hat{R}_s+1\right)-\frac{3}{2}\right]+2\hat{R}_s+\frac{3}{2}\right\}} + \frac{d^3\delta}{d\hat{y}^3} \tag{2.82}$$

In terms of the δ function

$$\left.\frac{d^3\delta}{d\hat{y}^3}\right|_{\hat{y}=1} = \frac{-2\hat{R}_s^{\,5}}{\left(\hat{R}_s+1\right)^3\left\{\left(\hat{R}_s+1\right)^2\left[\ln\left(\hat{R}_s+1\right)-\frac{3}{2}\right]+2\hat{R}_s+\frac{3}{2}\right\}} \tag{2.83}$$

In view of the definition of the bulk velocity from Eq. (2.9) and Eq. (2.75), the empirical function δ will have no effect on the bulk velocity. Thus, the bulk velocity associated with Eq. (2.71) will be exactly the same as that associated with Eq. (2.39). The dimensionless wall-scaled bulk velocity for fully rough pipe flow given by Eq. (2.43) can be written as

$$\frac{V_m}{u_\tau} = \frac{\left(\hat{R}_s+1\right)^2\left[\ln\left(\hat{R}_s+1\right)-\frac{3}{2}\right]+2\hat{R}_s+\frac{3}{2}}{\kappa\hat{R}_s^{\,2}} \tag{2.84}$$

Multiplying Eq. (2.71) through by this velocity ratio, the wall-scaled velocity profile can be written as

$$u^+\left(\hat{y},\kappa,\hat{R}_s\right) = \frac{\overline{V}_z}{V_m}\frac{V_m}{u_\tau} = \frac{\ln\left(\hat{R}_s\hat{y}+1\right)}{\kappa} + \frac{\left(\hat{R}_s+1\right)^2\left[\ln\left(\hat{R}_s+1\right)-\frac{3}{2}\right]+2\hat{R}_s+\frac{3}{2}}{\kappa\hat{R}_s^{\,2}}\delta\left(\hat{y},\hat{R}_s\right) \tag{2.85}$$

Using Eq. (2.84) in the definition of the Darcy friction factor given in Eq. (2.7), the corresponding fully rough relation for the friction factor is

$$4C_f = \frac{8\kappa^2 \hat{R}_s^{\,4}}{\left\{\left(\hat{R}_s+1\right)^2\left[\ln\left(\hat{R}_s+1\right)-\frac{3}{2}\right]+2\hat{R}_s+\frac{3}{2}\right\}^2} \tag{2.86}$$

Hence, we see that as a result of Eq. (2.75), the friction factor obtained from the velocity profile given by Eq. (2.71) is completely independent of the empirical function δ. The fully rough friction factor predicted from Eq. (2.86) depends only on the dimensionless inverse roughness ratio R/k_s and the two dimensionless closure constants κ and γ. Experimental data for the fully rough friction factor are known to correlate very well with the Nikuradse equation given in Eq. (2.20), which can also be written in terms of the dimensionless roughness ratio

$$4C_f = \frac{1}{\left[2.0\log_{10}\left(7.4\frac{R}{k_s}\right)\right]^2} \tag{2.87}$$

By minimizing the difference between Eqs. (2.86) and (2.87), the dimensionless closure constants κ and γ can be reevaluated independent of the empirical function δ. The resulting constants rounded to three significant digits are

$$\kappa = 0.403 \qquad \gamma = 0.0324 \tag{2.88}$$

Appendix CI presents the detailed optimization procedure used to evaluate the Von Kármán and Nikuradse constants based on the friction factor obtained from the well-accepted Nikuradse equation and the friction factor based on the fully rough limit of Prandtl's mixing length.

D. Corrective δ Function

A polynomial function could be used for the function δ to fit Nikuradse's velocity-profile data while satisfying Eqs. (2.72)–(2.75) and Eq. (2.83). Because there are five constraints on the velocity profile that must be satisfied independent of the experimental data, a seventh-order polynomial is necessary to provide three degrees of freedom for correlating the experimental data. For example, we could use a seventh-order polynomial in term of the relative coordinate $\hat{r} \equiv r/R$

$$\delta = C_0 + C_1\hat{r} + C_2\hat{r}^2 + C_3\hat{r}^3 + C_4\hat{r}^4 + C_5\hat{r}^5 + C_6\hat{r}^6 + C_7\hat{r}^7 \tag{2.89}$$

From the constraints imposed by Eqs. (2.72) and (2.83), the constants C_1 and C_3 can be evaluated independent of the experimental data,

$$C_1 = \frac{\hat{R}_s^{\,3}}{\left(\hat{R}_s+1\right)\left\{\left(\hat{R}_s+1\right)^2\left[\ln\left(\hat{R}_s+1\right)-\frac{3}{2}\right]+2\hat{R}_s+\frac{3}{2}\right\}} \tag{2.90}$$

and

$$C_3 = \frac{\hat{R}_s^{\,5}}{3\left(\hat{R}_s+1\right)^3\left\{\left(\hat{R}_s+1\right)^2\left[\ln\left(\hat{R}_s+1\right)-\frac{3}{2}\right]+2\hat{R}_s+\frac{3}{2}\right\}} \tag{2.91}$$

Because this correlation is being developed to estimate the eddy viscosity near the pipe centerline, it is useful to express the coefficient C_2 in terms of the eddy viscosity at the centerline. Although Eq. (2.76) is indeterminate at the centerline, applying l'Hospital's rule yields

$$\hat{\nu}_t\big|_{\hat{y}=1} = \frac{-\frac{u_\tau}{V_m}}{\left.\frac{d^2\hat{u}}{d\hat{y}^2}\right|_{\hat{y}=1}} \tag{2.92}$$

Using Eqs. (2.71), (2.84) and (2.89) in Eq. (2.92) and solving for the coefficient C_2 results in

$$C_2 = \frac{\hat{R}_s^2\left(\frac{\hat{R}_s^2}{\left(\hat{R}_s+1\right)^2}-\frac{\kappa}{\hat{\nu}_t\big|_{\hat{y}=1}}\right)}{2\left\{\left(\hat{R}_s+1\right)^2\left[\ln\left(\hat{R}_s+1\right)-\frac{3}{2}\right]+2\hat{R}_s+\frac{3}{2}\right\}} \tag{2.93}$$

Choosing the centerline eddy viscosity combined with C_0 and C_7 as the unknown correlation coefficients, the constraints imposed by Eqs. (2.72)–(2.75) can be used to evaluate C_4, C_5 and C_6

$$\begin{Bmatrix} D_1 \\ D_2 \\ D_3 \end{Bmatrix} = \begin{Bmatrix} -C_0 - C_1 - C_2 - C_3 - C_7 \\ -C_1 - 2C_2 - 3C_3 - 7C_7 \\ -C_0/2 - C_1/3 - C_2/4 - C_3/5 - C_7/9 \end{Bmatrix} \tag{2.94}$$

Using Gauss elimination, Eq. (2.94) can be rearranged easily to yield

$$\begin{Bmatrix} C_4 \\ C_5 \\ C_6 \end{Bmatrix} = \begin{bmatrix} -39 & 3 & 168 \\ 84 & -7 & -336 \\ -44 & 4 & 168 \end{bmatrix} \begin{Bmatrix} D_1 \\ D_2 \\ D_3 \end{Bmatrix} \tag{2.95}$$

The remaining unknown coefficients C_0 and C_7 and the centerline eddy viscosity could be obtained from an empirical correlation with Nikuradse's velocity profile data.

Notice that the dimensionless velocity profiles for fully rough flow, which are obtained from Eq. (2.71) combined with Eqs. (2.89)–(2.95), depend only on the inverse roughness ratio R/k_s. They are independent of the roughness Reynolds number k_s^+, as should be expected for fully rough limit. The maximum deviation from Eq. (2.70) for each of Nikuradse's velocity profiles is on the order of 1 to 6 percent of the mean velocity, which is likely the same order of magnitude as the experimental uncertainty. Nevertheless this small deviation from Eq. (2.70) is quite important because Eq. (2.70) does not satisfy the symmetry boundary conditions at the pipe centerline and the centerline symmetry has a significant effect on the eddy viscosity in the pipe core.

The coefficients C_0, C_7 and $\hat{\nu}_t\big|_{\hat{y}=1}$ are selected based on Nikuradse's velocity data . The coefficient C_0 describes the δ function at the centerline and is therefore a function of the roughness ratio. Although there are considerable scatter in Nikuradse's velocity data, an adequate correlation is provided by the relation

$$C_0 = 1.2\left(\frac{k_s}{R}\right)^{1.4} \tag{2.96}$$

Similarly, Eq. (2.89) provides good agreement with the mean of Nikuradse's velocity profile data in the region near the pipe centerline defined as $r/R \le 0.5$ if we use the empirical coefficients

$$\hat{\nu}_t\big|_{\hat{y}=1} = 0.056 \quad \text{and} \quad C_7 = -0.65 \tag{2.97}$$

Appendix CII gives more details on the δ function and shows the correlation between the experimental data and the near-wall fully rough limit. Notice that the dimensionless centerline eddy viscosity, $\hat{\nu}_t \equiv \nu_t/(u_\tau R)$, obtained from this empirical correlation is significantly less than the value 0.067 obtained from the empirical correlation proposed by Reichardt [79]

$$\frac{\nu_t}{u_\tau R} = \frac{\kappa}{6}\frac{y}{R}\left(2-\frac{y}{R}\right)\left[1+2\left(1-\frac{y}{R}\right)^2\right] \tag{2.98}$$

The turbulent eddy viscosity can be deduced from the dimensionless velocity given in Eq. (2.71). Using Eqs. (2.70) and (2.84) in Eq. (2.75), the dimensionless eddy-viscosity profile for fully rough pipe flow could be written as

$$\hat{\nu}_t = \kappa(1-\hat{y})\left(\frac{\hat{R}_s}{\hat{R}_s\hat{y}+1} + \frac{\left(\hat{R}_s+1\right)^2\left[\ln\left(\hat{R}_s+1\right)-\frac{3}{2}\right]+2\hat{R}_s+\frac{3}{2}}{\hat{R}_s{}^2}\frac{d\delta}{d\hat{y}}\right)^{-1} \tag{2.99}$$

Because the velocity profile is symmetric about the pipe centerline, Eqs. (2.75) and (2.99) are indeterminate at the centerline. Applying l'Hospital's rule to Eq. (2.99) yields

$$\hat{\nu}_t\big|_{\hat{y}=1} = \kappa\left(\frac{\hat{R}_s}{\left(\hat{R}_s+1\right)^2} - \frac{\left(\hat{R}_s+1\right)^2\left[\ln\left(\hat{R}_s+1\right)-\frac{3}{2}\right]+2\hat{R}_s+\frac{3}{2}}{\hat{R}_s{}^2}\left(\frac{d^2\delta}{d\hat{y}^2}\bigg|_{\hat{y}=1}\right)\right)^{-1} \tag{2.100}$$

and applying Eq. (2.88) to Eq. (2.100) results in

$$\hat{\nu}_t\big|_{\hat{y}=1} = \kappa\left(\frac{\hat{R}_s}{\left(\hat{R}_s+1\right)^2} - 2\frac{\left(\hat{R}_s+1\right)^2\left[\ln\left(\hat{R}_s+1\right)-\frac{3}{2}\right]+2\hat{R}_s+\frac{3}{2}}{\hat{R}_s{}^2}C_2\right)^{-1} \tag{2.101}$$

In summary, using the definitions $\hat{y} \equiv y/R$, $\hat{u} \equiv \overline{V}_z / V_m$, $u^+ \equiv \overline{V}_z / u_\tau$, $\hat{\nu}_t \equiv \nu_t / (u_\tau R)$ and $\hat{k}_s \equiv k_s / R$, the following algorithm provides empirical relations for the velocity and eddy viscosity profiles that satisfy Eqs. (2.71)–(2.74) and (2.82) while correlating Nikuradse's velocity profile data to within experimental uncertainty. The three parameters C_7, $\hat{\nu}_{tc}$ and C_0 that are used in the δ function were optimized to give the closest match to Nikuradse experimental data for fully rough flow. A detailed overview of the optimization process is given in Appendix CI.

The following constants are used

$$\kappa = 0.403, \quad \gamma = 0.0324, \quad \hat{\nu}_{tc} = 0.056, \quad \hat{\mathrm{r}} = 1 - \hat{y}, \quad \hat{R}_s = \frac{1}{\gamma \hat{k}_s}, \quad C_7 = -0.65 \tag{2.102}$$

The coefficients $D_0 - D_3$ and $C_0 - C_6$ are calculated from

$$\begin{aligned}
&D_0 = \left(\hat{R}_s + 1\right)^2 \left[\ln\left(\hat{R}_s + 1\right) - \frac{3}{2}\right] + 2\hat{R}_s + \frac{3}{2}, \quad C_0 = 1.2\left(\frac{k_s}{R}\right)^{1.4} \\
&C_1 = \frac{\hat{R}_s^3}{\left(\hat{R}_s + 1\right) D_0}, \quad C_2 = \frac{\hat{R}_s^2}{2D_0}\left[\frac{\hat{R}_s^2}{\left(\hat{R}_s + 1\right)^2} - \frac{\kappa}{\hat{\nu}_{tc}}\right], \quad C_3 = \frac{\hat{R}_s^5}{3\left(\hat{R}_s + 1\right)^3 D_0} \\
&D_1 = -C_0 - C_1 - C_2 - C_3 - C_7, \quad D_2 = -C_1 - 2C_2 - 3C_3 - 7C_7 \\
&D_3 = -C_0/2 - C_1/3 - C_2/4 - C_3/5 - C_7/9 \\
&C_4 = -39D_1 + 3D_2 + 168D_3, \quad C_5 = 84D_1 - 7D_2 - 336D_3 \\
&C_6 = -44D_1 + 4D_2 + 168D_3
\end{aligned} \tag{2.103}$$

The δ and its derivative with respect to $\hat{y}$ are obtained from

$$\begin{aligned}
&\delta = C_0 + C_1\hat{r} + C_2\hat{r}^2 + C_3\hat{r}^3 + C_4\hat{r}^4 + C_5\hat{r}^5 + C_6\hat{r}^6 + C_7\hat{r}^7 \\
&\delta' = \frac{d\delta}{d\hat{y}} = -C_1 - 2C_2\hat{r} - 3C_3\hat{r}^2 - 4C_4\hat{r}^3 - 5C_5\hat{r}^4 - 6C_6\hat{r}^5 - 7C_7\hat{r}^6
\end{aligned} \tag{2.104}$$

Finally the dimensionless velocity can be computed from

$$\hat{u} = \frac{\hat{R}_s^2 \ln\left(\hat{R}_s \hat{y} + 1\right)}{D_0} + \delta$$
$$u^+ = \frac{1}{\kappa} \ln\left(\hat{R}_s \hat{y} + 1\right) + \frac{D_0 \delta}{\kappa \hat{R}_s^2} \tag{2.105}$$

The dimensionless turbulent eddy viscosity is obtained from

$$\text{if } \left(\hat{y} = 1\right) \text{then}$$
$$\hat{\nu}_t = \frac{\kappa}{\dfrac{\hat{R}_s^2}{\left(\hat{R}_s + 1\right)^2} - \dfrac{2 D_0 C_2}{\hat{R}_s^2}}$$
$$\text{else} \tag{2.106}$$
$$\hat{\nu}_t = \frac{\kappa\left(1 - \hat{y}\right)}{\dfrac{\hat{R}_s}{\hat{R}_s \hat{y} + 1} + \dfrac{D_0 \delta'}{\hat{R}_s^2}}$$
$$\text{end if}$$

The velocity profiles obtained from this nondimensional turbulent eddy viscosity defined in Eqs. (2.105) and (2.106) are shown in Fig. 2.9 and are compared with the velocity profiles given by the log law in Eq. (2.40) from which the Colebrook equation and the Moody chart are based on using the constants given in Eq. (2.49). It can be seen in Fig. 2.9 that the velocity profiles obtained using the empirical function δ from Eq. (2.105) deviate slightly from the law of the wall but satisfy the symmetry boundary condition at the centerline.

The experimental dimensionless velocity profiles displayed in Fig. 2.9 are based on Nikuradse experimental data [47] carried out at six inverse roughness ratio $R/k_s = [15, 30.6, 60, 126, 252, 507]$ and at the different roughness Reynolds numbers shown in Table 2.1. The experiments carried out at an inverse roughness ratios of 252 and 507 do not qualify as fully rough flow because the roughness Reynolds number is less than 70, which is already a low limit for fully rough flow. Nevertheless, the empirical correlation given in Eq. (2.105) still gives satisfactory velocity profiles when compared to Nikuradse experimental data. There is a wide uncertainty in Nikuradse velocity data close to the wall. Therefore, only the points corresponding to $y/R \geq 0.5$ where taken into consideration when optimizing the coefficients of the empirical δ function to fit Nikuradse experimental velocity profiles.

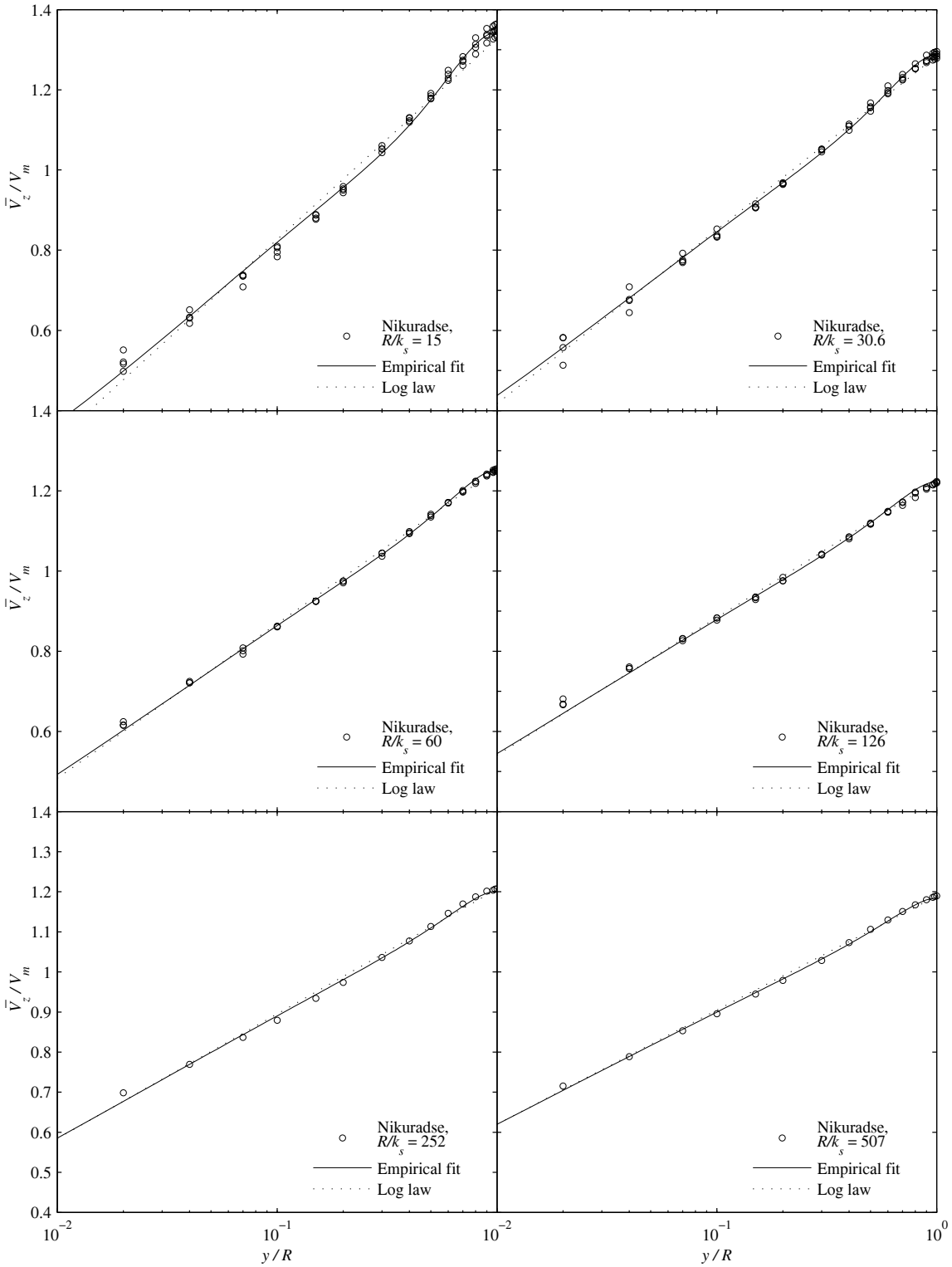

Fig. 2.9 Dimensionless velocity profiles as predicted by Eq. (2.105).

Table 2.1 Nikuradse's experiments roughness Reynolds numbers

R/k_s	15.0	30.6	60.0	126.0	252.0	507.0
	1248	790	378	277	67	49
k_s^+	584	456	244	152		
	320	242	151	99		
	125	129				

The nondimensional turbulent eddy viscosity obtained from the empirical δ function given in Eqs. (2.104)–(2.106) and corresponding to the velocity profiles shown in Fig. 2.9 is displayed in Fig. 2.10 and corresponds to the same range of roughness ratio. It is compared to hydraulically-smooth-wall pipe experimental data taken by Reichardt [79].

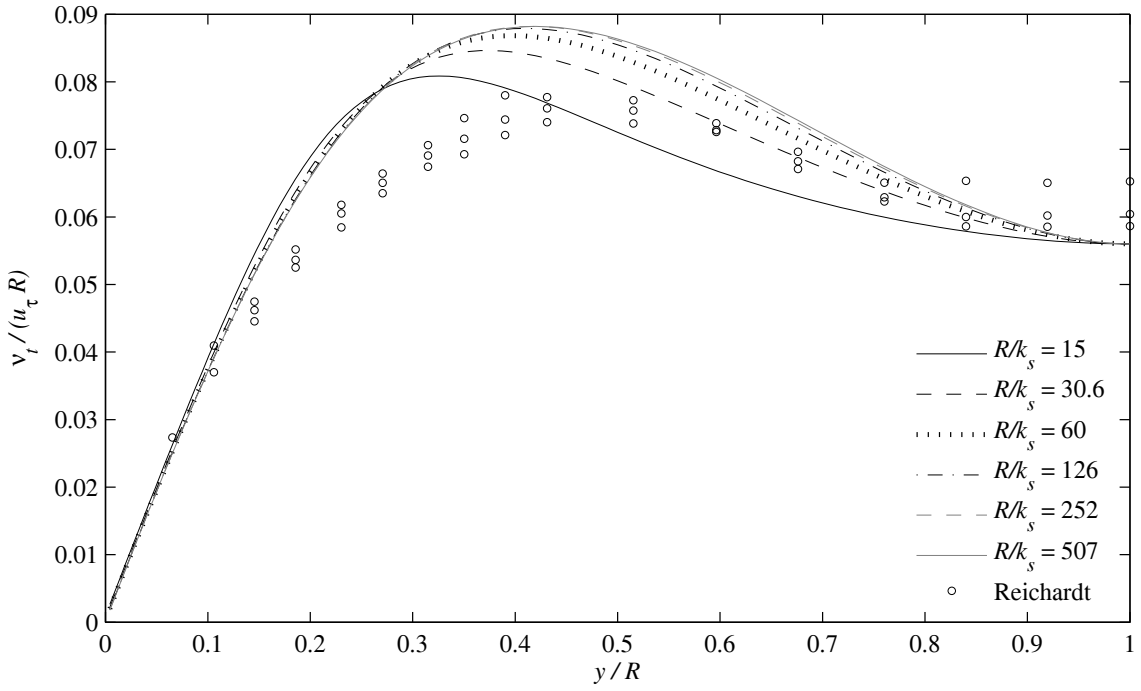

Fig. 2.10 Dimensionless Turbulent Eddy Viscosity Obtained From Eqs. (2.102)–(2.106).

From the empirical turbulent eddy viscosity function given in Eqs. (2.102)–(2.106), the turbulent-kinetic-energy transport equation may be solved by substituting the second turbulence variable with the

turbulent eddy viscosity equivalence. At this point, the turbulent-kinetic-energy wall boundary condition and the two closure coefficients σ_k and C_λ are still unknown for fully rough pipe.

IV. Fully Rough Flow Algebraic-Mean-Vortex-Wavelength Function

The turbulent wavelength profile may be written in terms of the turbulent kinetic energy and the turbulent eddy viscosity.

$$\lambda = \frac{\nu_t}{k^{1/2}} \tag{2.107}$$

Using the nondimensional parameters

$$\hat{\lambda} \equiv \frac{\lambda}{R}, \quad \hat{\nu}_t \equiv \frac{\nu_t}{u_\tau R}, \quad k^+ \equiv \frac{k}{u_\tau^2} \tag{2.108}$$

Equation (2.107) may be written in nondimensional form as

$$\hat{\lambda} = \frac{\hat{\nu}_t}{\sqrt{k^+}} \tag{2.109}$$

Reichardt [79] proposed the following empirical correlation for the turbulent eddy viscosity in hydraulically-smooth-wall pipes

$$\hat{\nu}_t = \frac{\kappa}{6}\left(1-\hat{r}^2\right)\left(1+2\hat{r}^2\right) \tag{2.110}$$

Or in terms of the wall coordinate $\hat{y}$

$$\hat{\nu}_t = \kappa\hat{y}\left(1-\frac{1}{2}\hat{y}\right)\left[\frac{1}{3}+\frac{2}{3}\left(1-\hat{y}\right)^2\right] \tag{2.111}$$

The turbulent eddy viscosity derivative is then

$$\frac{d\hat{\nu}_t}{d\hat{y}} = \kappa\left(1-\hat{y}\right)\left[\frac{1}{3}+\frac{2}{3}\left(1-\hat{y}\right)^2\right] - \kappa\,\hat{y}\left(1-\frac{1}{2}\hat{y}\right)\left[\frac{4}{3}\left(1-\hat{y}\right)\right] \tag{2.112}$$

In this form, it can easily be seen that as $\hat{y}$ approaches zero, $\hat{\nu}_t$ approaches $\kappa\hat{y}$. It can also be seen that as $\hat{y}$ approaches unity, $\hat{\nu}_t$ approaches $\kappa/6$ and the derivative of $\hat{\nu}_t$ with respect to $\hat{y}$ approaches zero.

Reichardt's empirical correlation for the turbulent eddy viscosity in circular pipes with hydraulically smooth walls is easily modified to agree closely with the well established near-wall behavior for the fully rough limit, which comes from Prandtl's mixing length empirical correlation $\ell = \kappa\left(y+\gamma\, k_s\right)$. Using the notation $\hat{k}_s = k_s / R$ and applying this near-wall correction to Eq. (2.111) yields

$$\hat{\nu}_t = \kappa\left(\gamma\,\hat{k}_s + \hat{y}\right)\left(1-\frac{1}{2}\hat{y}\right)\left[\frac{1}{3}+\frac{2}{3}\left(1-\hat{y}\right)^2\right] \tag{2.113}$$

Because $\gamma\, k_s << R$, for all practical purposes Eqs. (2.111) and (2.113) are identical, except for flow within a very thin layer near the pipe wall where y is on the order of k_s.

A similar but somewhat more flexible relation might be used for the mean vortex wavelength. For example, the limiting expressions for smooth-wall pipes might be

$$\hat{\lambda} \equiv \frac{\lambda}{R} = A_{\lambda 1}\hat{y}\left(1-\frac{1}{2}\hat{y}\right)\left[B_{\lambda 0} + B_{\lambda 1}\hat{r}^2 + B_{\lambda 2}\hat{r}^4 + \left(1-B_{\lambda 0}-B_{\lambda 1}-B_{\lambda 2}\right)\hat{r}^6\right] \tag{2.114}$$

with a derivative

$$\begin{aligned}\frac{d\hat{\lambda}}{d\hat{y}} = {} & A_{\lambda 1}\left(1-\hat{y}\right)\left[B_{\lambda 0} + B_{\lambda 1}\hat{r}^2 + B_{\lambda 2}\hat{r}^4 + \left(1-B_{\lambda 0}-B_{\lambda 1}-B_{\lambda 2}\right)\hat{r}^6\right] \\ & - A_{\lambda 1}\hat{y}\left(1-\frac{1}{2}\hat{y}\right)\left[2B_{\lambda 1}\hat{r} + 4B_{\lambda 2}\hat{r}^3 + 6\left(1-B_{\lambda 0}-B_{\lambda 1}-B_{\lambda 2}\right)\hat{r}^5\right]\end{aligned} \tag{2.115}$$

Note that at the pipe wall where $\hat{y} = 0$ and $\hat{r} = 1$, $\hat{\lambda}$ approaches $A_{\lambda 1}\hat{y}$. At the pipe centerline where $\hat{y} = 1$ and $\hat{r} = 0$, $\hat{\lambda} = A_{\lambda 1}\frac{B_{\lambda 0}}{2}$ and the derivative of $\hat{\lambda}$ with respect to $\hat{y}$ is zero. Following the development of Eq. (2.113), for fully rough walls, the following expression might be used for $\hat{\lambda}$

$$\hat{\lambda} \equiv \frac{\lambda}{R} = \left(A_{\lambda 0}\hat{k}_s + A_{\lambda 1}\hat{y}\right)\left(1 - \frac{1}{2}\hat{y}\right)\left[B_{\lambda 0} + B_{\lambda 1}\hat{r}^2 + B_{\lambda 2}\hat{r}^4 + \left(1 - B_{\lambda 0} - B_{\lambda 1} - B_{\lambda 2}\right)\hat{r}^6\right] \tag{2.116}$$

The five coefficients $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$, $B_{\lambda 2}$ are closure coefficients and need to be evaluated.

V. Closure Coefficients

The present turbulence model is composed of a transport equation for the turbulent kinetic energy and an algebraic equation for the second turbulence variable, the mean vortex wavelength. The *k*-**λ** model presents two closure coefficients associated with the turbulent-kinetic-energy transport equation and one algebraic relation for the mean vortex wavelength. The turbulent-kinetic-energy transport equation reduced to axisymmetric pipe flow is given by

$$\frac{d}{d\hat{r}}\left[-\left(\frac{\hat{\nu}}{3} + \frac{5\hat{\nu}_t}{3\sigma_k}\right)\hat{r}\frac{dk^+}{d\hat{r}}\right] = \frac{\hat{\nu}_t\hat{r}^3}{\left(\hat{\nu} + \hat{\nu}_t\right)^2} - C_\lambda\frac{\hat{\nu}\,k^+}{\hat{\lambda}^2}\hat{r} \tag{2.117}$$

There are two closure coefficients associated to this turbulent kinetic energy equation: C_λ and σ_k. The turbulent eddy viscosity is related to the turbulent kinetic energy and the mean vortex wavelength through an algebraic relation

$$\hat{\nu}_t = \hat{\lambda}\sqrt{k^+} \tag{2.118}$$

The empirical mean turbulent wavelength function for fully rough flow is

$$\hat{\lambda} \equiv \frac{\lambda}{R} = \left(A_{\lambda 0}\hat{k}_s + A_{\lambda 1}\hat{y}\right)\left(1 - \frac{1}{2}\hat{y}\right)\left[B_{\lambda 0} + B_{\lambda 1}\hat{r}^2 + B_{\lambda 2}\hat{r}^4 + \left(1 - B_{\lambda 0} - B_{\lambda 1} - B_{\lambda 2}\right)\hat{r}^6\right] \tag{2.119}$$

This empirical mean turbulent wavelength features five closure coefficients: $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$, $B_{\lambda 2}$. The symmetry boundary condition at the centerline is

$$\left.\frac{dk^+}{d\hat{r}}\right|_{\hat{r}=0} = 0 \tag{2.120}$$

One additional boundary condition is required. Experimental data show that the turbulent eddy viscosity remains finite at a fully rough surface. From Eq. (2.118), both the specific turbulent kinetic energy and the mean vortex wavelength remain finite at a fully rough surface. From Eq. (2.119), the mean vortex wavelength at the wall is proportional to the equivalent sand-grain roughness

$$\lambda\big|_{r=R} = A_{\lambda 0} k_s \tag{2.121}$$

From Eq. (2.65), the fully rough eddy viscosity at the pipe wall is

$$\nu_t\big|_{r=R} = \kappa \gamma u_\tau k_s \tag{2.122}$$

From Eqs. (2.118), (2.121) and (2.122), the turbulent kinetic energy at the wall is

$$k\big|_{r=R} = \left(\frac{\kappa \gamma}{A_{\lambda 0}}\right)^2 {u_\tau}^2 \tag{2.123}$$

A more general form is

$$k\big|_{r=R} = k^+_{\text{wall}} {u_\tau}^2 \tag{2.124}$$

where k^+_{wall} is a dimensionless proportionality coefficient. We find that the specific turbulent kinetic energy is proportional to the shear velocity squared.

VI. Summary and Conclusions

An algebraic relation for the mean-vortex-wavelength distribution in fully rough pipe flow was proposed and is given in Eq. (2.119). This equation has five unknown coefficients, $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$,

which can be tuned in order to match experimental data and empirical relations for the friction factor and mean velocity profile.

An algebraic relation for the mean velocity profile in fully rough pipe flow was obtained and is given in Eq. (2.105). This algebraic relation is based on a deviation from the log law, which is given in Eq. (2.40). The log law does not satisfy the centerline symmetry boundary condition or the no-slip boundary condition at the pipe wall. In order to better agree with the Nikuradse velocity profile data for fully rough flows and in order to satisfy physical constraints, a deviation from the log law was proposed as a seventh order polynomial function.

From this algebraic relation for the mean velocity distribution, the turbulent eddy viscosity profile can be deduced analytically for fully rough flow. The resulting turbulent eddy viscosity profile, which is given in Eq. (2.106), was compared to turbulent eddy viscosity profiles estimated by other authors directly from experimental data. The turbulent eddy viscosity cannot be measured directly. Instead, it has been traditionally estimated from numerical derivatives of experimental velocity data. The velocity gradients are extremely small near the center of the pipe. It was shown that, if central difference were used to extract the velocity gradients from the measured velocity profiles, and the experimental uncertainty in the velocity data was ± 0.2%, the uncertainty in the estimated velocity gradients would be on the order of ± 100% near the pipe centerline. From previously published results estimated from experimental data, about all that can be said concerning the turbulent eddy viscosity near the pipe centerline is that the pipe-scaled dimensionless kinematic eddy viscosity, $\hat{\nu}_t$, most likely falls somewhere between 0.0 and about 0.1.

Substituting the algebraic expressions developed in this chapter for the velocity distribution and the turbulent eddy viscosity profiles, the turbulent-kinetic-energy profile can be deduced from the turbulent kinetic-energy transport equation given in Eq. (2.117). Once the turbulent-kinetic-energy profile has been found, the second turbulent variable, the mean vortex wavelength, can be deduced from Eq. (2.109). This is the foundation of the proposed algebraic relation for the mean vortex wavelength.

Eventually, the Phillips k-λ model will be a two-equation turbulence model based on two transportable turbulence variables, the mean turbulent kinetic energy and the mean fluctuating vorticity. At the present time, the transport equation describing the mean fluctuating vorticity has not yet developed. Once it is developed, the resulting fluctuating vorticity distribution must be compared to a reference distribution for

closure coefficient evaluation and model assessment. The work presented in this dissertation is aimed at developing reference distributions for the mean fluctuating vorticity and mean turbulent wave length.

The proposed k-λ models feature several unknown coefficients. The turbulent-kinetic-energy transport equation given in Eq. (2.117) has two closure coefficients σ_k and C_λ. The rough wall boundary condition given in Eq. (2.124) contains the proportionality coefficient, k^+_{wall}, between the turbulent kinetic energy and at the wall and the shear velocity squared. The algebraic relation for the mean turbulent wavelength given in Eq. (2.119) has five unknown coefficients $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$. These eight coefficients need to be evaluated in order to find the distribution of the second turbulence variable. The closure coefficients might be evaluated by minimizing the deviations in the predicted friction factor from the Nikuradse equation given in Eq. (2.17), which is the the fully rough limit of the Colebrook equation, and the deviations in the predicted velocity profile from the well established law of the wall. There are no experimental data available for the velocity fluctuations at very high Reynolds number as it is not possible to obtain the instantaneous components of velocity accurately. Therefore, the closure coefficients will need to be evaluated by minimizing the differences in the friction factor and the velocity distribution when compared to the well established relation for fully rough pipe flow.

CHAPTER 3

PHILLIPS *K-Λ* TURBULENCE MODEL CLOSURE COEFFICIENTS EVALUATION

I. Introduction

The proposed *k-λ* model is based on a transport equation for the turbulent kinetic energy and an algebraic relation for the mean vortex wavelength. The turbulent-kinetic-energy transport equation for axisymmetric fully developed pipe flow is given by

$$\frac{1}{r}\frac{d}{dr}\left[-\left(\frac{\nu}{3}+\frac{5\nu_t}{3\sigma_k}\right)r\frac{dk}{d\hat{r}}\right]=\nu_t\left(\frac{u_\tau^2 r}{(\nu+\nu_t)R}\right)^2-C_\lambda\frac{\nu k}{\lambda^2} \tag{3.1}$$

where $\nu_t = \lambda k^{1/2}$. This equation contains two unknown closure coefficients σ_k and C_λ. These two closure coefficients should be dimensionless universal constants. Equation (3.1) requires a wall boundary condition for the turbulent kinetic energy. In Chapter 2, it was shown that, for fully rough flow, the specific turbulent kinetic energy at the wall is proportional to the shear velocity squared,

$$k\big|_{r=R}=k^+_{\text{wall}}u_\tau^2 \tag{3.2}$$

where k^+_{wall} is a dimensionless proportionality coefficient. The algebraic function for the mean vortex wavelength that was developed in the previous chapter is given by

$$\lambda=\left(A_{\lambda0}k_s+A_{\lambda1}\frac{y}{R}\right)\left(1-\frac{1}{2}\frac{y}{R}\right)\left[B_{\lambda0}+B_{\lambda1}\left(\frac{r}{R}\right)^2+B_{\lambda2}\left(\frac{r}{R}\right)^4+\left(1-B_{\lambda0}-B_{\lambda1}-B_{\lambda2}\right)\left(\frac{r}{R}\right)^6\right] \tag{3.3}$$

where $y = R - r$. This equation contains five unknown coefficients $A_{\lambda0}$, $A_{\lambda1}$, $B_{\lambda0}$, $B_{\lambda1}$ and $B_{\lambda2}$. At the wall, the mean-vortex-wavelength equation reduces to

$$\lambda\big|_{r=R}=A_{\lambda0}k_s \tag{3.4}$$

For fully rough pipe flow, the value of the mean vortex wavelength at the wall should depend only on the surface roughness k_s. Therefore, the coefficient $A_{\lambda0}$ should be a constant. The remaining four coefficients

of the mean vortex wavelength $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ need not be constants, but could be functions of the flow parameters such as the Reynolds number and roughness ratio.

This chapter is aimed at evaluating eight coefficients: two closure coefficients σ_k and C_λ from the turbulent-kinetic-energy transport equation, the proportionality coefficient k^+_{wall} from the wall boundary condition and five coefficients $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$, $B_{\lambda 2}$ from the algebraic relation for the mean-vortex-wavelength empirical relation. The values for these eight coefficients have a significant impact on the accuracy of the model and therefore need to be evaluated carefully.

II. Optimization Process

The eight coefficients σ_k, C_λ, k^+_{wall}, $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$, $B_{\lambda 2}$ were evaluated based on a computer optimization program. The model is first evaluated for a given set of closure coefficients. By solving the differential equations and comparing the resulting solution with a desired solution, the computer program returns a fitness parameter that quantifies how well the solution matches the desired solution. This fitness parameter is a single real number and is used in the value to be minimized in the optimization code. The optimization program calculates a gradient vector for each of the variables to be optimized. This gradient vector represents the rate of change of the fitness value with respect to each variable being optimized. In order to minimize the difference to the desired solution, the optimization program uses the negative of the gradient vector as the search direction at the each iteration.

A. Fitness Parameter

The first step in any optimization program is to define a fitness parameter, which, if minimized, results in the optimal solution. A fitness parameter is returned based on a comparison of velocity distributions obtained for fully rough pipe flows and their corresponding friction factor. The friction factor can be obtained experimentally by measuring the pressure drop. Instantaneous velocity distribution may be acquired via hot wire anemometry. However, no experimental data can be obtained at very high Reynolds numbers because the frequency of the fluctuating components of the velocity is of the same order of magnitude or greater than the internal frequency of the hot wire anemometers used to capture the velocity profile during the experiments. Therefore, the fully rough flow model can only be compared with an

empirical correlation rather than experimental data. A well established empirical correlation for the friction factor is the Colebrook equation, from which the Moody diagram is based. The Colebrook equation can be used at high Reynolds numbers and covers the fully rough flow region. Some of Nikuradse experiments [47] were carried out at roughness Reynolds numbers k_s^+ greater than 100 and could be used to compare the velocity distribution against.

The friction factor for fully rough flow is compared to the limit of the Colebrook equation at high Reynolds numbers which is given by the Nikuradse equation. The Nikuradse equation can be expressed as

$$4C_f = \frac{1}{\left[1.74 + 2\log_{10}\left(R/k_s\right)\right]^2} \tag{3.5}$$

For fully rough flows, the friction factor is independent of the Reynolds number and depends only on the roughness ratio. Equation (2.17) is used as a reference for the friction factor for fully rough flow.

The velocity distribution is compared with an empirical relation based on Nikuradse's [47] rough flow experiments. A corrective function is superimposed on the log law in order to provide better agreement to Nikuradse's experimental data. The corrective function is selected such that the velocity satisfies the symmetry boundary condition at the centerline and the no-slip boundary condition at the wall. The reference for the velocity profile for fully rough flow is given by

$$\begin{aligned} \hat{u} &= \frac{\hat{R}_s^2 \ln\left(\hat{R}_s \hat{y} + 1\right)}{D_0} + \delta \\ u^+ &= \frac{1}{\kappa}\ln\left(\hat{R}_s \hat{y} + 1\right) + \frac{D_0 \delta}{\kappa \hat{R}_s^2} \end{aligned} \tag{3.6}$$

This function was derived in detail in the previous chapter. The δ function is obtained from

$$\delta = C_0 + C_1\hat{r} + C_2\hat{r}^2 + C_3\hat{r}^3 + C_4\hat{r}^4 + C_5\hat{r}^5 + C_6\hat{r}^6 + C_7\hat{r}^7 \tag{3.7}$$

The eight coefficients C_0, C_1, C_2, C_3, C_4, C_5, C_6 and C_7 are given by the following system of equation

$$
\begin{aligned}
&D_0 = \left(\hat{R}_s + 1\right)^2 \left[\ln\left(\hat{R}_s + 1\right) - \frac{3}{2}\right] + 2\hat{R}_s + \frac{3}{2}, \quad C_0 = 1.2\left(\frac{k_s}{R}\right)^{1.4} \\
&C_1 = \frac{\hat{R}_s^3}{\left(\hat{R}_s + 1\right)D_0}, \quad C_2 = \frac{\hat{R}_s^2}{2D_0}\left[\frac{\hat{R}_s^2}{\left(\hat{R}_s + 1\right)^2} - \frac{\kappa}{\hat{\nu}_{tc}}\right], \quad C_3 = \frac{\hat{R}_s^5}{3\left(\hat{R}_s + 1\right)^3 D_0} \\
&D_1 = -C_0 - C_1 - C_2 - C_3 - C_7\,, \quad D_2 = -C_1 - 2C_2 - 3C_3 - 7C_7 \\
&D_3 = -C_0/2 - C_1/3 - C_2/4 - C_3/5 - C_7/9 \\
&C_4 = -39D_1 + 3D_2 + 168D_3\,, \quad C_5 = 84D_1 - 7D_2 - 336D_3 \\
&C_6 = -44D_1 + 4D_2 + 168D_3 \\
&C_7 = -0.65
\end{aligned}
\tag{3.8}
$$

The velocity profile obtained from the *k*-λ turbulence model is discretized into twelve points and compared at a roughness Reynolds number of 80,000 to the reference distribution given by Eqs. (2.105)–(3.8). The choice of the roughness Reynolds number of 80,000 arises from the fact that at this high of roughness Reynolds number, the ratio of turbulent eddy viscosity over the molecular viscosity is close to one thousand. The flow is said to be fully rough when the molecular viscosity is insignificant compared to the turbulent eddy viscosity throughout the flow field, which is the case at a roughness Reynolds number of 80,000. The velocity profile is compared at 12 sample points ranging from the centerline to the wall

$$
\hat{r} = [0.0\,,\ 0.05\,,\ 0.1\,,\ 0.2\,,\ 0.3\,,\ \ 0.4\,,\ 0.5\,,\ \ 0.6\,,\ 0.7\,,\ 0.8\,,\ \ 0.9\ \ \ 0.95] \tag{3.9}
$$

It is not possible to accurately measure the velocity really close to the wall, so the first sample point is only located at 5% of the pipe diameter from the wall. At the wall, the no-slip boundary condition imposes a zero velocity. The difference between the velocities obtained from the *k*-λ turbulence model and the reference velocities given by Eqs. (3.5)–(3.8) is calculated at each sample points. The sum of the absolute value of the difference divided by the reference velocities describes the fitness to the velocity distribution for fully rough flow.

The friction factor from the current *k*-λ turbulence model is compared to that obtained from the Nikuradse equation while varying the roughness Reynolds number from 100 to a roughness Reynolds number that gives a Reynolds number of 10^8. The moody diagram is usually represented for Reynolds numbers up to 10^8. For this reason, a limit of 10^8 for the Reynolds number was chosen to be the high limit when varying the roughness Reynolds number k_s^+. The flow is fully rough when the molecular viscosity

can be neglected when compared to the turbulent eddy viscosity. At a roughness Reynolds number of 1000, the ratio of the turbulent eddy viscosity to the molecular viscosity is 12. At a roughness Reynolds number of 100, this ratio is 1.2, which is obviously not large compared to unity. The model was first evaluated by comparing the friction factor for roughness Reynolds numbers higher than 1000. It was found that the model could estimate the friction factor and velocity distribution really accurately at given σ_k and C_λ values.

The fully rough flow solution is independent of C_λ. In the *k*-λ formulation given in Eq. (3.1), the coefficient C_λ only occurs as the product of C_λ with the molecular viscosity. By definition, fully rough flow is attained when the Reynolds number is independent of the kinematic molecular viscosity. From Eq. (3.1), when the solution becomes independent of the molecular viscosity it will also become independent of C_λ. Therefore, an estimate for the coefficient C_λ can only be obtained for flow that are not fully rough, that is at a lower roughness Reynolds number. In order to estimate the closure coefficients σ_k and C_λ the friction factor was compared to that obtained from the Nikuradse equation at a roughness Reynolds number as low as 100. The logarithm of the roughness Reynolds number is then increased by 0.25 until the Reynolds number becomes large enough (10^8).

The fitness parameter is defined as a weighted value between the fit to the Nikuradse equation for fully rough flow given in Eq. (2.17) and the fit to the velocity distribution profile given in Eqs. (2.105)–(3.8). The weighting factor is 50%.

The turbulent kinetic energy, mean turbulent wavelength and turbulent eddy viscosity were not weighted when estimating the coefficients. At high Reynolds numbers, the sampling rate of a hot wire anemometry has a frequency smaller or of the same order of magnitude as the fluctuating components of the velocity. Because the current technology does not allow sampling the fluid velocity at high Reynolds numbers, it is not possible to obtain accurate estimates for the turbulent kinetic energy, mean vortex turbulent wavelength or turbulent eddy viscosity.

B. BFGS Algorithm

A gradient-based optimization method is used in order to minimize the difference between the desired solution and the current model. A gradient based method was chosen for this optimization routine because

the fitness function is expected to be continuous throughout the design space. The quasi-Newton's method is based on Newton's method to find a stationary point of a function where the gradient is 0. Newton's method assumes that the function can locally be approximated as a quadratic region around the optimum and use the first and second derivatives to find the stationary point. Quasi-Newton's methods are a generalization of the secant method to find the root of the first derivative for multidimensional problems. One of the most common quasi-Newton methods is the BFGS method, suggested independently by Broyden [81], Fletcher [82], Goldfarb [83] and Shanno [84]. For details on the implementation of the BFGS method, see the work of Hunsaker [85] and Appendix E.

The BGFS method does not search for the global minimum. An initial value is provided for the variables to be optimized. Because of the complexity of the model, the optimization technique can easily be entrapped in a local minimum. To ensure that the solution found is a global minimum, the problem may be started from different initial values. If all converge to the same solution, then the optimized variables represent a global minimum. In some instances, one of the optimized variables returned from the optimization solver present very little variations from its initial estimate, even when started from different values. A possible reason is that the solution is independent from this variable. Another possibility is that the step gradient is too small and the solution found represents a local minimum. In which case, the step gradient was increased. To find the minimum of a function using a step gradient method like the BFGS algorithm, one selects a step proportional to the negative of the gradient of the function at the current point. This step is called the step gradient. The optimization code is given in Appendix D.

III. Closure Coefficient Evaluation

The eight turbulence model coefficients σ_k, C_λ, $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$, $B_{\lambda 2}$ and k^+_{wall} were evaluated over the discrete range of the inverse roughness ratios 15.0, 30.6, 60.0, 126.0, 252.0, 507.0, 0.5/0.000058. The first six inverse roughness ratios correspond to those used by Nikuradse during his rough flow experiments [47]. The last inverse roughness Reynolds number (0.5/0.000058 ≈ 8621) was used by Shockling et al. [72] during their fully rough flow measurements. In order to study the dependancy of the closure coefficients on the roughness ratios, those eight closure coefficients were first optimized as a set for each of the 7 roughness ratios.

A. Variations of the Model Coefficients with the Roughness Ratio

The two closure coefficients σ_k and C_λ from the turbulent-kinetic-energy transport equation should be universal constants. The optimization code requires a set of initial values for all of the variables being optimized. When the variable σ_k was being optimized over a range of roughness ratios, the optimized returned value was always really close to the initial value. Therefore, this variable was initially not optimized and was kept constant. The same phenomenon happened for the value of the proportionality coefficient in the wall boundary condition. Therefore those two variables were at first held constants. The proportionality coefficient in the wall boundary condition, k^+_{wall}, was at first expected to be in the range 0.05 to 2.0. The closure coefficient σ_k was at first expected to be in the range 0.5 to 8.0. The coefficient C_λ and the other five coefficients $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$, $B_{\lambda 2}$ were all optimized simultaneously for each of the 7 roughness ratios at a constant σ_k and constant k^+_{wall}. This gave an estimate for the closure coefficient C_λ (which should be a universal constant).

This process of running the optimization code at a given σ_k and k^+_{wall} to optimize the remaining six coefficients ($A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$, $B_{\lambda 2}$ and C_λ) was repeated several times over a range of σ_k and k^+_{wall}. The following section gives an example of the optimization results for $\sigma_k = 6.0$ and $k^+_{\text{wall}} = 0.1$.

An example of the optimized C_λ coefficient is given in Fig. 3.1. The coefficients σ_k and k^+_{wall} were held constant. The optimization code returned the optimal values for C_λ, $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ for each of the seven roughness ratios. The coefficient C_λ should be a universal constant. For the case $\sigma_k = 6.0$ and $k^+_{\text{wall}} = 0.1$, the coefficient C_λ was selected to be 0.00036. This value is only a rough estimate. It will be reoptimized at a later stage.

Once an estimate for C_λ is obtained at each σ_k and k^+_{wall} values, the remaining five coefficients $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$, $B_{\lambda 2}$ can be reevaluated. The optimization code was repeatedly used for different values of σ_k, in the range 1.0 to 7.0 and different values of k^+_{wall} in the range 0.05 to 1.0 holding C_λ constant. An example is shown in Fig. 3.2 for $\sigma_k = 6.0$, $C_\lambda = 0.00036$ and $k^+_{\text{wall}} = 0.1$. For other values of k^+_{wall} and σ_k, the five coefficients $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$, $B_{\lambda 2}$ showed a similar trend.

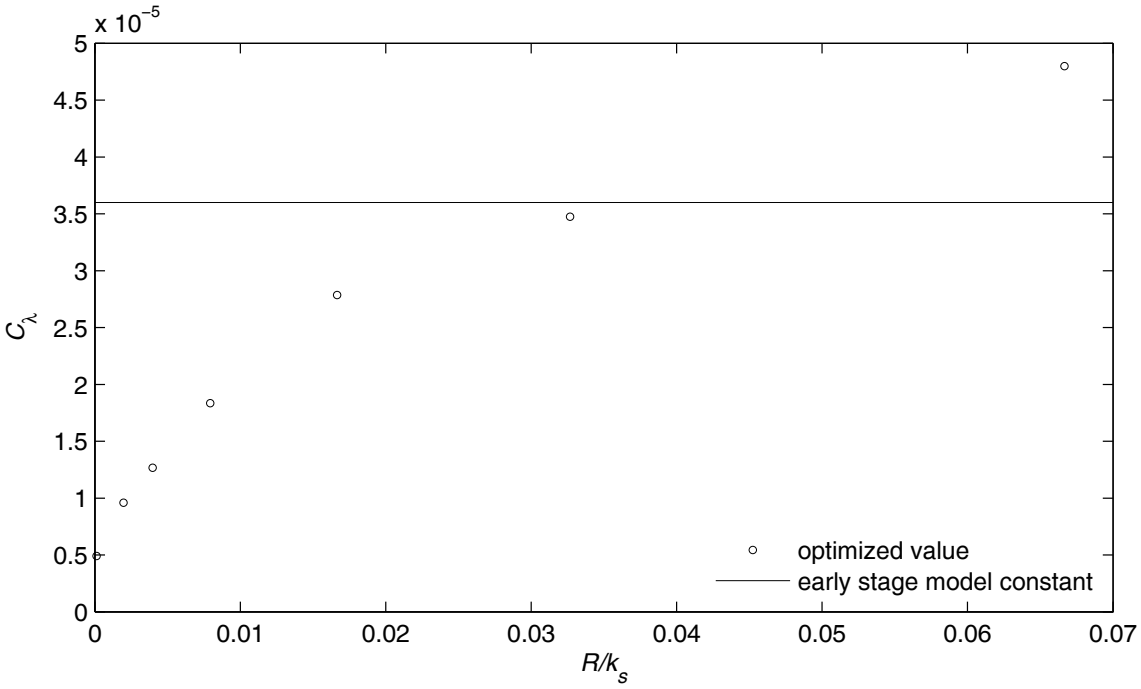

Fig. 3.1 Optimized coefficient C_λ at selected roughness ratios using σ_k = 6.0 and k^+_{wall} = 0.1.

The coefficient $A_{\lambda 0}$ representing the value of the mean turbulent wavelength at the wall and should be a constant for fully rough flow. The mean vortex wavelength is given by

$$\lambda\big|_{r=R} = A_{\lambda 0} k_s \tag{3.10}$$

The mean vortex wavelength should only depend on the roughness ratio at the wall. Therefore $A_{\lambda 0}$ should be a constant. Figure 3.2 shows that the optimized value is not a constant but varies slightly with the roughness Reynolds number. Just like it was the case for the C_λ coefficient, a constant is at first selected, to be used as a rough estimate. The remaining four coefficients are re-optimized, as shown in Fig. 3.3. Comparing Fig. 3.3 to Fig. 3.2, a few differences can be seen in the actual values, but the variations of the coefficients with respect to the roughness Reynolds number remain the same. From the dependency of the four remaining coefficients $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ with the roughness ratio that is shown in Fig. 3.3, functions were proposed for these four coefficients. A linear function for the coefficients $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ is originally proposed.

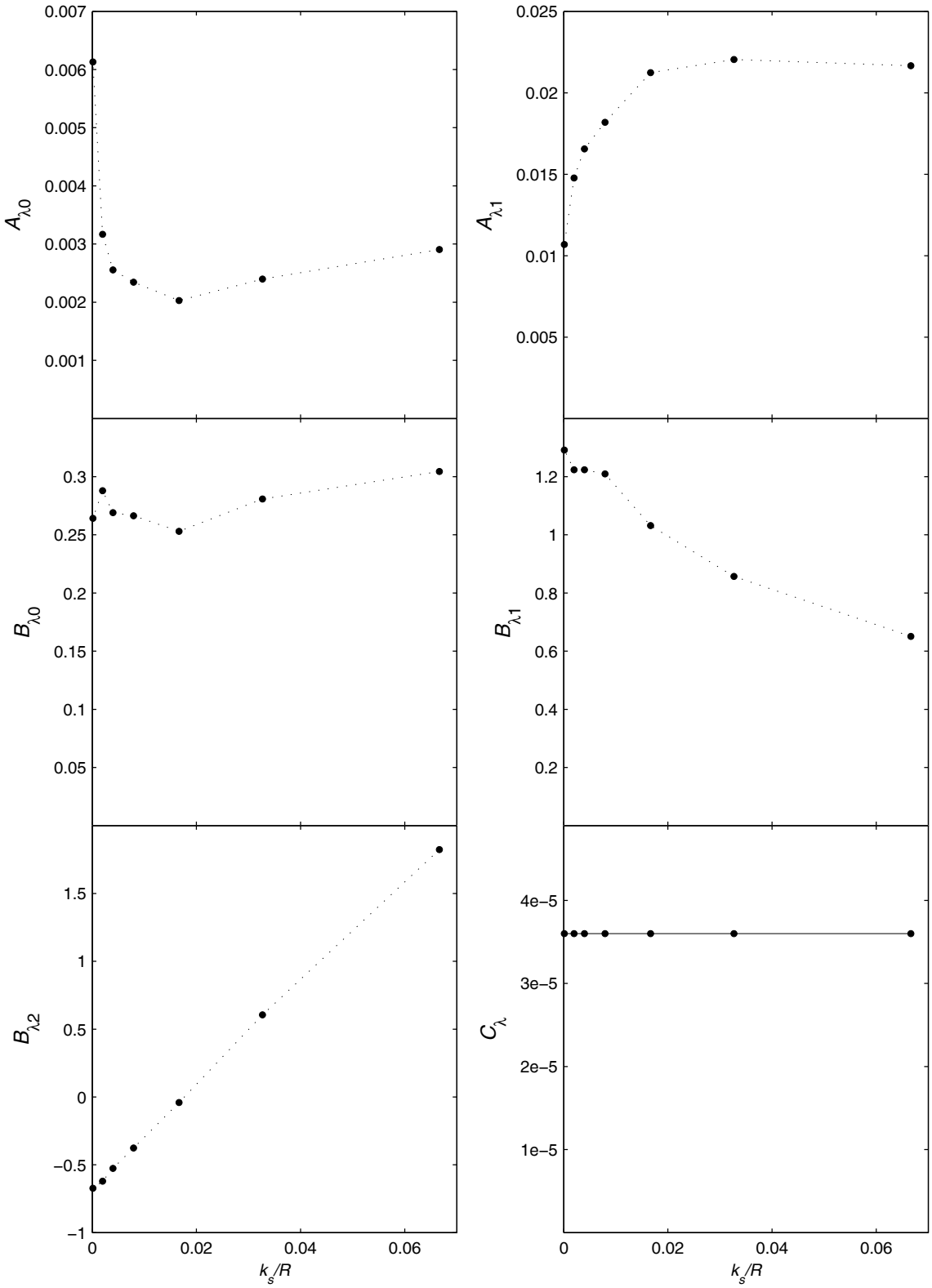

Fig. 3.2 Optimized $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$, $B_{\lambda 2}$ coefficients using $\sigma_k = 6$, $k^+_{\text{wall}} = 0.1$ and with a non-optimized C_λ held constant.

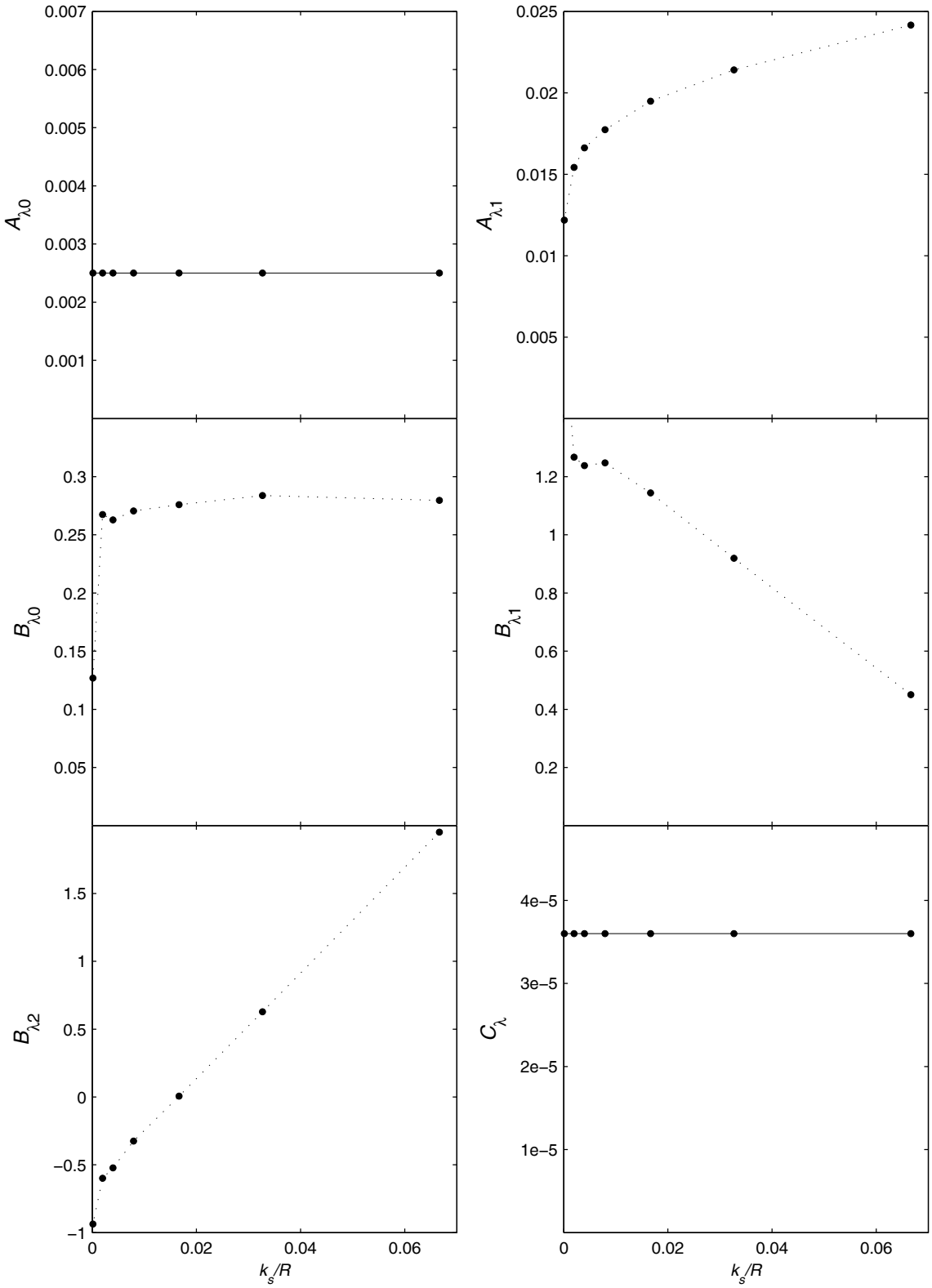

Fig. 3.3 Optimized $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$, $B_{\lambda 2}$ coefficients using $\sigma_k = 6$, $k^+_{\text{wall}} = 0.1$ and with a non-optimized C_λ and $A_{\lambda 0}$ held constant.

The coefficient $A_{\lambda 1}$ is fitted with a more complex function that asymptotically approaches a linear function for high roughness ratios and approaches an exponential function for low roughness ratios. The following functions are initially suggested

$$A_{\lambda 1} = A_{\lambda 11} + A_{\lambda 12}\hat{k}_s\left(A_{\lambda 10} - A_{\lambda 11} - A_{\lambda 12}\hat{k}_s\right)\exp\left(-\hat{k}_s^{\,A_{\lambda 13}}\right) \tag{3.11}$$

$$B_{\lambda 0} = B_{\lambda 00} + B_{\lambda 01}\hat{k}_s \tag{3.12}$$

$$B_{\lambda 1} = B_{\lambda 10} + B_{\lambda 11}\hat{k}_s \tag{3.13}$$

$$B_{\lambda 2} = B_{\lambda 20} + B_{\lambda 21}\hat{k}_s \tag{3.14}$$

The BFGS optimization method does not search for the global minimum. Therefore the initial values given to the optimization solver are of extreme importance. Initial estimates for the ten coefficients $A_{\lambda 10}$, $A_{\lambda 11}$, $A_{\lambda 12}$, $A_{\lambda 13}$, $B_{\lambda 00}$, $B_{\lambda 01}$, $B_{\lambda 10}$, $B_{\lambda 11}$, $B_{\lambda 20}$ and $B_{\lambda 21}$ are obtained using the build in optimization solver of Excel. At $\sigma_k = 6.0$, $C_\lambda = 0.000036$ and $k^+_{\text{wall}} = 0.1$, the following initial estimates are used

$$\begin{aligned} &A_{\lambda 0} = 0.002500, \quad A_{\lambda 10} = 0.01197, \quad A_{\lambda 11} = 0.05278, \quad A_{\lambda 12} = 0.009954 \\ &A_{\lambda 13} = 0.3900, \quad B_{\lambda 00} = 0.2694, \quad B_{\lambda 01} = 0.1572, \quad B_{\lambda 10} = 1.353 \\ &B_{\lambda 11} = -13.27, \quad B_{\lambda 20} = -0.6674, \quad B_{\lambda 21} = 39.30 \end{aligned} \tag{3.15}$$

Figure 3.4 shows the proposed functions given in Eqs. (3.11)–(3.14) for the four coefficients of the mean turbulent wavelength, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$. The coefficients $A_{\lambda 0}$ and C_λ are kept constant. The ten coefficients $A_{\lambda 10}$, $A_{\lambda 11}$, $A_{\lambda 12}$, $A_{\lambda 13}$, $B_{\lambda 00}$, $B_{\lambda 01}$, $B_{\lambda 10}$, $B_{\lambda 11}$, $B_{\lambda 20}$ and $B_{\lambda 21}$ used in Eqs. (3.11)–(3.14) are given in Eq. (3.15). These ten coefficients were obtained by minimizing the fitness parameter at each roughness ratio using the BFGS optimization algorithm.

The value for the closure coefficient C_λ and the constant $A_{\lambda 0}$ used in Figs. 3.2–3.4 were only rough estimates. The ten coefficients $A_{\lambda 10}$, $A_{\lambda 11}$, $A_{\lambda 12}$, $A_{\lambda 13}$, $B_{\lambda 00}$, $B_{\lambda 01}$, $B_{\lambda 10}$, $B_{\lambda 11}$, $B_{\lambda 20}$ and $B_{\lambda 21}$ given in Eq. (3.15) along with the closure coefficient C_λ and the constant $A_{\lambda 0}$ need to be reevaluated when the seven roughness ratios are optimized simultaneously.

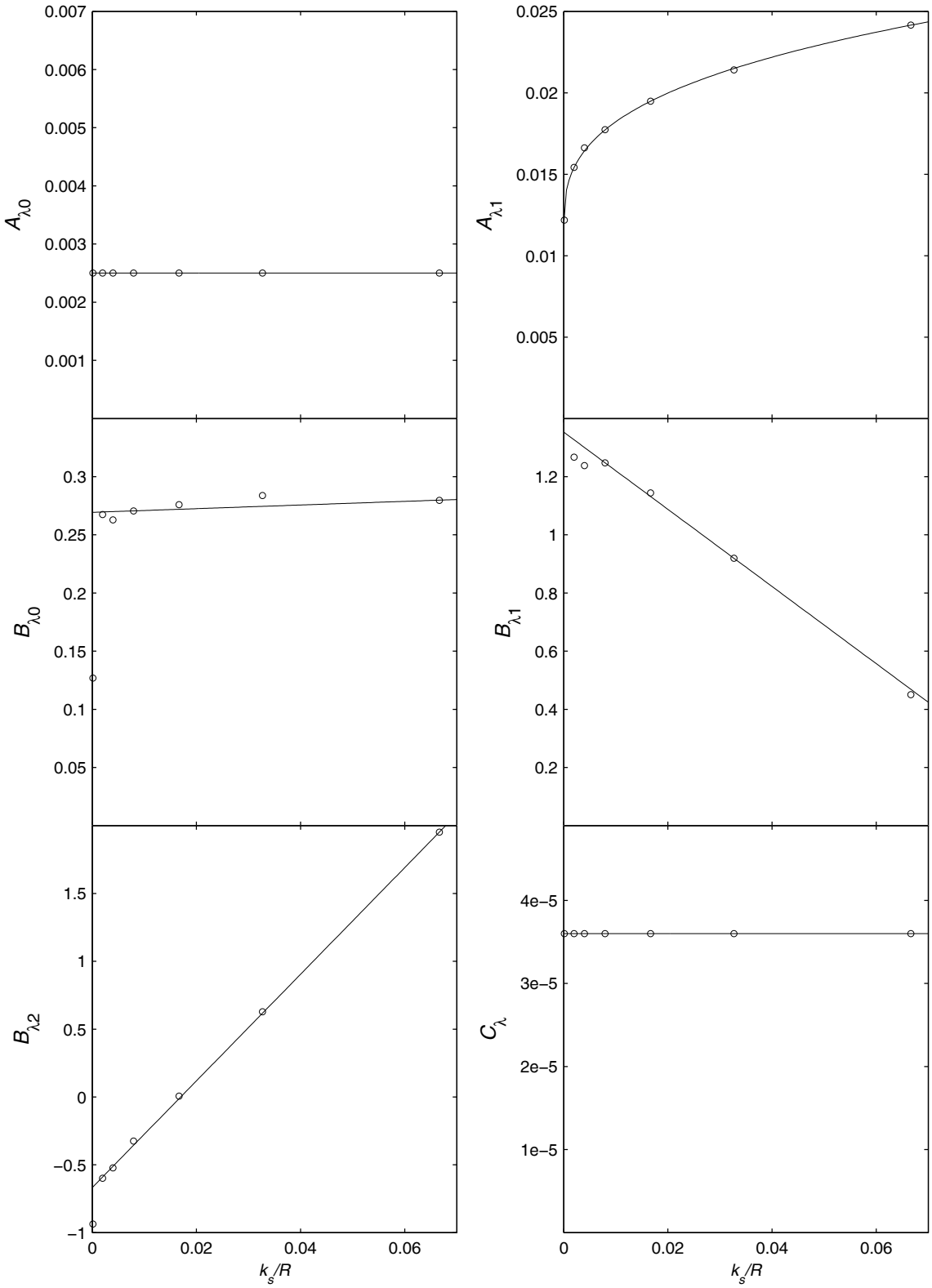

Fig. 3.4 Proposed functions for $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ using σ_k = 6 and k^+_{wall} = 0.1 and with a non-optimized C_λ and $A_{\lambda 0}$ held constant.

After iterating several times on different σ_k and k^+_{wall}, the range of σ_k and k^+_{wall} was slightly decreased. Figures 3.5 and 3.6 show the fitness values and the coefficient C_λ for the range of σ_k 0.5 to 8.0 and based on the optimization of the coefficients C_λ, $A_{\lambda 0}$, $A_{\lambda 10}$, $A_{\lambda 11}$, $A_{\lambda 12}$, $A_{\lambda 13}$, $B_{\lambda 00}$, $B_{\lambda 01}$, $B_{\lambda 10}$, $B_{\lambda 11}$, $B_{\lambda 20}$ and $B_{\lambda 21}$ simultaneously. It can be seen in Fig. 3.6 that the coefficient C_λ clearly converges to 0 for values of σ_k greater than 6. From Fig. 3.5, the fitness decreases as σ_k increases. Therefore the closure coefficient σ_k is expected to be in the range 2.0 to 6.0. Figures 3.5 and 3.6 were generated for k^+_{wall} in the range 0.1 to 2.0. For values of σ_k greater than 2.0, the value of the proportionality coefficient in the wall boundary condition does not affect the fitness parameter much.

In order to validate the equations suggested in Eqs. (3.11)–(3.14), the four coefficients $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ are reevaluated for each individual roughness ratios. If after the optimization, the four coefficients do not change from their initial values, then the proposed equations given in Eqs. (3.11)–(3.14) can be validated. The coefficients $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ are computed from the optimized $A_{\lambda 10}$, $A_{\lambda 11}$, $A_{\lambda 12}$, $A_{\lambda 13}$, $B_{\lambda 00}$, $B_{\lambda 01}$, $B_{\lambda 10}$, $B_{\lambda 11}$, $B_{\lambda 20}$ and $B_{\lambda 21}$ coefficients. This serves as an initial estimate in the optimization code. In Fig. 3.7, the circles represent the initial estimates and the solid dots the final optimized values. There are very little changes in the $A_{\lambda 1}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ coefficients. However, the optimized $B_{\lambda 0}$ coefficient shows a nonlinear trend.

A similar function as for the $A_{\lambda 1}$ coefficient is fitted to $B_{\lambda 0}$. The coefficient $B_{\lambda 0}$ is now written as a function that asymptotically approaches a linear function for low roughness ratios and an exponential function for high roughness ratios.

$$B_{\lambda 0} = B_{\lambda 01} + B_{\lambda 02}\hat{k}_s\left(B_{\lambda 00} - B_{\lambda 01} - B_{\lambda 02}\hat{k}_s\right)\exp\left(-\hat{k}_s^{\,B_{\lambda 03}}\right) \tag{3.16}$$

With this new function for $B_{\lambda 0}$, the fourteen coefficients $A_{\lambda 0}$, $A_{\lambda 10}$, $A_{\lambda 11}$, $A_{\lambda 12}$, $A_{\lambda 13}$, $B_{\lambda 00}$, $B_{\lambda 01}$, $B_{\lambda 02}$, $B_{\lambda 03}$, $B_{\lambda 10}$, $B_{\lambda 1}$, $B_{\lambda 20}$, $B_{\lambda 21}$ and C_λ are reoptimizeded all together for all the seven roughness ratios. From the optimized coefficients, the $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ are evaluated for each roughness ratios using Eqs. (3.11), (3.13), (3.14) and (3.16). Those four coefficients are then optimized at each roughness ratios.

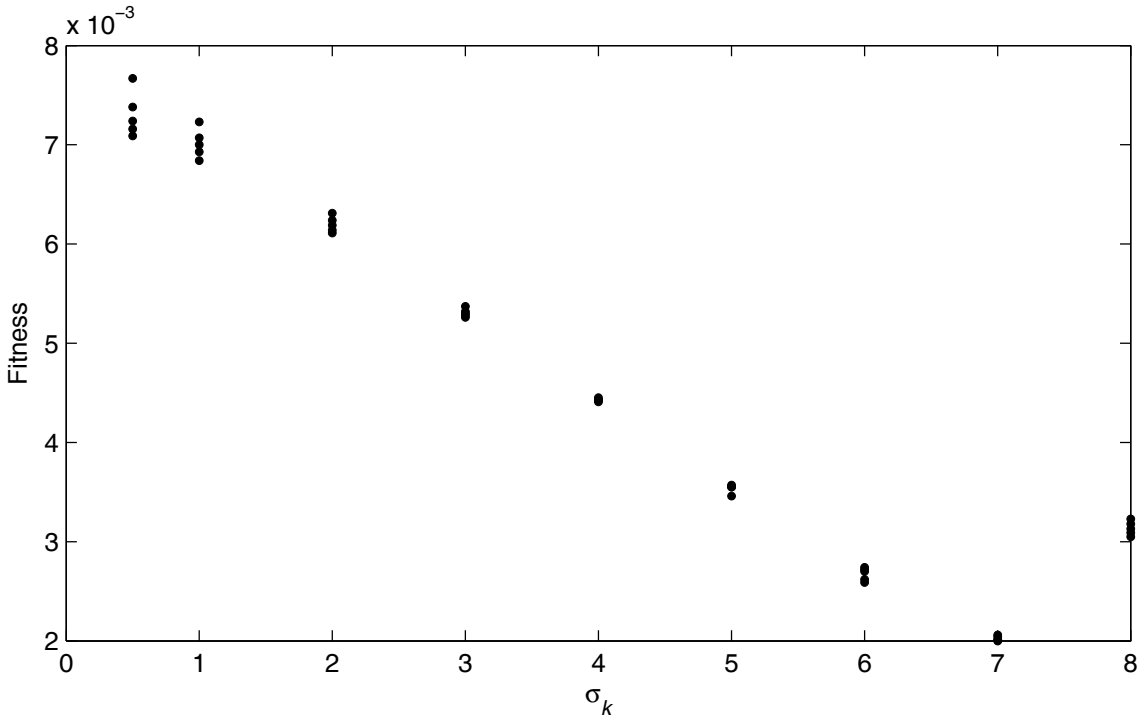

Fig. 3.5 Fitness Values for different k^+_{wall} in the range [0.1 – 2.0] resulting from the optimization of C_λ, $A_{\lambda 0}$, $A_{\lambda 10}$, $A_{\lambda 11}$, $A_{\lambda 12}$, $A_{\lambda 13}$, $B_{\lambda 00}$, $B_{\lambda 01}$, $B_{\lambda 10}$, $B_{\lambda 11}$, $B_{\lambda 20}$ and $B_{\lambda 21}$ simultaneously.

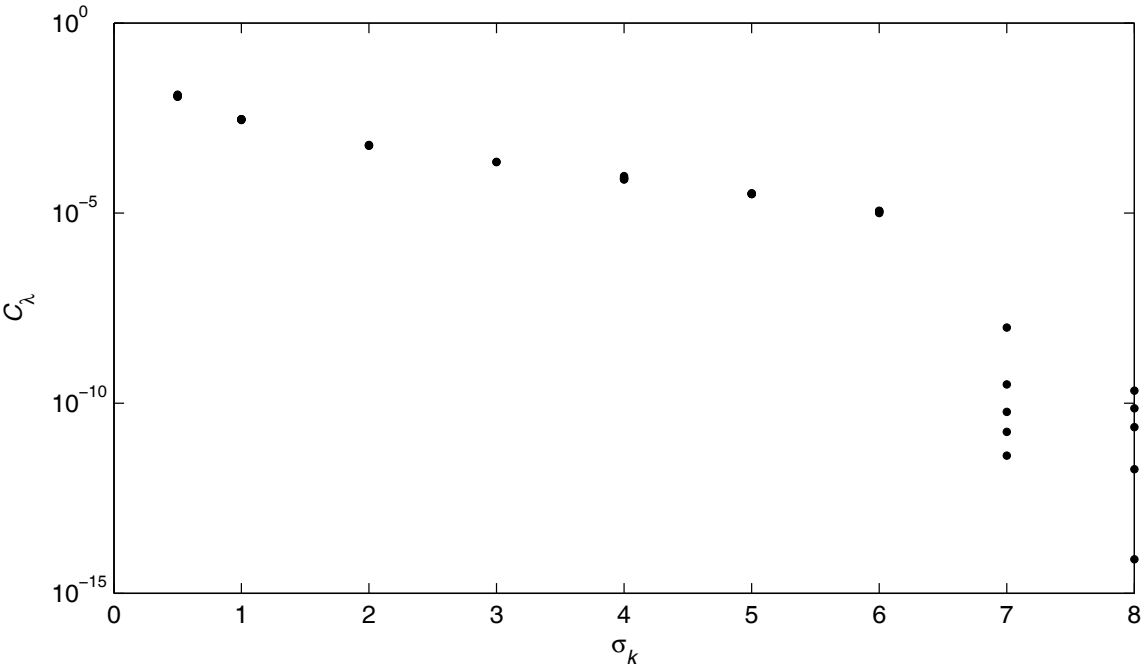

Fig. 3.6 Closure coefficient C_λ for different k^+_{wall} in the range [0.1 – 2.0] resulting from the optimization of C_λ, $A_{\lambda 0}$, $A_{\lambda 10}$, $A_{\lambda 11}$, $A_{\lambda 12}$, $A_{\lambda 13}$, $B_{\lambda 00}$, $B_{\lambda 01}$, $B_{\lambda 10}$, $B_{\lambda 11}$, $B_{\lambda 20}$ and $B_{\lambda 21}$ simultaneously.

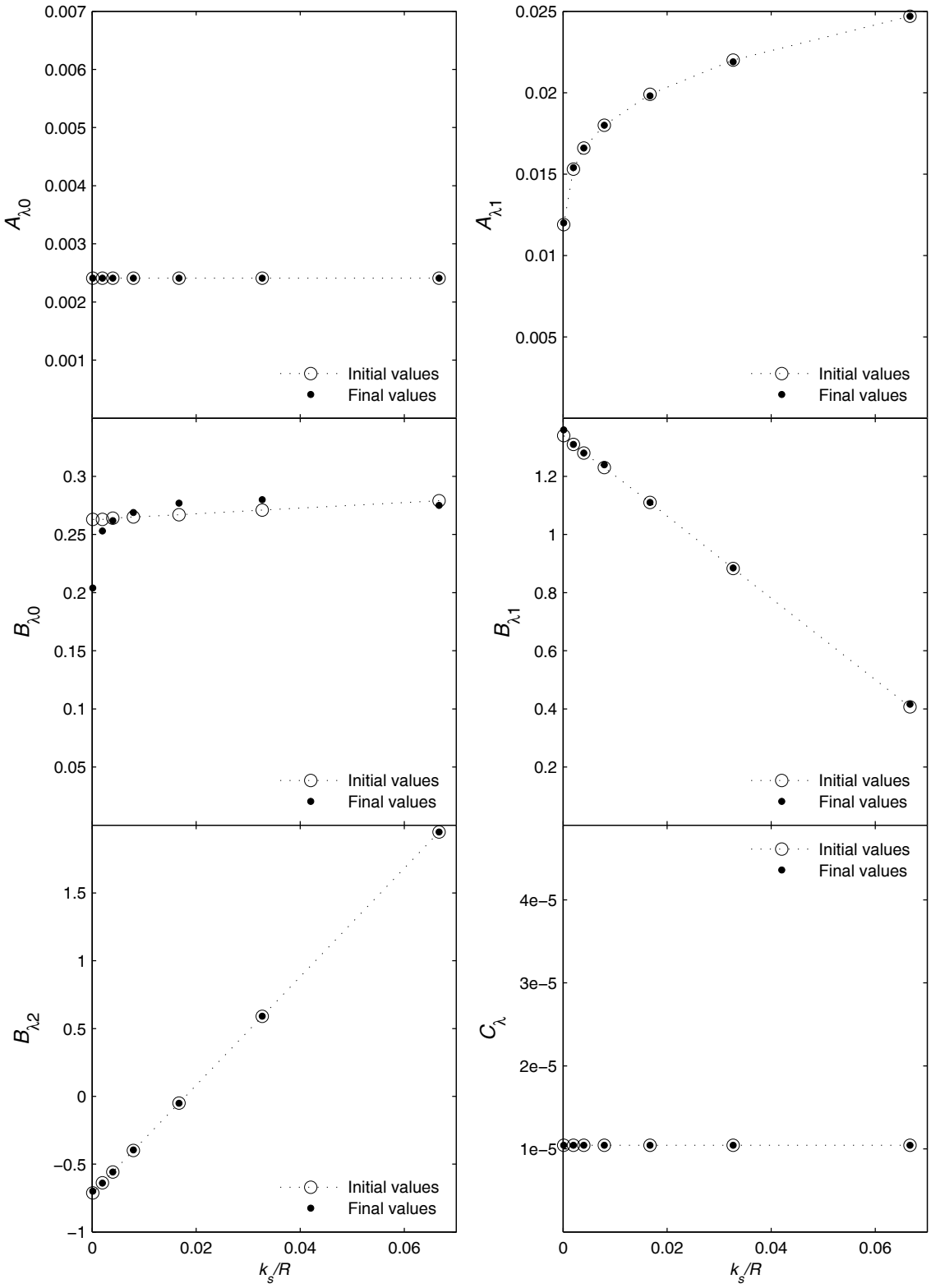

Fig. 3.7 Initial estimates and optimized $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ coefficients obtained at σ_k = 6 and $k^+_{\mathbf{wall}}$ = 0.1 based on the linear function for $B_{\lambda 0}$ given in Eq. (3.12).

Figure 3.8 shows the $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ coefficients before and after optimization. There are no noticeable changes before or after the optimization in any of the coefficients. Therefore, the functions proposed in Eqs. (3.11), (3.13), (3.14) and (3.16) can be used for the $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ coefficients respectively. Figure 3.8 was generated based on σ_k = 6 and k^+_{wall}= 0.1. Different values of σ_k in the range 2.0 to 6.0 and k^+_{wall} in the range 0.05 to 2.0 yield similar trends to that shown in Fig. 3.8.

The improvement of the function given in Eq. (3.16) compared to that given in (3.12) to model the coefficient $B_{\lambda 0}$ is shown by the fitness values of Fig. 3.9. This figure was generated at k^+_{wall} = 0.1. For any value of σ_k, the fitness parameter is decreased by at least 14%.

Using Eq. (3.16) for $B_{\lambda 0}$, the friction factor obtained from the k-λ turbulence model agrees well with the Nikuradse equation for fully rough flow. Figure 3.10 shows the friction factor compared to the Colebrook equation. This was generated for σ_k = 2.0, which has one of the worst fitness values, as seen in Fig. 3.9. The coefficients used to generate Fig. 3.8 are the same as those used to generate Fig. 3.10. Figure 3.10 was generated starting the roughness Reynolds number k_s^+ at 100. Equation (3.16) compared to Eq. (3.12) does not yield any noticeable differences on a friction factor figure. The open circles shown on Fig. 3.10 correspond to a friction factor obtained using the linear equation given in Eq. (3.12) for $B_{\lambda 0}$. The solid dots are based on the nonlinear equation given in Eq. (3.16) for $B_{\lambda 0}$.

There is a slight improvement in the velocity distribution when using the nonlinear relation for $B_{\lambda 0}$. Figure 3.12 shows the velocity distribution for a range of roughness ratios using the linear relation for $B_{\lambda 0}$. Figure 3.13 shows the velocity distribution for a range of roughness ratios using the nonlinear relation for $B_{\lambda 0}$. Both figures were generated for fully rough flow at a roughness Reynolds number of 80,000. Figures 3.12 and 3.13 used σ_k = 2.0 and all the same coefficients as those that were used to generate Fig. 3.10. From a comparison between Figs. 3.12 and 3.13, it can be seen that for low roughness ratios, the fit to the log law is slightly improved when using the nonlinear relation. Figure 3.11 shows a comparison between the log law and the velocity distribution obtained using σ_k = 2.0. From Figs. 3.10 and 3.11, it can be seen that the current k-λ turbulence model for fully rough flow predict the velocity and friction factors very accurately.

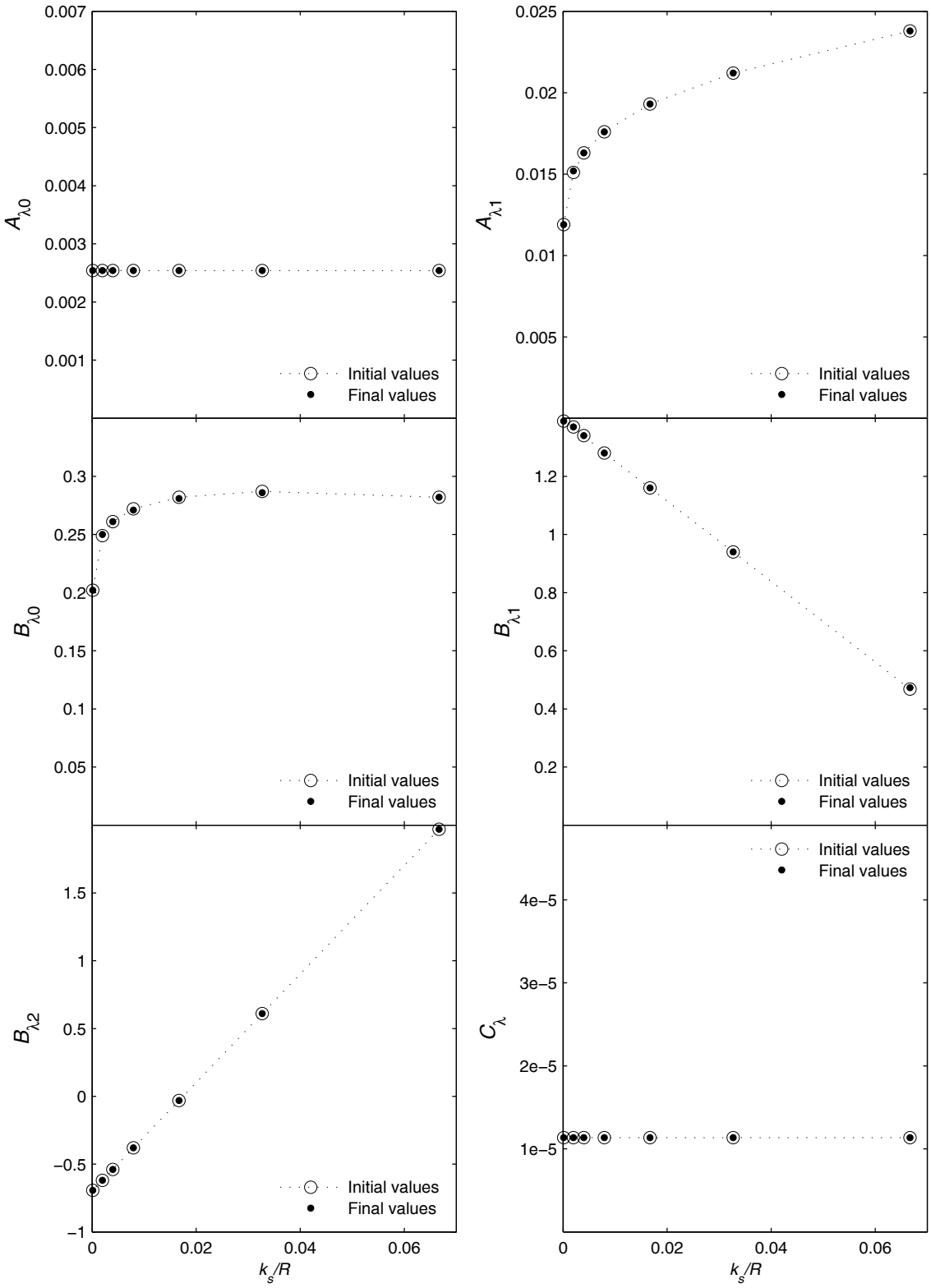

Fig. 3.8 Initial estimates and optimized $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ coefficients obtained at σ_k = 6 and k^+_{wall} = 0.1 based on the nonlinear function for $B_{\lambda 0}$ given in Eq. (3.16).

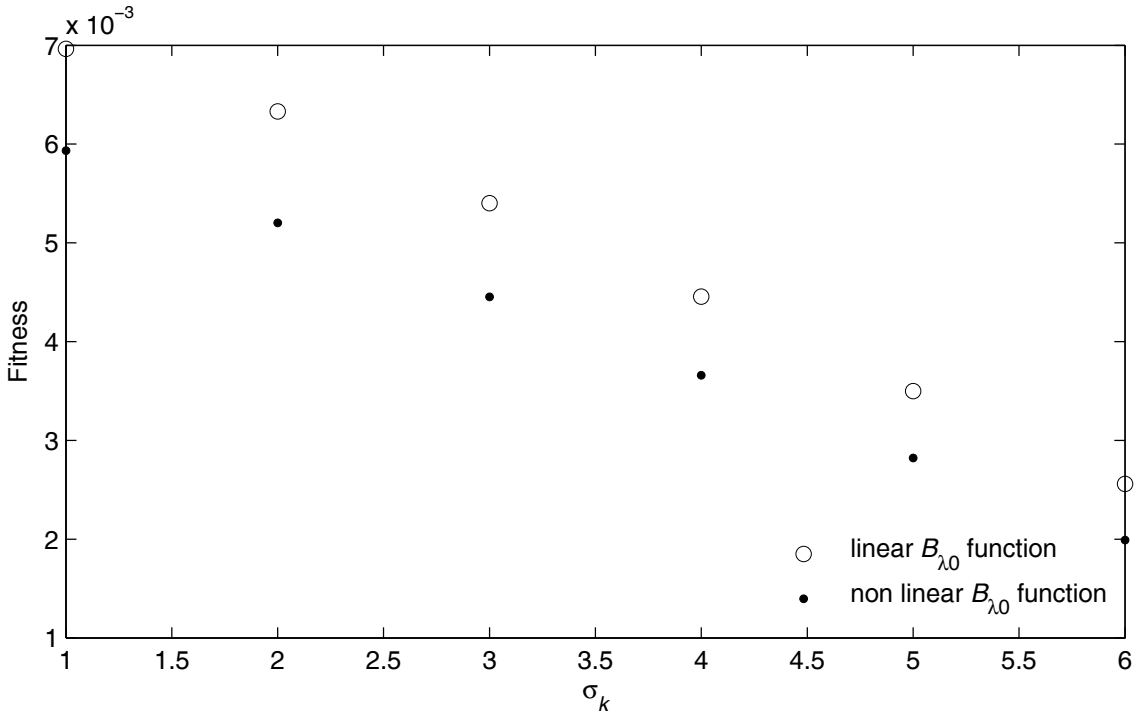

Fig. 3.9 Fitness comparison obtained from the linear and nonlinear relation for $B_{\lambda 0}$ given in Eqs. (3.12) and (3.16), based on k^{+}_{wall}= 0.1.

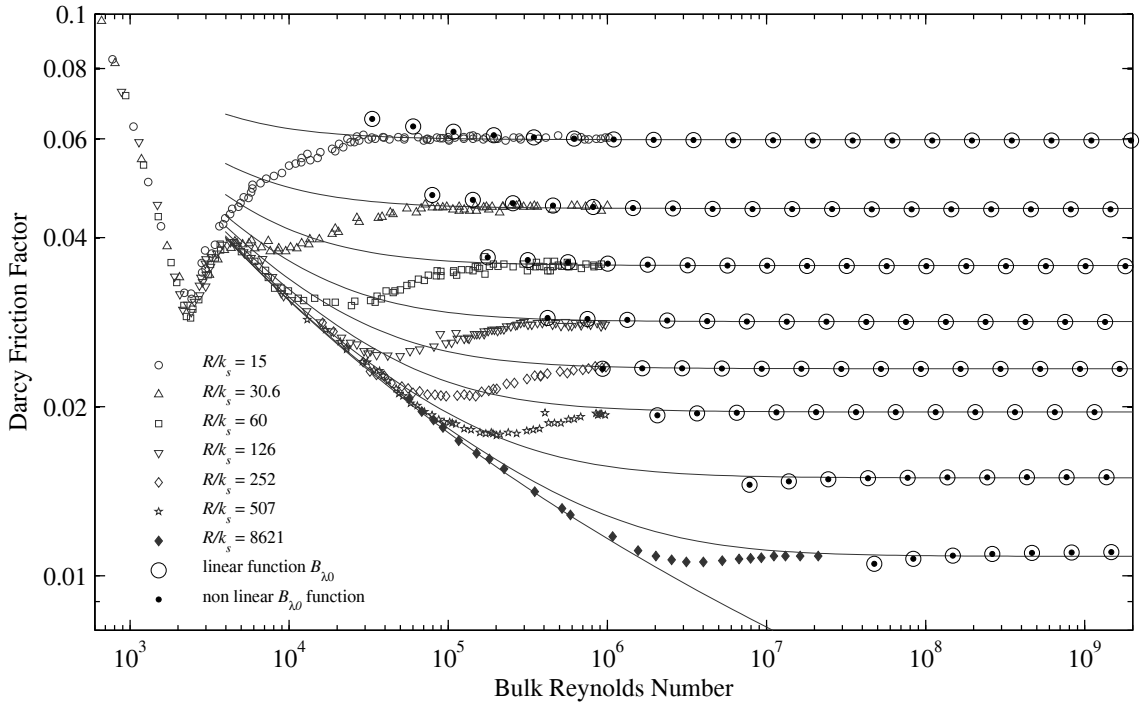

Fig. 3.10 Friction factor obtained from the linear and nonlinear relation for $B_{\lambda 0}$ given in Eqs. (3.12) and (3.16) and based on σ_k = 2 and k^{+}_{wall}= 0.1.

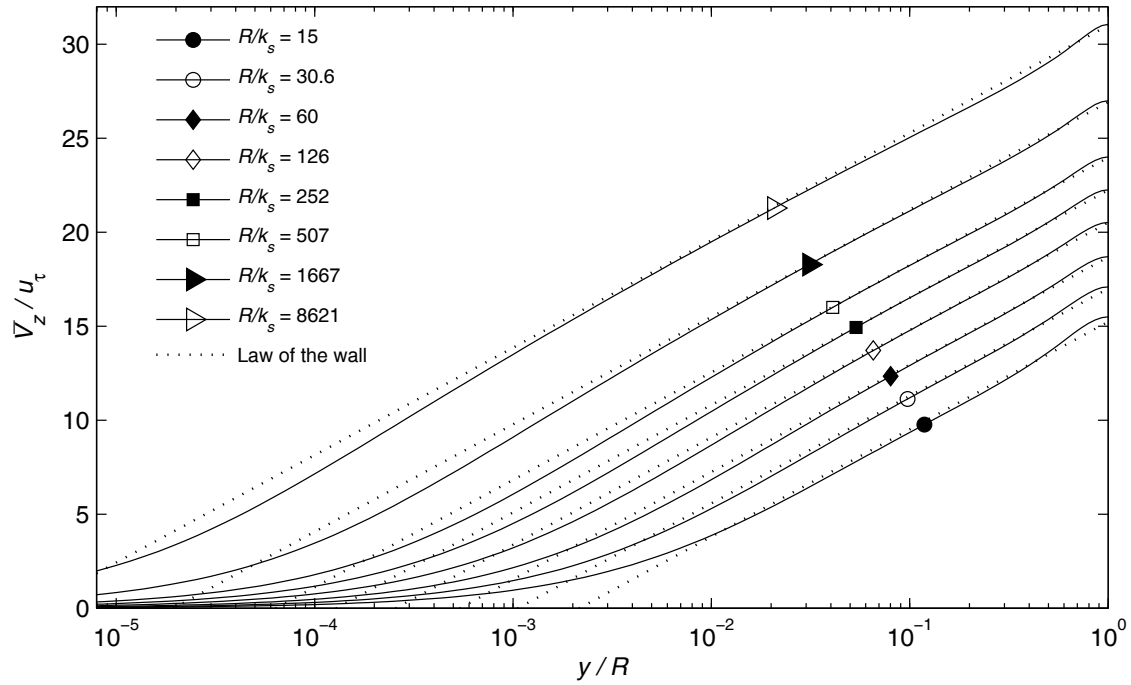

Fig. 3.11 Velocity distribution based on the nonlinear equation for $B_{\lambda 0}$ given in Eq. (3.16), using $\sigma_k = 2$ and $k^+_{\text{wall}} = 0.1$ and compared with the log law.

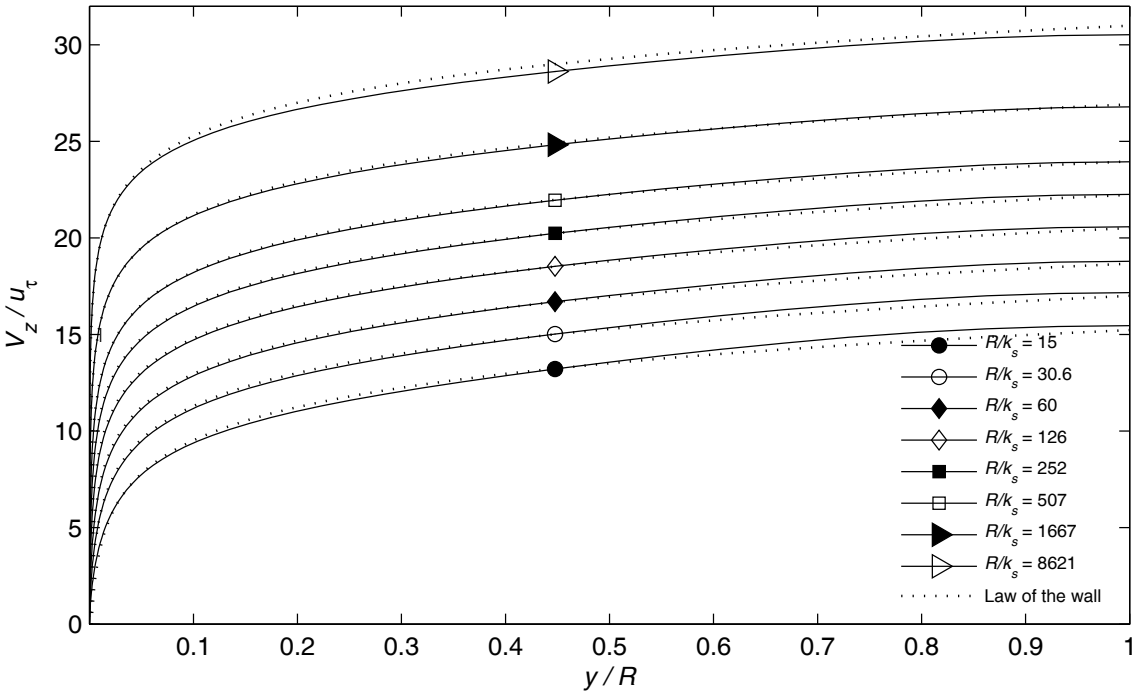

Fig. 3.12 Velocity distribution based on the linear equation for $B_{\lambda 0}$ given in Eq. (3.12) and using $\sigma_k = 2$ and $k^+_{\text{wall}} = 0.1$ and compared with a reference velocity distribution.

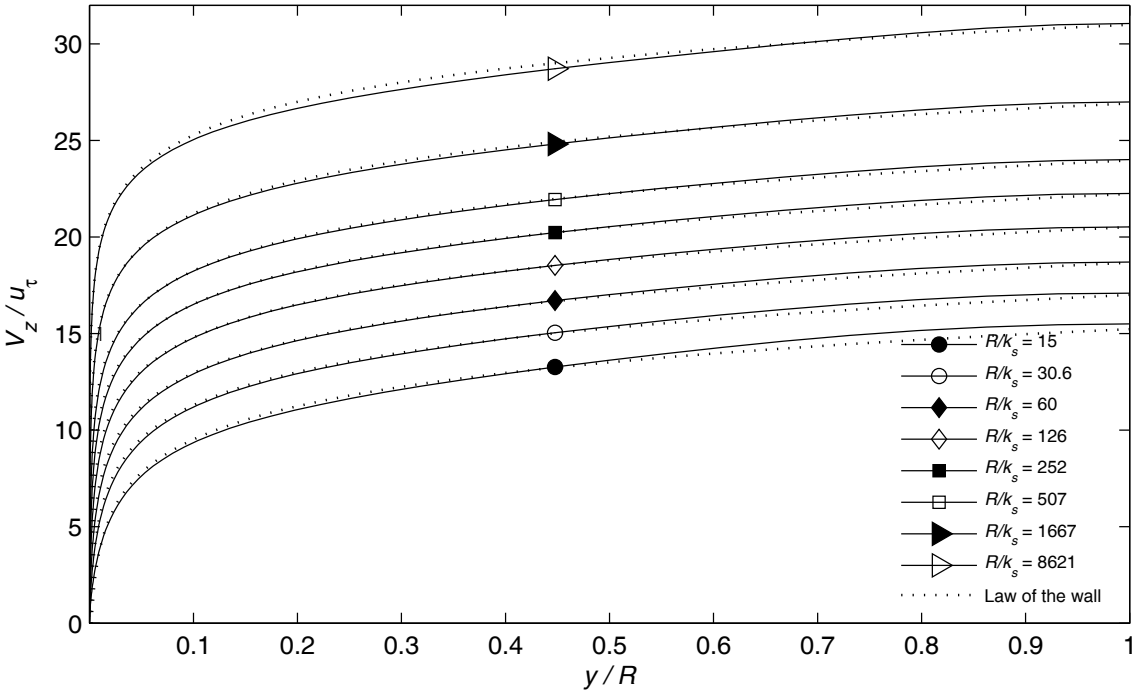

Fig. 3.13 Velocity distribution using the nonlinear equation for $B_{\lambda 0}$ given in Eq. (3.16), using σ_k = 2 and k^+_{wall}= 0.1 and compared with a reference velocity distribution.

B. Variations of the Model Coefficients with σ_k

A sixth order function is originally used to model the variation of the closure coefficient C_λ with respect to σ_k. As seen on Fig. 3.6, the closure coefficient C_λ showed very little dependency on the proportionality coefficient in the wall boundary condition. After iterating several times on all the coefficients to study the dependence of the coefficients with respect to σ_k, the polynomial function to model C_λ was reduced to a fourth order, and the range of σ_k was reduced to 2.0 to 6.0.

Figures 3.14–3.26 were generated using a turbulent kinetic energy k^+_{wall} = 0.05, 0.1, 0.5 and 1.0. Three coefficients $A_{\lambda 12}$, $B_{\lambda 02}$ and $B_{\lambda 20}$ are all functions of the proportionality coefficient in the wall boundary condition. All the other coefficients are assumed to be independent of the proportionality coefficient in the wall boundary condition, k^+_{wall}. A fourth order polynomial function was used to represent the variations of the coefficients to the closure coefficient σ_k. Because the coefficient σ_k is sampled five times at regular intervals in the range 2.0 to 6.0, the fourth order polynomial function passes through the average in k^+_{wall} at each of the five σ_k.

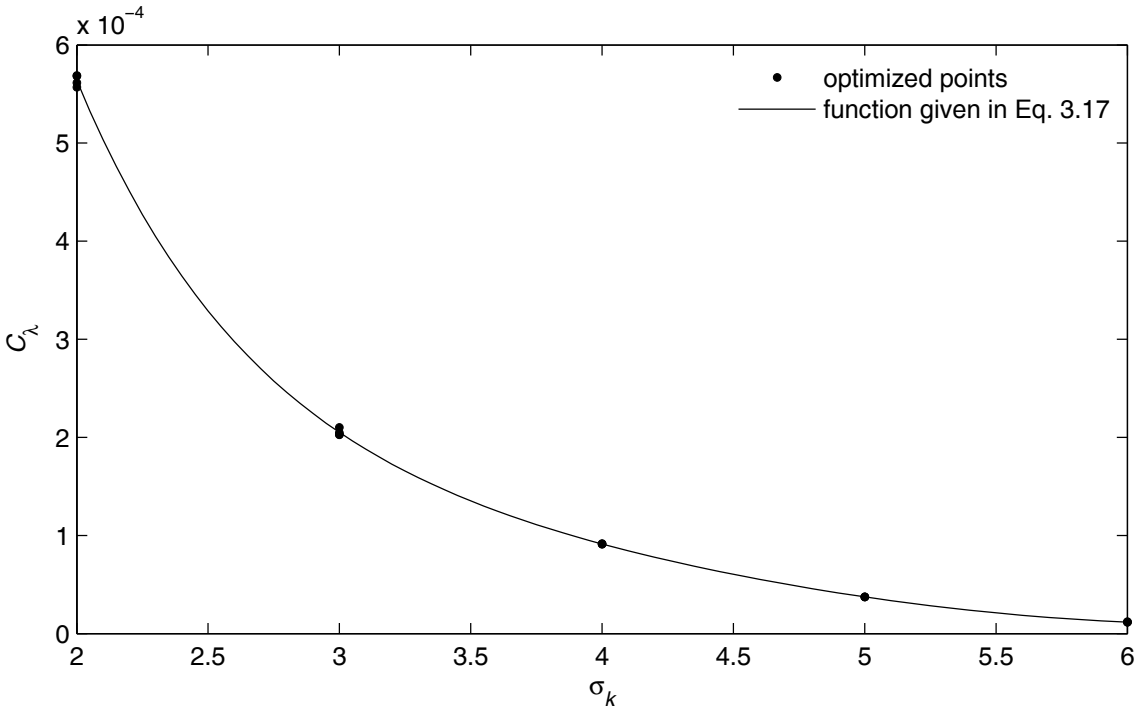

Fig. 3.14 Dependence of C_λ with σ_k for the current fully rough flow k-λ model.

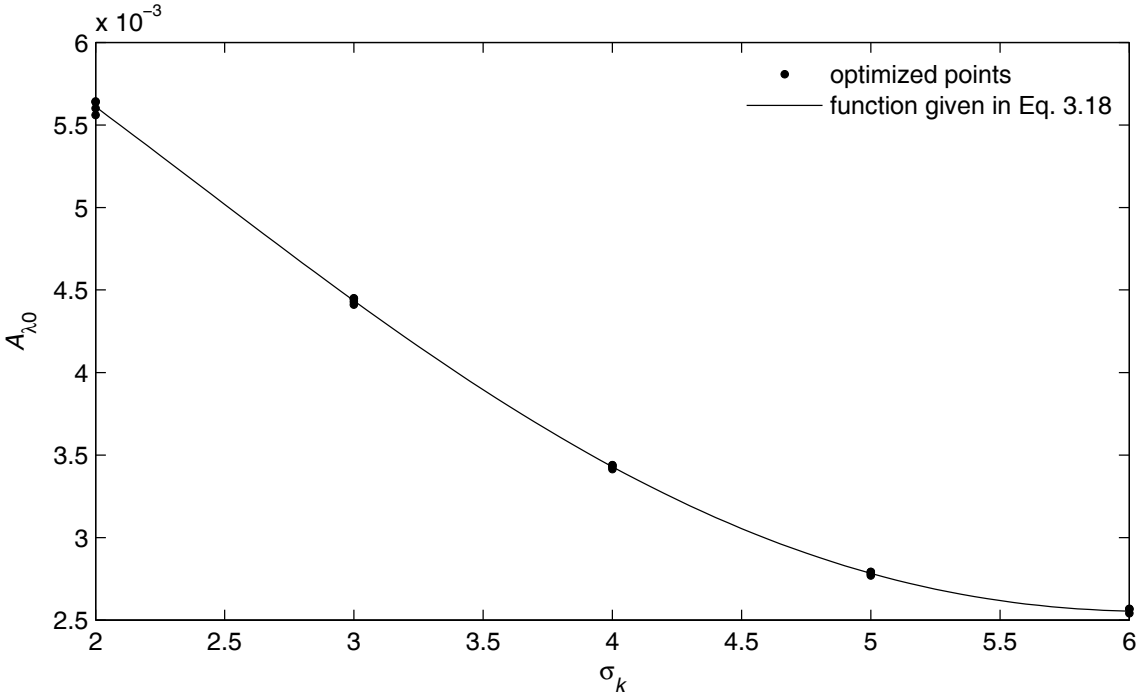

Fig. 3.15 Dependence of $A_{\lambda 0}$ with σ_k for the current fully rough flow k-λ model.

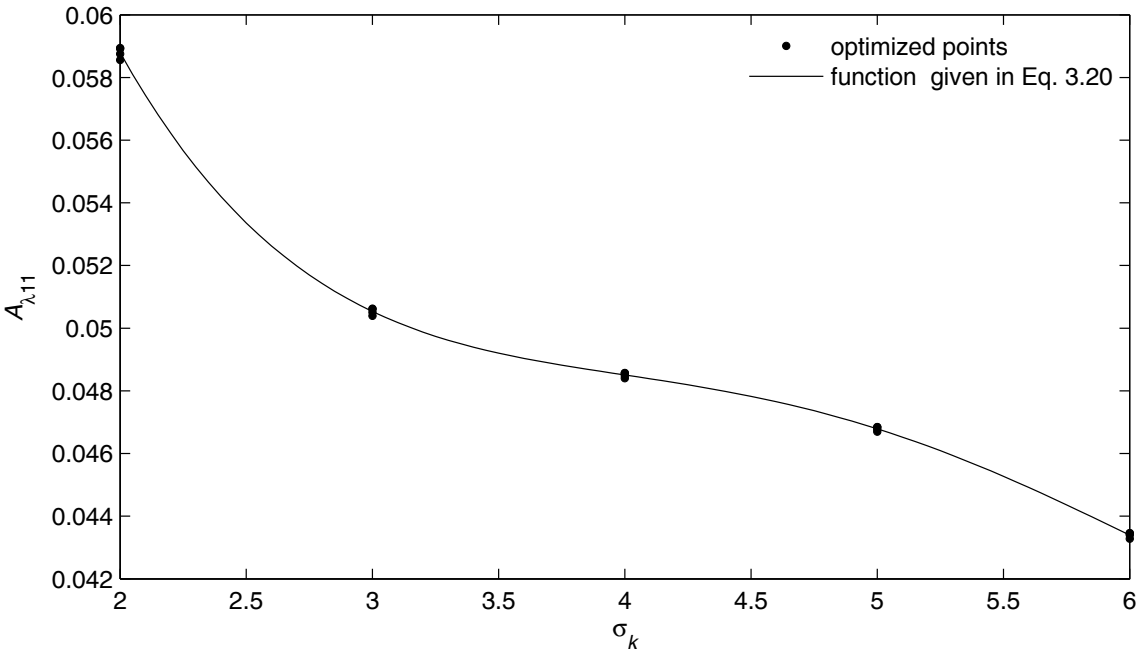

Fig. 3.16 Dependence of $A_{\lambda 11}$ with σ_k for the current fully rough flow k-λ model.

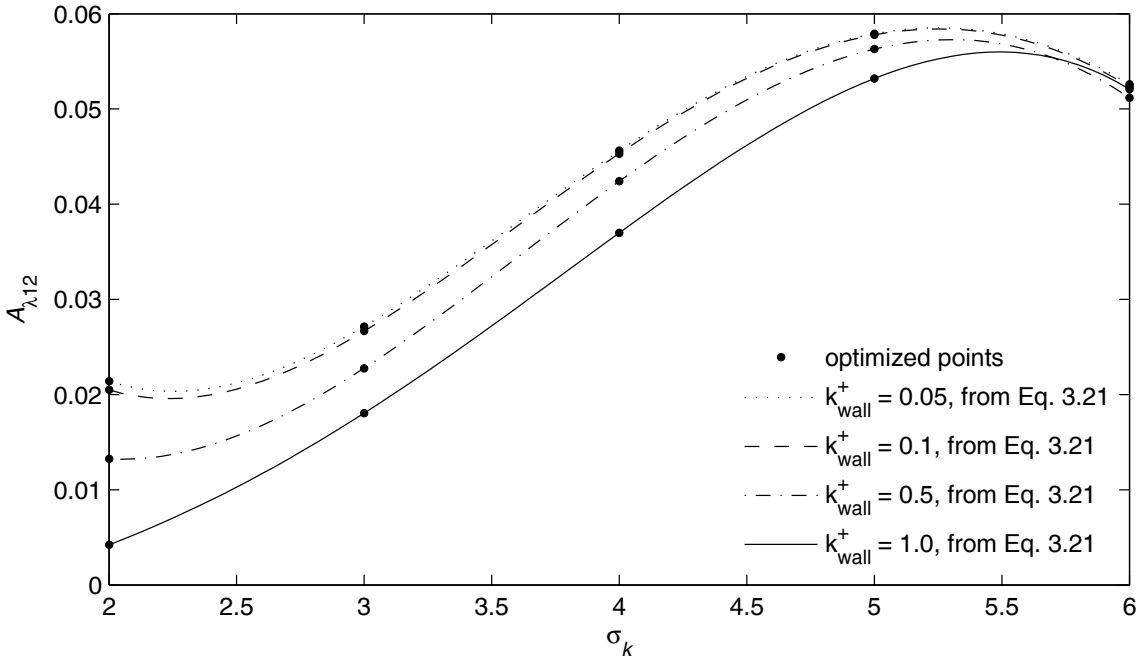

Fig. 3.17 Dependence of $A_{\lambda 12}$ with σ_k for the current fully rough flow k-λ model.

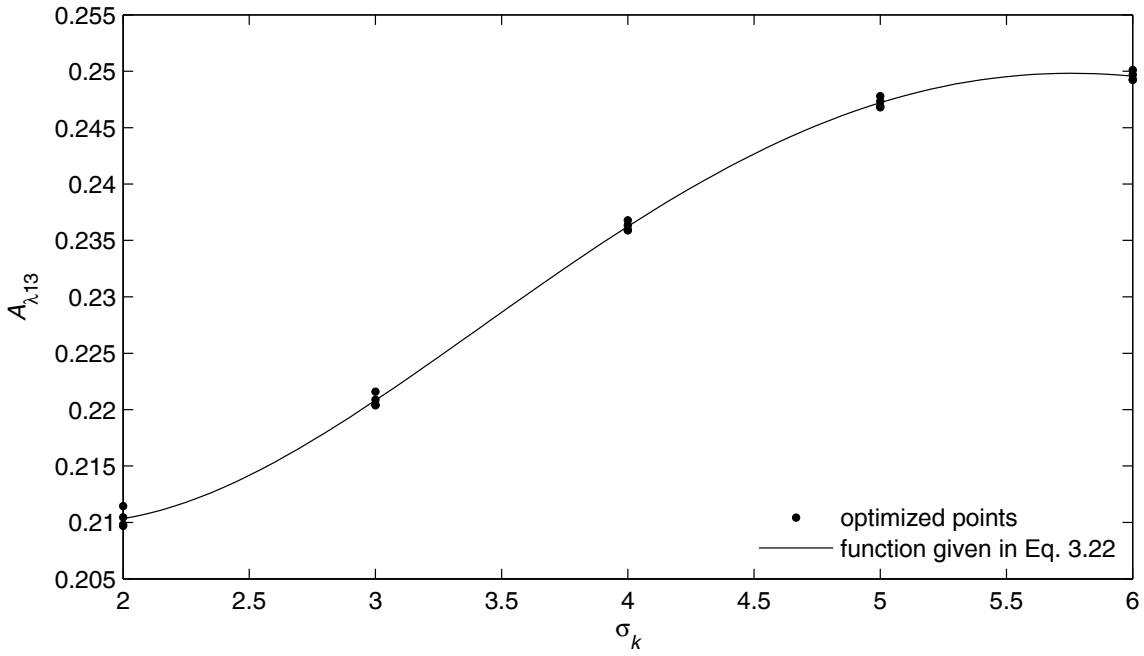

Fig. 3.18 Dependence of $A_{\lambda 13}$ with σ_k for the current fully rough flow k-λ model.

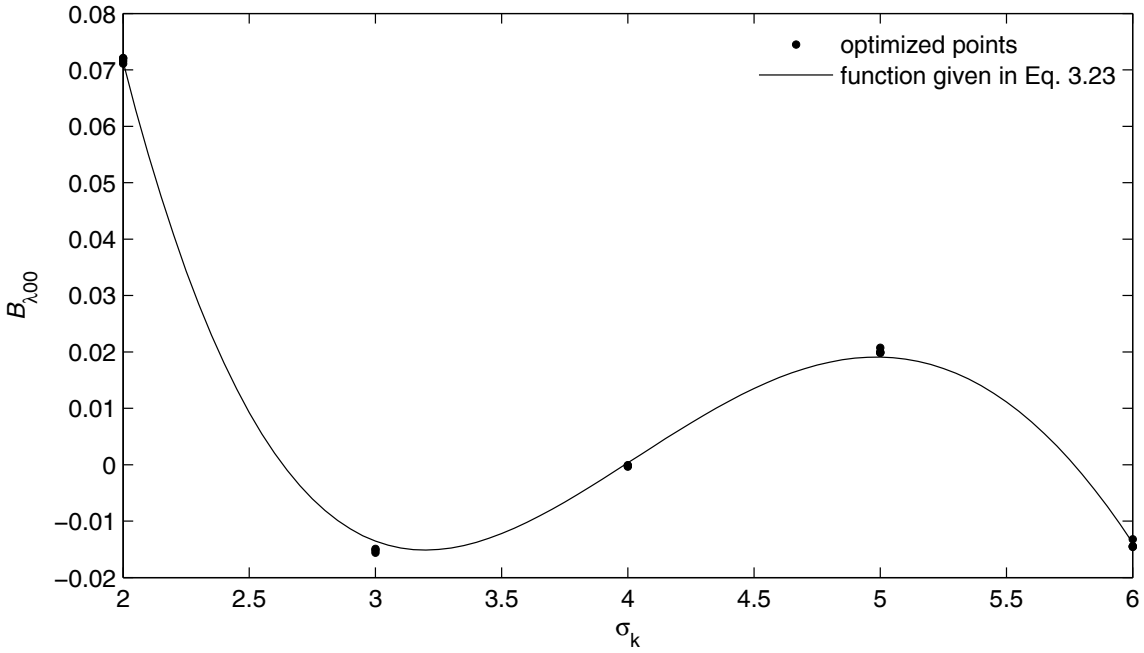

Fig. 3.19 Dependence of $B_{\lambda 00}$ with σ_k for the current fully rough flow k-λ model.

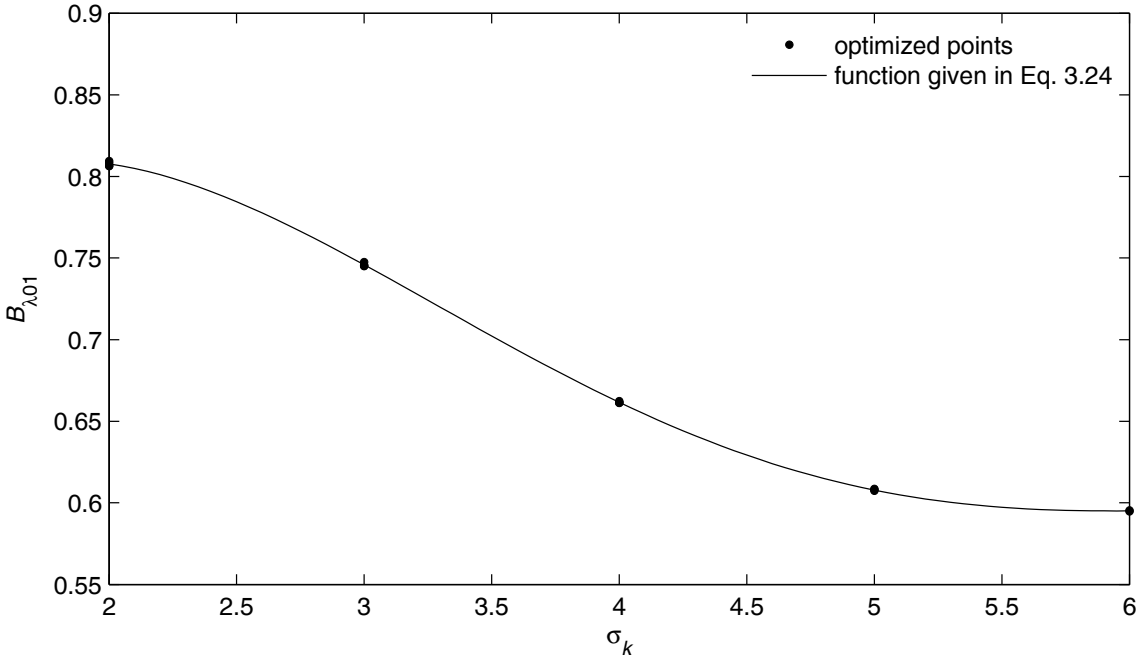

Fig. 3.20 Dependence of $B_{\lambda 01}$ with σ_k for the current fully rough flow k-λ model.

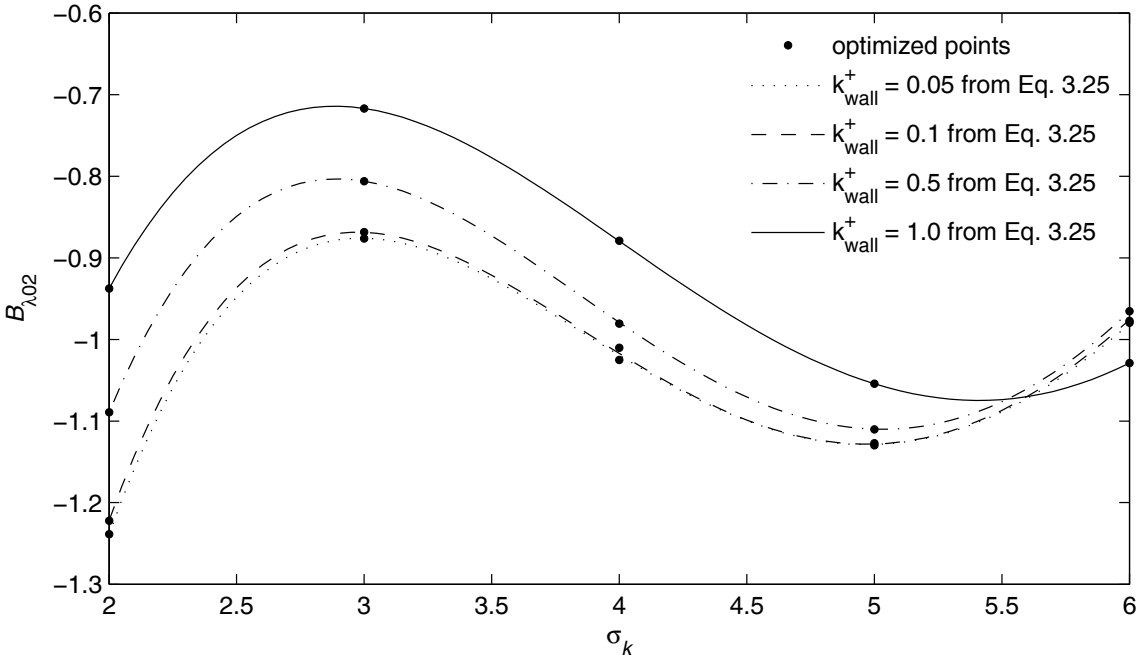

Fig. 3.21 Dependence of $B_{\lambda 02}$ with σ_k for the current fully rough flow k-λ model.

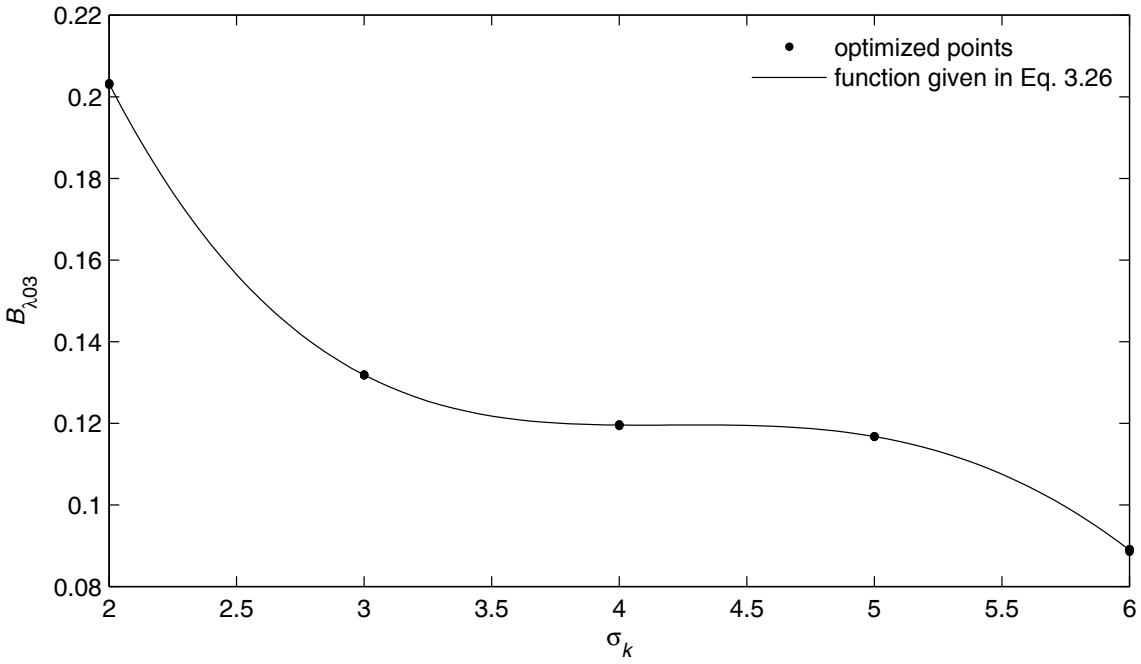

Fig. 3.22 Dependence of $B_{\lambda 03}$ with σ_k for the current fully rough flow k-λ model.

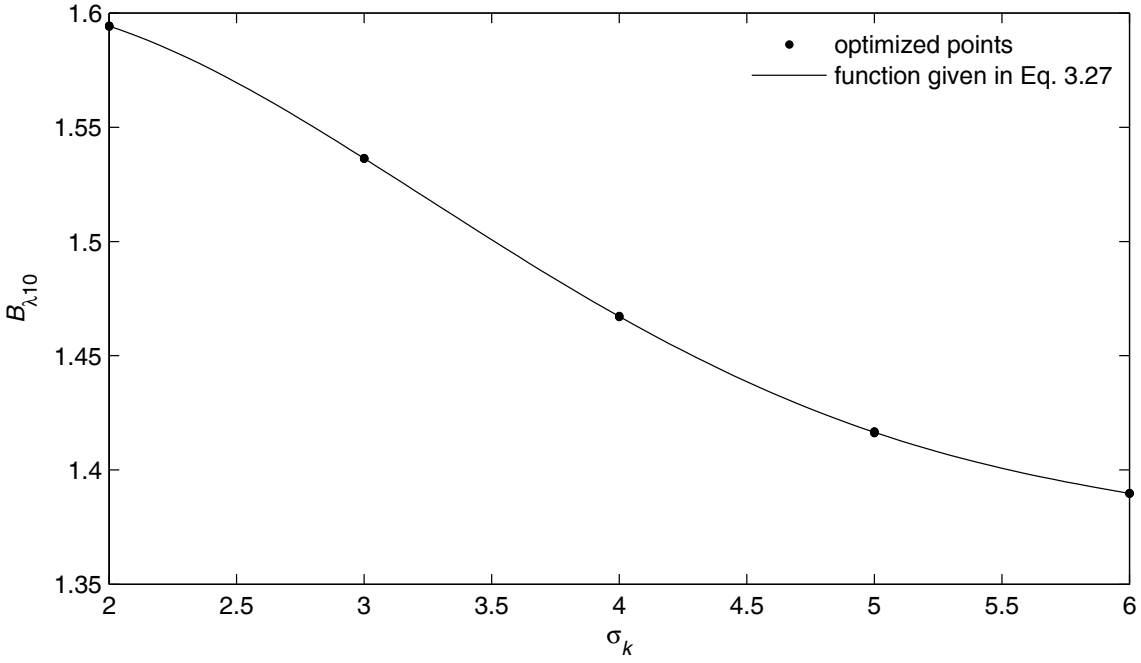

Fig. 3.23 Dependence of $B_{\lambda 10}$ with σ_k for the current fully rough flow k-λ model.

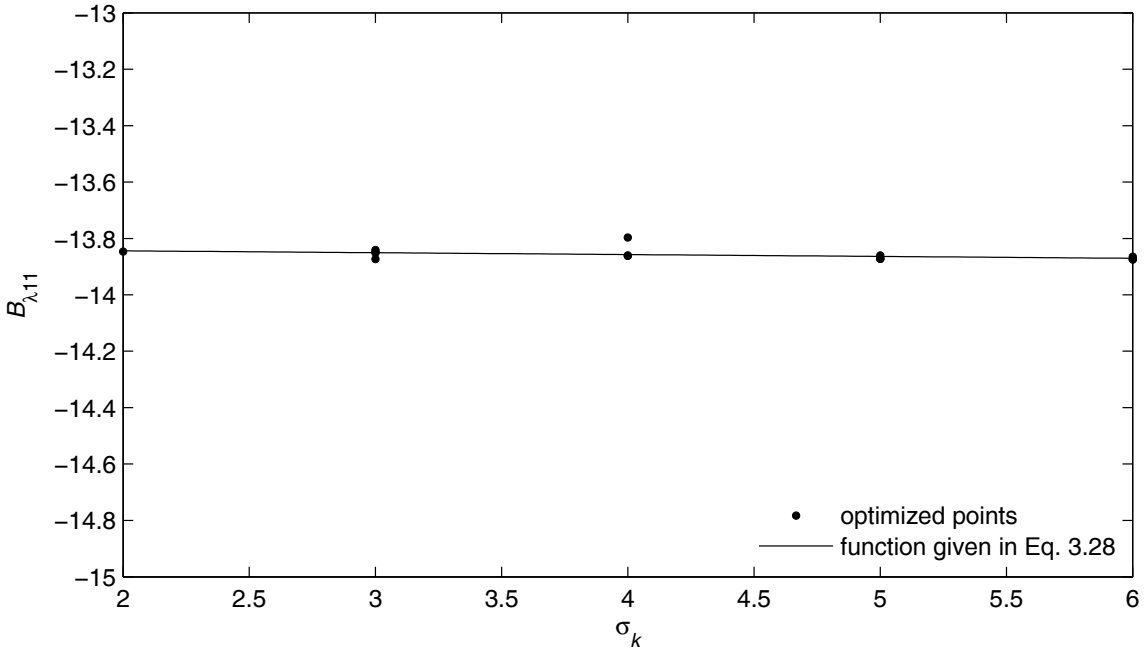

Fig. 3.24 Dependence of $B_{\lambda 11}$ with σ_k for the current fully rough flow k-λ model.

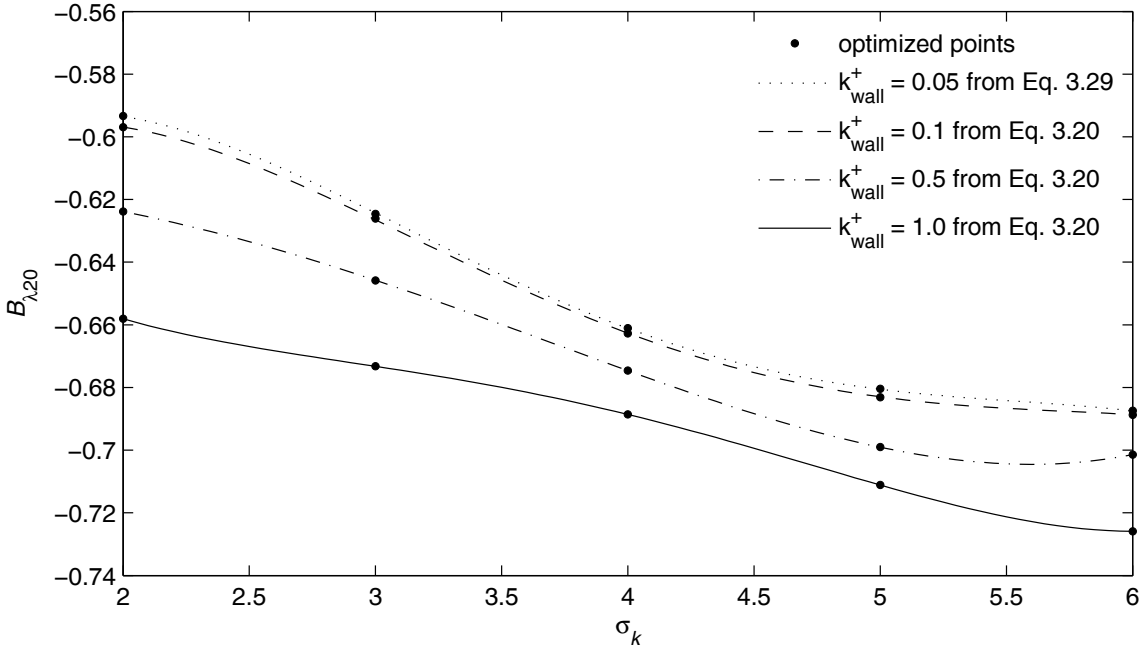

Fig. 3.25 Dependence of $B_{\lambda 20}$ with σ_k for the current fully rough flow k-λ model.

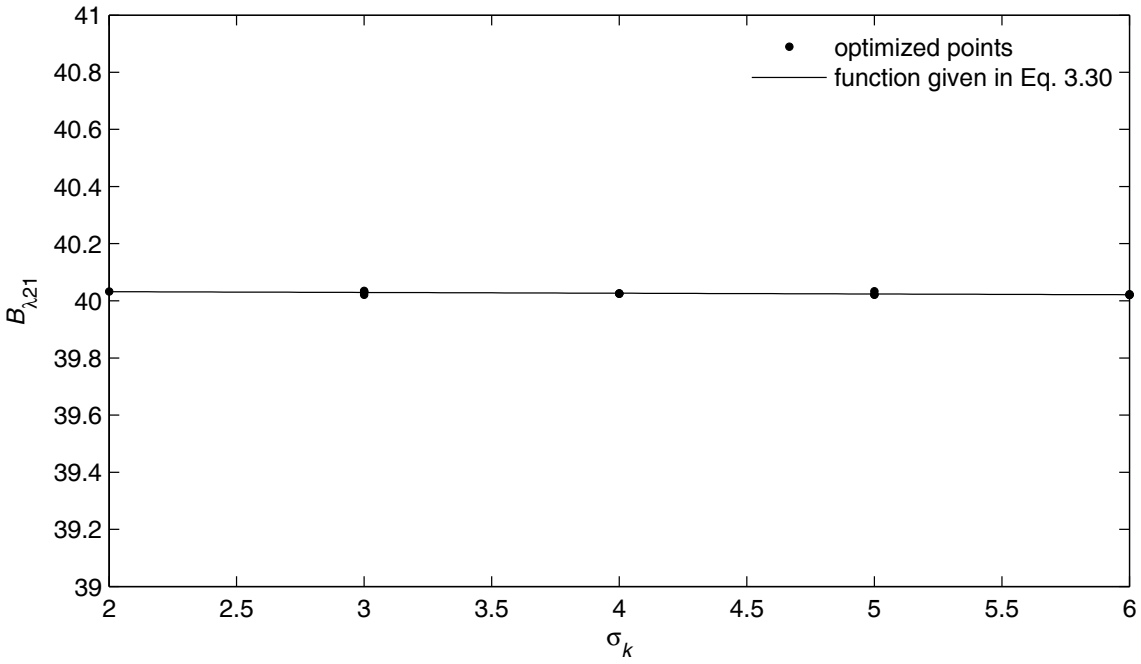

Fig. 3.26 Dependence of $B_{\lambda 21}$ with σ_k for the current fully rough flow k-λ model.

The following fourteen relations among the sixteen unknown constants were used to generate Figs. 3.14–3.26.

$$C_\lambda = \left(\begin{array}{l} 1.1430803817E\text{-}04\sigma_k^4 - 2.1568032682E\text{-}03\sigma_k^3 + 1.5454411906E\text{-}02\sigma_k^2 \\ - 5.3147683089E\text{-}02\sigma_k + 8.3648609370E\text{-}02 \end{array} \right)^2 \tag{3.17}$$

$$A_{\lambda 0} = - 5.4806699600E\text{-}06\sigma_k^4 + 1.0829889892E\text{-}04\sigma_k^3 - 5.8824903497E\text{-}04\sigma_k^2 + 6.4266598213E\text{-}05\sigma_k + 7.0559295507E\text{-}03 \tag{3.18}$$

$$A_{\lambda 10} = 6.0548405176E\text{-}06\sigma_k^4 - 1.7464041639E\text{-}04\sigma_k^3 + 1.7083135016E\text{-}03\sigma_k^2 - 7.8015697237E\text{-}03\sigma_k + 2.3620283331E\text{-}02 \tag{3.19}$$

$$A_{\lambda 11} = 1.6430210539E\text{-}04\sigma_k^4 - 3.2876048062E\text{-}03\sigma_k^3 + 2.3668508652E\text{-}02\sigma_k^2 - 7.4817705811E\text{-}02\sigma_k + 1.3742572373E\text{-}01 \tag{3.20}$$

$$\begin{aligned}
C_4 &= -3.6249854255\text{E-}04k_{\text{wall}}^{+\,2} - 2.3771783095\text{E-}04k_{\text{wall}}^{+} + 3.1811391789\text{E-}04 \\
C_3 &= 6.7456577268\text{E-}03k_{\text{wall}}^{+\,2} + 3.5031152723\text{E-}03k_{\text{wall}}^{+} - 7.6039769524\text{E-}03 \\
C_2 &= -4.3056163634\text{E-}02k_{\text{wall}}^{+\,2} - 2.0050955122\text{E-}02k_{\text{wall}}^{+} + 5.7370834148\text{E-}02 \\
C_1 &= 1.1098738361\text{E-}01k_{\text{wall}}^{+\,2} + 5.7323021287\text{E-}02k_{\text{wall}}^{+} - 1.5770938564\text{E-}01 \\
C_0 &= -9.7888362645\text{E-}02k_{\text{wall}}^{+\,2} - 7.6772775598\text{E-}02k_{\text{wall}}^{+} + 1.6397877418\text{E-}01 \\
A_{\lambda 12} &= C_4\,\sigma_k^4 + C_3\,\sigma_k^3 + C_2\,\sigma_k^2 + C_1\,\sigma_k + C_0
\end{aligned} \tag{3.21}$$

$$\begin{aligned}
A_{\lambda 13} = {} & 2.1775071643E\text{-}04\sigma_k^4 - 4.6125648020E\text{-}03\sigma_k^3 + 3.2004745006E\text{-}02\sigma_k^2 \\
& - 7.6059412219E\text{-}02\sigma_k + 2.6786227707E\text{-}01
\end{aligned} \tag{3.22}$$

$$\begin{aligned}
B_{\lambda 00} = {} & 1.5394390008E\text{-}03\sigma_k^4 - 3.7675845262E\text{-}02\sigma_k^3 + 3.0539951413E\text{-}01\sigma_k^2 \\
& - 9.9814632971E\text{-}01\sigma_k + 1.1231750962E+00
\end{aligned} \tag{3.23}$$

$$\begin{aligned}
B_{\lambda 01} = {} & -1.7611480539E\text{-}03\sigma_k^4 + 3.3472445299E\text{-}02\sigma_k^3 - 2.1562199190E\text{-}01\sigma_k^2 \\
& + 4.9483981773E\text{-}01\sigma_k + 4.4081967221E\text{-}01
\end{aligned} \tag{3.24}$$

$$\begin{aligned}
C_4 &= 4.5958241000\text{E-}03k_{\text{wall}}^{+\,2} + 2.0834333483\text{E-}03k_{\text{wall}}^{+} - 1.3277370846\text{E-}02 \\
C_3 &= -9.3116503886\text{E-}02k_{\text{wall}}^{+\,2} - 2.8569790820\text{E-}02k_{\text{wall}}^{+} + 2.7587164438\text{E-}01 \\
C_2 &= 6.0395589241\text{E-}01k_{\text{wall}}^{+\,2} + 1.8761243453\text{E-}01k_{\text{wall}}^{+} - 2.0079111202\text{E+}00 \\
C_1 &= -1.4946016082\text{E+}00k_{\text{wall}}^{+\,2} - 7.3780887177\text{E-}01k_{\text{wall}}^{+} + 6.0340918034\text{E+}00 \\
C_0 &= 1.2142113765\text{E+}00k_{\text{wall}}^{+\,2} + 1.2696368571\text{E+}00k_{\text{wall}}^{+} - 7.2873561012\text{E+}00 \\
B_{\lambda 02} &= C_4\,\sigma_k^4 + C_3\,\sigma_k^3 + C_2\,\sigma_k^2 + C_1\,\sigma_k + C_0
\end{aligned} \tag{3.25}$$

$$\begin{aligned}
B_{\lambda 03} = {} & 6.3453858956E\text{-}04\sigma_k^4 - 1.7156286122E\text{-}02\sigma_k^3 + 1.4904850864E\text{-}01\sigma_k^2 \\
& - 5.3187100685E\text{-}01\sigma_k + 7.9782349433E\text{-}01
\end{aligned} \tag{3.26}$$

$$\begin{aligned}
B_{\lambda 10} = {} & -1.0134626360E\text{-}03\sigma_k^4 + 1.9147367622E\text{-}02\sigma_k^3 - 1.2222170912E\text{-}01\sigma_k^2 \\
& + 2.5530412815E\text{-}01\sigma_k + 1.4355452292E+00
\end{aligned} \tag{3.27}$$

$$B_{\lambda 11} = -6.6206887937E\text{-}03\sigma_k - 1.3830728894E+01 \tag{3.28}$$

$$
\begin{aligned}
C_4 &= -2.0201876364E\text{-}03k_{\text{wall}}^{+\,2} + 4.3279392861E\text{-}03k_{\text{wall}}^{+} - 1.3949382336E-03 \\
C_3 &= 2.6570830671E\text{-}02k_{\text{wall}}^{+\,2} - 6.4136620678E\text{-}02k_{\text{wall}}^{+} + 2.3621757343E\text{-}02 \\
C_2 &= -1.1989031734E\text{-}01k_{\text{wall}}^{+\,2} + 3.3373664338E\text{-}01k_{\text{wall}}^{+} - 1.3864710895E\text{-}01 \\
C_1 &= 2.1942375688E\text{-}01k_{\text{wall}}^{+\,2} - 7.0814005373E\text{-}01k_{\text{wall}}^{+} + 3.0315634574E\text{-}01 \\
C_0 &= -1.4066477454E\text{-}01k_{\text{wall}}^{+\,2} + 4.5837394149E\text{-}01k_{\text{wall}}^{+} - 8.0853545298E\text{-}01 \\
B_{\lambda 20} &= C_4\,\sigma_k^4 + C_3\,\sigma_k^3 + C_2\,\sigma_k^2 + C_1\,\sigma_k + C_0
\end{aligned}
\tag{3.29}
$$

$$
B_{\lambda 21} = -2.6815510578\text{E-}03\sigma_k + 4.0037296283\text{E+}01 \tag{3.30}
$$

The ten constants $A_{\lambda 10}$, $A_{\lambda 11}$, $A_{\lambda 12}$, $A_{\lambda 13}$, $B_{\lambda 0}$, $B_{\lambda 01}$, $B_{\lambda 10}$, $B_{\lambda 11}$, $B_{\lambda 20}$ and $B_{\lambda 21}$ are related to the four coefficients $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$, $B_{\lambda 2}$ according to

$$
A_{\lambda 1} = A_{\lambda 11} + A_{\lambda 12}\hat{k}_s\left(A_{\lambda 10} - A_{\lambda 11} - A_{\lambda 12}\hat{k}_s\right)\exp\left(-\hat{k}_s^{\,A_{\lambda 13}}\right) \tag{3.31}
$$

$$
B_{\lambda 0} = B_{\lambda 01} + B_{\lambda 02}\hat{k}_s\left(B_{\lambda 00} - B_{\lambda 01} - B_{\lambda 02}\hat{k}_s\right)\exp\left(-\hat{k}_s^{\,B_{\lambda 03}}\right) \tag{3.32}
$$

$$
B_{\lambda 1} = B_{\lambda 10} + B_{\lambda 11}\hat{k}_s \tag{3.33}
$$

$$
B_{\lambda 2} = B_{\lambda 20} + B_{\lambda 21}\hat{k}_s \tag{3.34}
$$

The mean vortex wavelength is given by

$$
\hat{\lambda} \equiv \frac{\lambda}{R} = \left(A_{\lambda 0}\hat{k}_s + A_{\lambda 1}\hat{y}\right)\left(1 - \frac{1}{2}\hat{y}\right)\left[B_{\lambda 0} + B_{\lambda 1}\hat{r}^2 + B_{\lambda 2}\hat{r}^4 + \left(1 - B_{\lambda 0} - B_{\lambda 1} - B_{\lambda 2}\right)\hat{r}^6\right] \tag{3.35}
$$

The coefficient $A_{\lambda 0}$ was chosen to be a constant. In fact, for fully rough flow, the wall value of the turbulent kinetic energy should be a constant. The mean vortex wavelength at the wall depends only on the roughness ratio. This requires the coefficient $A_{\lambda 0}$ to be a constant.

The coefficients were optimized one at a time. Hundreds of iterations were performed on the coefficients $A_{\lambda 0}$, $A_{\lambda 10}$, $A_{\lambda 11}$, $A_{\lambda 12}$, $A_{\lambda 13}$, $B_{\lambda 00}$, $B_{\lambda 01}$, $B_{\lambda 02}$, $B_{\lambda 03}$, $B_{\lambda 10}$, $B_{\lambda 1}$, $B_{\lambda 20}$, $B_{\lambda 21}$, C_λ in order to obtain the relations given in Eqs. (3.17)–(3.30). See Appendix E for details.

C. Fitness Value for the Complete Model

The fitness parameter for the complete model is shown in Fig. 3.27. This fitness parameter quantifies how well the model matches empirical relation established for fully rough pipe flows. The fitness parameter was obtained based on the relations given in Eqs. (3.17)–(3.34). This figure can be compared to Fig. 3.9. As an example at $\sigma_k = 2.0$, the fitness shown in Fig. 3.9 was 0.00520. For this same σ_k, the fitness shown in Fig. 3.27 is 0.00527. The increase comes from the fact that the exact optimal values for all the coefficients involved in the turbulence model were used to generate Fig. 3.9 whereas a function was used to generate Fig. 3.27. The difference in the fitness parameter between Figs. 3.9 and 3.27 is so small that it is not noticeable on the friction factor and velocity distribution figures.

The total fitness is the sum of the fitness for the friction factor reference equation and the fitness for the velocity distribution reference function. These fitnesses are obtained from a RMS of a reference function. The fitness for the velocity distribution and the fitness for the friction factor are shown in Fig. 3.28. The fitness of the velocity distribution is better than that for the friction factor. At $\sigma_k = 4.0$, the fitness to the friction factor equation is 0.536% whereas the fitness to the velocity distribution is 0.223%, which is still excellent.

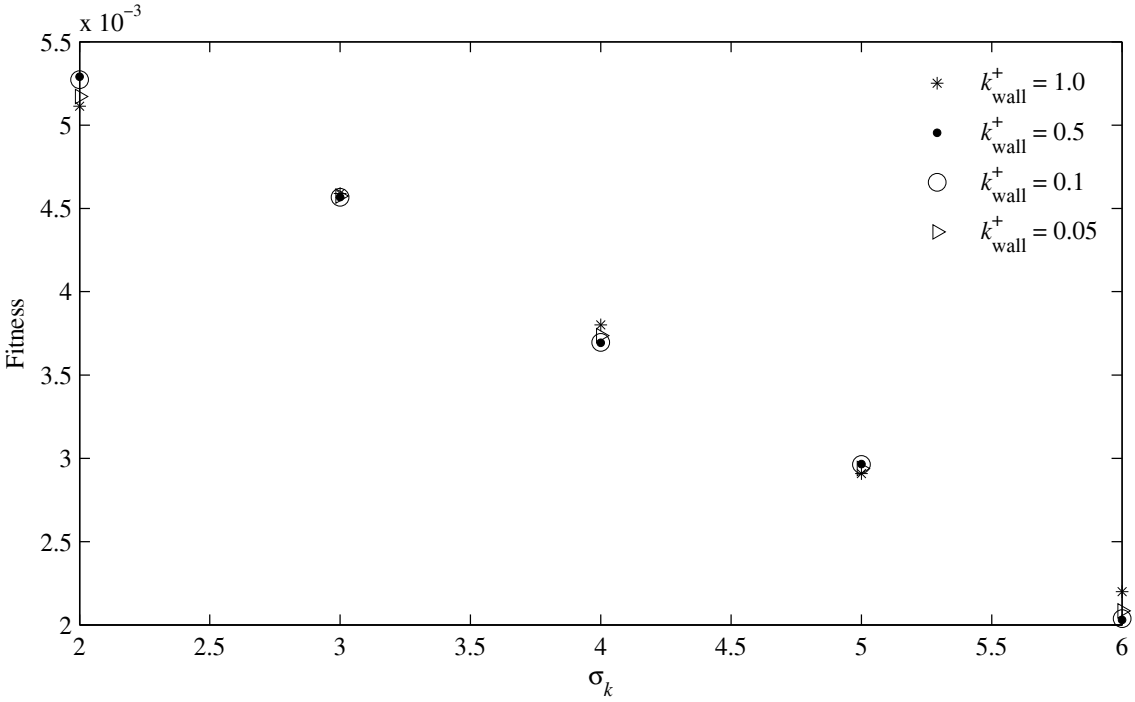

Fig. 3.27 Total fitness parameter versus σ_k.

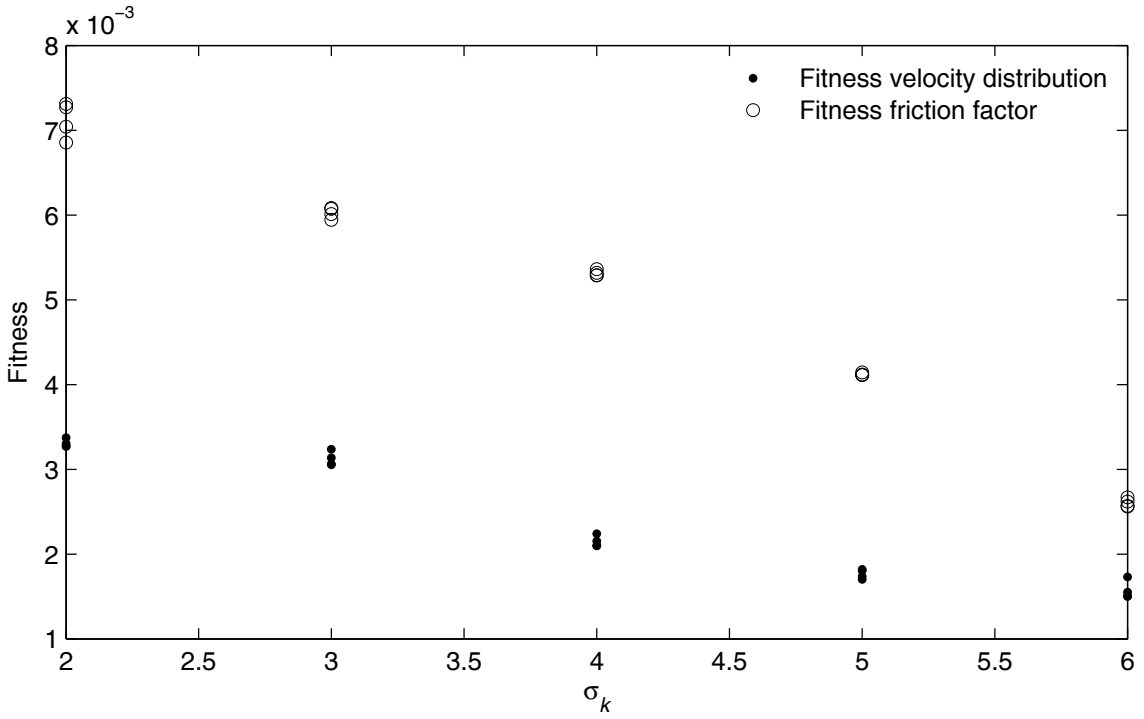

Fig. 3.28 Velocity distribution and friction factor fitnesses over a range of σ_k.

IV. Summary and Conclusion

The turbulent-kinetic-energy transport equation proposed by Phillips contains two closure coefficients, σ_k and C_λ, which should be dimensionless universal constants. It has been shown here that excellent agreement with experimental data for fully rough pipe flow can be attained over the range of about $2 < \sigma_k < 6$ and $0.00001 < C_\lambda < 0.00057$, provided that the relation between σ_k and C_λ, which is given in Eq. (3.17) and shown in Fig. 3.14 is maintained.

In addition, the turbulent-kinetic-energy transport equation requires a wall boundary condition for the specific turbulent kinetic energy. In general, the specific turbulent kinetic energy at a rough surface should be proportional to the square of the friction velocity,

$$k\big|_{r=R} = k^{+}_{\text{wall}} u_\tau^2 \tag{3.36}$$

where k_{wall}^{+} is a dimensionless proportionality coefficient, which is expected to be a unique function of the roughness Reynolds number, $k_s u_\tau / \nu$. However, by definition, fully rough flow occurs when the roughness Reynolds number is high enough so that the solution becomes independent of molecular viscosity. Hence, for fully rough pipe flow, the proportionality coefficient, k_{wall}^{+}, must be another dimensionless universal constant associated with the turbulence model. It has been shown here that excellent agreement with experimental data for fully rough pipe flow can be attained over the range of about $0.05 < k_{\text{wall}}^{+} < 1.0$, provided that certain algebraic relations are maintained between the mean vortex wavelength, λ, and the proportionality coefficient, k_{wall}^{+}.

From the discussion above, it is important to recognize a significant result that was developed in this chapter. Excellent agreement with experimental data for fully rough pipe flow can be attained over a range of model constants, which are σ_k, C_λ and the fully rough limit for k_{wall}^{+}. In terms of future development, this is fortunate, because it provides a great deal of flexibility that can be used when tuning the model to agree with experimental data for other turbulent flows.

The empirical relation obtained in Chapter 2 for the mean vortex wavelength contains five unknown closure coefficients, $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$. Unlike the closure coefficients in the turbulent-kinetic-energy transport equation, the closure coefficients in the algebraic relation for the mean vortex wavelength, which is given in Eq. (3.3), need not be universal constants. Because Eq. (3.3) was used simply to develop an empirical correlation to fit experimental data, the only restriction that must be placed on $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ is that these closure coefficients may not depend on the radial position within the pipe. In general, the turbulent eddy viscosity and mean vortex wavelength at a rough surface should be proportional to the equivalent sand-grain roughness height. Therefore, the coefficient $A_{\lambda 0}$ should be a constant. It has been shown that excellent agreement with experimental data for fully rough pipe flow can be attained, if $A_{\lambda 0}$ is a constant and $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$ and $B_{\lambda 2}$ are functions of the roughness ratio, k_s / R, as given by Eqs. (3.31)–(3.34). Combining the constant $A_{\lambda 0}$ with the 12 constants in Eqs. (3.31)–(3.34), there are 13 dimensionless constants, $A_{\lambda 0}$, $A_{\lambda 10}$, $A_{\lambda 11}$, $A_{\lambda 12}$, $A_{\lambda 13}$, $B_{\lambda 00}$, $B_{\lambda 01}$, $B_{\lambda 02}$, $B_{\lambda 03}$, $B_{\lambda 10}$, $B_{\lambda 11}$, $B_{\lambda 20}$ and $B_{\lambda 21}$, that must be known to determine the variation in the mean vortex wavelength over the pipe radius.

Combining the proposed turbulent-kinetic-energy transport equation and boundary conditions with the proposed algebraic relation for the mean vortex wavelength, there are a total of 16 unknown model constants for the case of fully rough pipe flow, σ_k, C_λ, k^+_{wall}, $A_{\lambda 0}$, $A_{\lambda 10}$, $A_{\lambda 11}$, $A_{\lambda 12}$, $A_{\lambda 13}$, $B_{\lambda 00}$, $B_{\lambda 01}$, $B_{\lambda 02}$, $B_{\lambda 03}$, $B_{\lambda 10}$, $B_{\lambda 11}$, $B_{\lambda 20}$ and $B_{\lambda 21}$. Using an optimization code, 14 algebraic relations were established among these 16 constants. The determination of these 14 algebraic relations was based on minimizing a fitness parameter, which assesses how close the model solutions are to a target solution set. The target solution set was based on a friction factor and a velocity distribution comparison to algebraic relations obtained for fully rough flows. The friction factor was compared to the Nikuradse equation, which is the fully rough flow limit of the Colebrook equation. The velocity distribution was compared to an algebraic relation derived using a deviation from the log law in order to satisfy the symmetry centerline boundary condition and the no-slip wall boundary condition.

The minimization of the fitness parameter is based on a quasi-Newton, gradient-based algorithm. This algorithm uses the BFGS update method. Hundreds of optimization cases were run over a wide range of values for the closure coefficients. From those optimization cases, functions were derived for the five coefficients of the mean vortex wavelength; $A_{\lambda 0}$, $A_{\lambda 1}$, $B_{\lambda 0}$, $B_{\lambda 1}$, $B_{\lambda 2}$. These coefficients were found to depend on the roughness ratio $\hat{k}_s$, the proportionality coefficient k^+_{wall} and the closure coefficient σ_k.

Excellent agreement with well established relations for fully rough pipe flow can be obtained provided that the relations between the model coefficients are maintained. The fully rough flow model shows that a good fitness can be obtained for any value of σ_k in the range 2.0 to 6.0 and any value of the k^+_{wall} in the range 0.05 to 1.0. The RMS error is 0.511% at σ_k = 2.0 and 0.203% at σ_k = 6.0, when the relations between the model coefficients are maintained. The choice of the closure coefficients σ_k and k^+_{wall} could be decided based on lower roughness Reynolds numbers because fully rough pipe flow gives good results over a range of values for σ_k and k^+_{wall}, provided that the relations between the other coefficients are maintained.

The model needs to be compared to other turbulence models, which support rough flow. This will be presented in the next chapter. The current model gives excellent agreement with the velocity distribution and friction factor. Because there are no data available for fully rough pipe flow for the turbulent kinetic energy and the second turbulent variable, no comparison is possible. The distribution of the fluctuating vorticity and the mean turbulent wavelength will be presented in the next chapter.

CHAPTER 4

PHILLIPS *K-Λ* TURBULENCE MODEL RESULTS AND CONCLUSIONS

I. Introduction

The closure coefficients associated with the present *k-λ* one-equation turbulence model for fully rough pipe flow were evaluated by minimizing a fitness parameter in order to match algebraic relations obtained for the velocity distributions and the friction factor. It was shown that excellent agreement with experimental data could be obtained over ranges of model constants, which are $2 < \sigma_k < 6$, $0.00001 < C_\lambda < 0.00057$ and $0.05 < k^+_{\text{wall}} < 1.0$, provided that certain algebraic relations are maintained among the mean vortex wavelength, λ, the two closure coefficients C_λ and σ_k and the proportionality coefficient, k^+_{wall}.

During the optimization process, the mean vortex wavelength, turbulent kinetic energy and root-mean-square fluctuating vorticity were not compared to reference distributions. In fact, no reliable experimental data can be obtained at very high Reynolds numbers for the turbulent kinetic energy, fluctuating vorticity and mean vortex wavelength because modern experimental systems cannot currently measure the very high velocity frequencies.

This chapter presents the root-mean-square fluctuating vorticity and the mean-vortex-wavelength distributions obtained from the *k-λ* model for fully rough flow over a range of relative roughness. The model is first compared to well established relations for the velocity distribution and resulting friction factor. Then the second turbulence variable distribution is presented. The *k-λ* model is compared to the Wilcox 1998 and 2006 models at high Reynolds numbers.

II. Model Validation

In order to assess the model's ability to predict fully rough flows, the current turbulence model is compared to well established relations for the velocity distribution and friction factor. The velocity distribution is compared to the log law and experimental data taken from Nikuradse's rough flow

study [46]. The friction factor is compared to the Colebrook equation from which the Moody diagram is based on.

The friction factor obtained from the current turbulence model is shown in Fig. 4.1 along with published experimental results. The first solid dot of each roughness ratio line corresponds to a roughness Reynolds number of 100. The flow is considered fully rough for a roughness Reynolds number greater than 1000, which is the fourth solid dots on each of the roughness ratio lines. The model shows excellent agreement with the fully rough limit of the Colebrook equation.

The model extrapolates well for rough flows having a roughness Reynolds number smaller than 1000. The friction factor is well within the scatter of the experimental data for roughness Reynolds number greater than 1000.

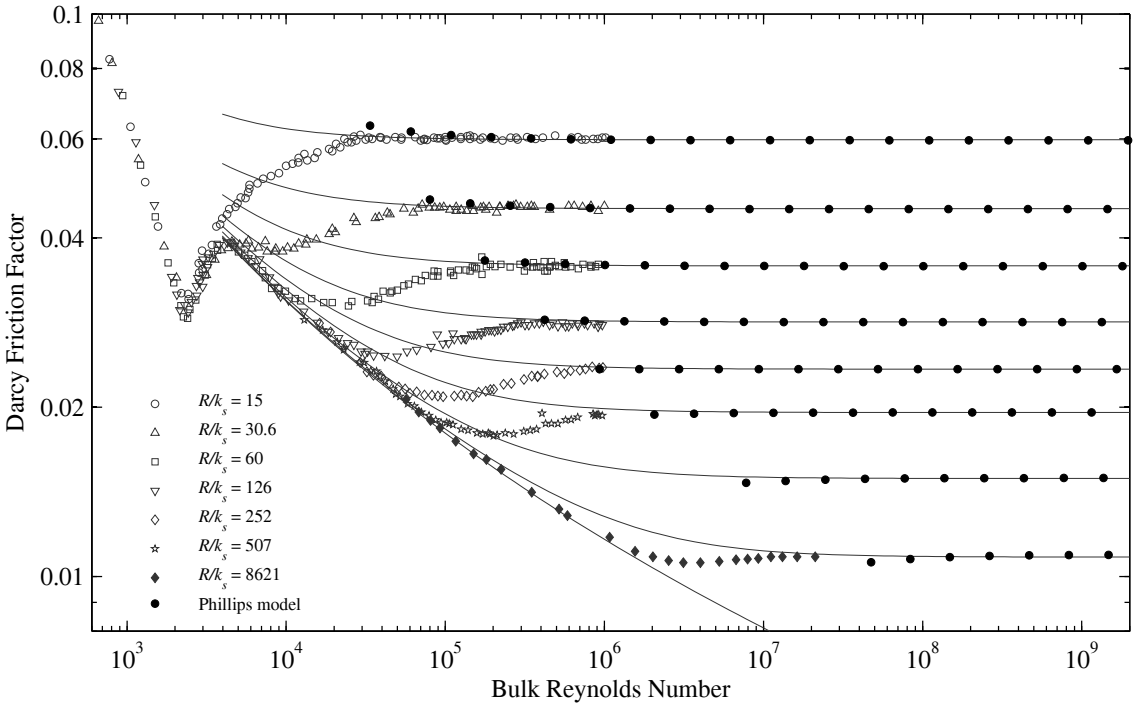

Fig. 4.1 Phillips *k-λ* turbulence model for σ_k = 4.0 and $k^+_{\mathbf{wall}}$= 0.1.

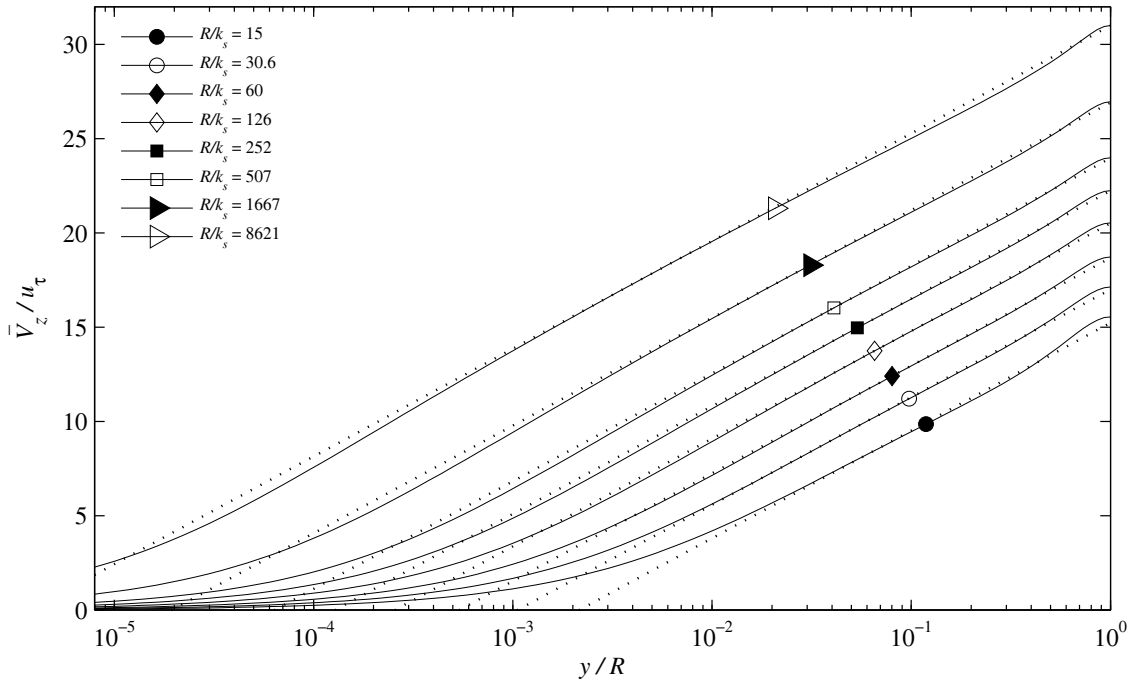

Fig. 4.2 Velocity distribution using σ_k = 4.0 and k^+_{wall}= 0.1.

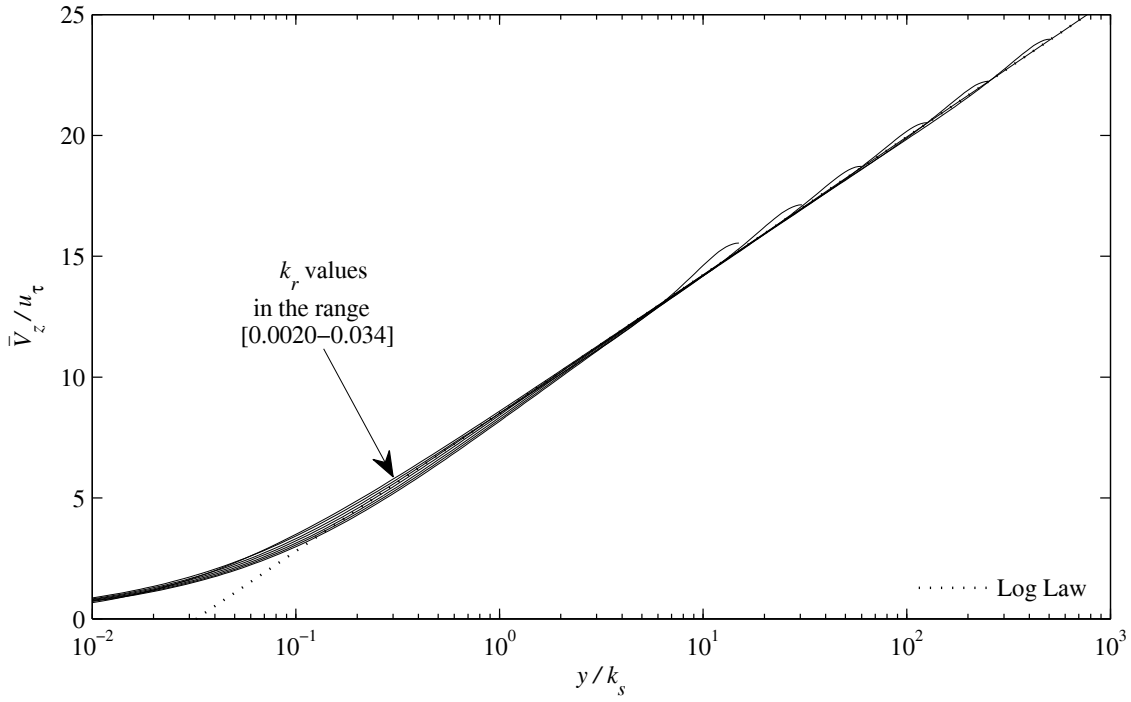

Fig. 4.3 Velocity profile in rough pipes using σ_k = 4.0 and k^+_{wall}= 0.1.

The velocity distribution obtained from the present k-λ turbulence model is compared to the log law for fully rough flow in Fig. 4.2. The log law is supported by Nikuradse's rough flow experiments of 1933. Figure 4.2 was generated at a roughness Reynolds number of 80,000. The agreement with the log law is excellent. Note that close to the centerline the centerline symmetry boundary condition imposes a deviation from the log law. Figure 4.3 shows a different representation of the velocity distribution. It can be seen that for any relative roughness number, all curves collapse to the log law curve. From Figs 4.2 and 4.3, the turbulence model predicts the velocity distribution for fully rough pipe flow to a high degree of accuracy. Both figures were generated using σ_k = 4.0 and k^+_{wall} = 0.1. The model gives similar results to those presented in Figs. 4.1–4.3 for values of σ_k in the range of 2.0 to 6.0 and values of k^+_{wall} in the range of 0.05 to 1.0.

III. The Second Turbulence Variable

The present turbulence model predicts the velocity distribution and friction factor accurately for fully rough pipe flows. The model is currently based on a transport equation for the turbulent kinetic energy and an algebraic relation for the mean vortex wavelength. Eventually, the model will be based on a transport equation for the root-mean-square fluctuating vorticity as a replacement of the empirical relation for the mean vortex wavelength. The square of the mean vortex wavelength is proportional to the turbulent kinetic energy over the square of the root-mean-square fluctuating vorticity.

$$\lambda^2 = C_\lambda \frac{k}{\tilde{\omega}^2} \tag{4.1}$$

This fully rough pipe flow model will serve as a reference for the turbulent kinetic energy, turbulent eddy viscosity and root-mean-square fluctuating vorticity distributions because no experimental data or empirical relations are available for these three variables. Using the following nondimensional parameters:

$$k^+ \equiv \frac{k}{u_\tau^2}, \qquad \hat{\lambda} \equiv \frac{\lambda}{R}, \qquad \omega^+ \equiv \frac{\omega R}{u_\tau} \tag{4.2}$$

Eq. (4.1) can be written in nondimensional form as

$$\hat{\lambda}^2 = C_\nu{}^2 \frac{k^+}{\omega^{+2}} \tag{4.3}$$

Figures. 4.4–4.7 show the turbulent kinetic energy, the mean vortex wavelength, the turbulent eddy viscosity and the root-mean-square fluctuating vorticity for fully rough flow. These four figures were generated using a roughness Reynolds number of 80,000, σ_k = 4.0 and k^+_{wall} = 0.1. Three inverse roughness ratios are given: 15, 126 and 507. The variations of the turbulent kinetic energy and the second turbulence variable corresponding to the roughness ratio, results from a variation of the velocity distribution with respect to the roughness ratio, as shown in Fig. 4.8. As the inverse roughness ratio is decreased, the friction factor decreases as well. The friction factor is inversely proportional to the mean velocity. At low inverse roughness ratio (e.g. 15), the mean velocity is smaller than for higher inverse roughness ratio (e.g. 252). Therefore the friction factor is greater at high inverse roughness ratio (e.g. 15). Figure 4.8 shows that at high inverse roughness ratio, the mean velocity is greater than for low inverse roughness ratio.

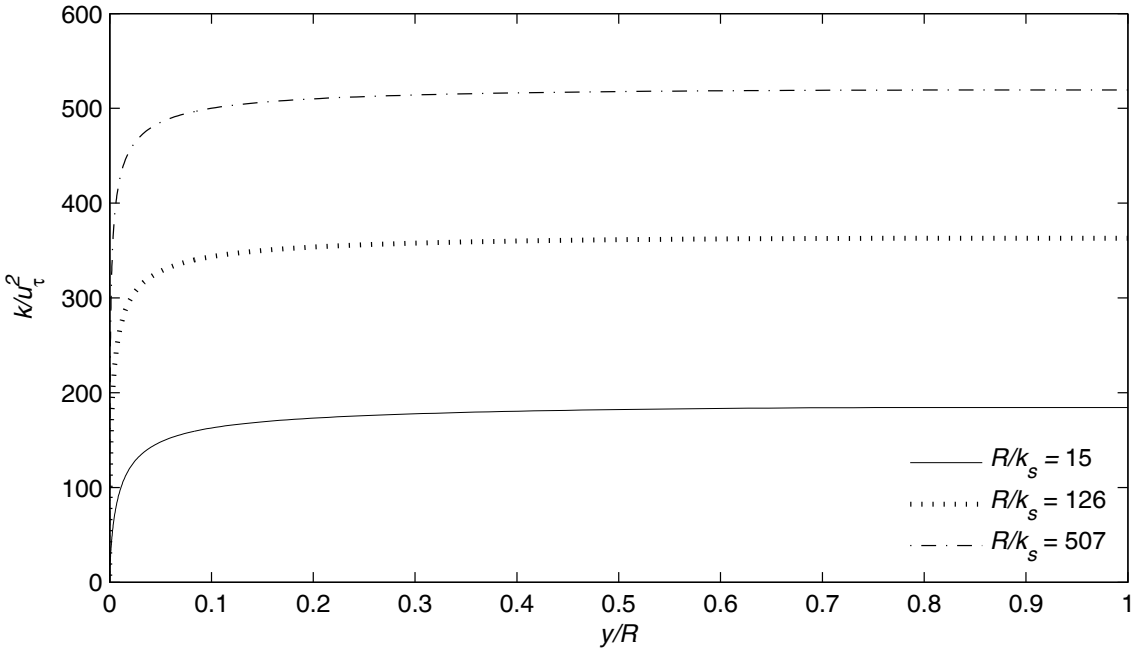

Fig. 4.4 Turbulent-kinetic-energy distribution for different inverse roughness ratios σ_k = 4, k^+_{wall} = 0.1, k_s^+ = 80,000.

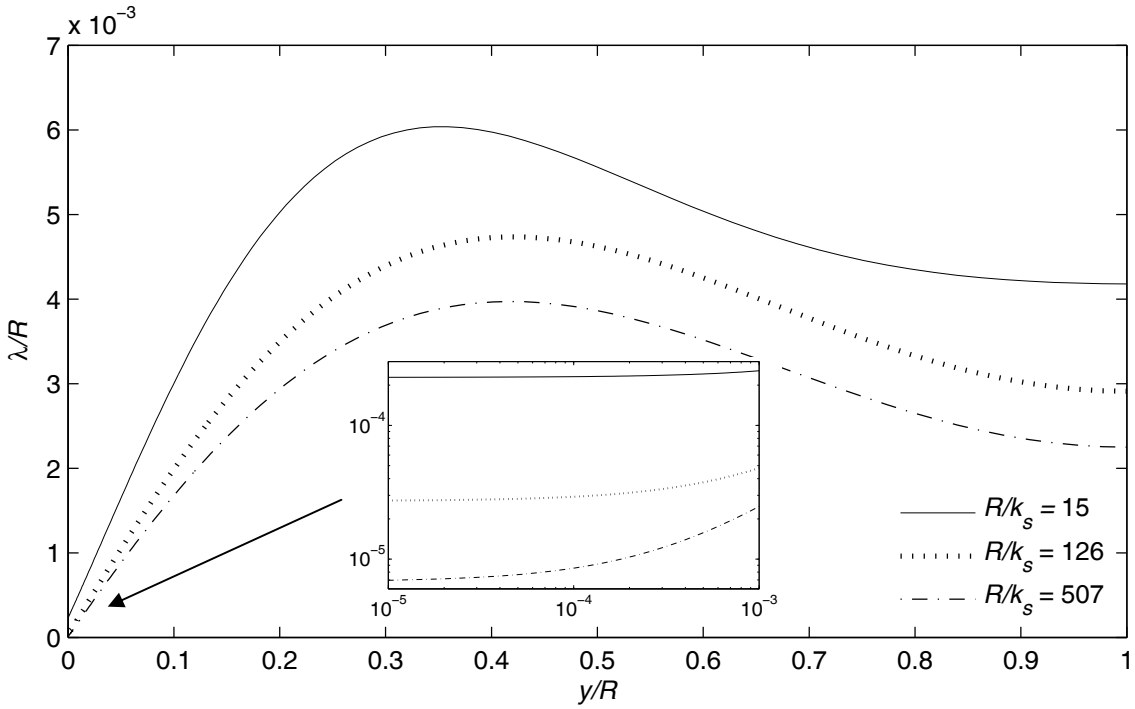

Fig. 4.5 Mean-vortex-wavelength distribution for different inverse roughness ratios, σ_k = 4, k^+_{wall} = 0.1, k_s^+ = 80,000.

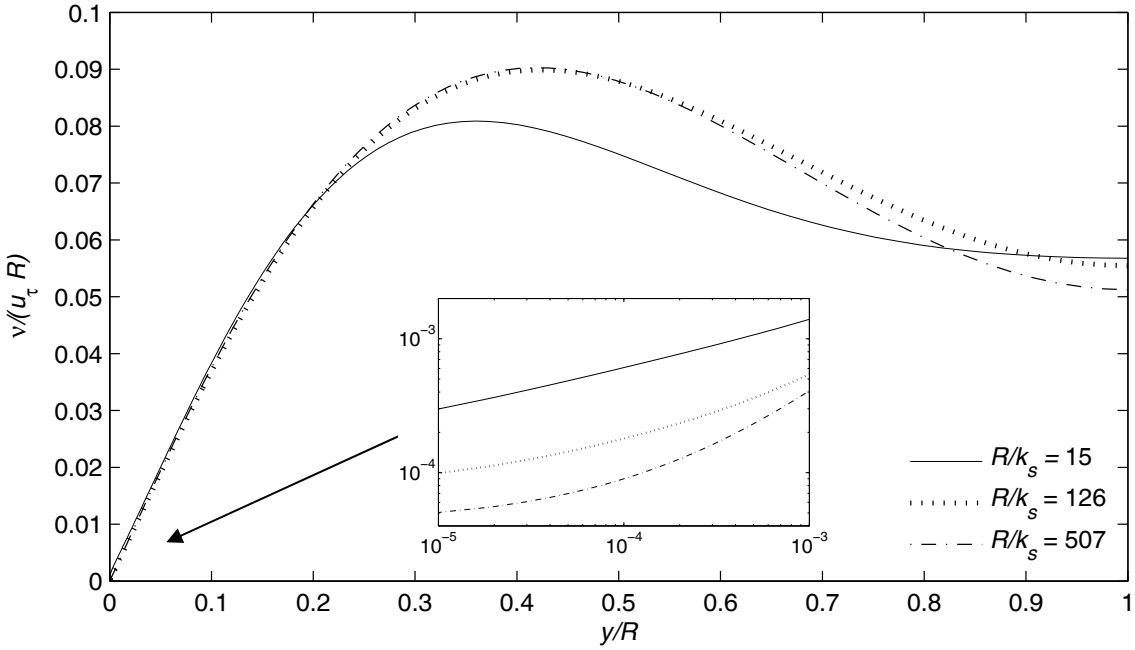

Fig. 4.6 Turbulent eddy viscosity distribution for different inverse roughness ratios, σ_k = 4, k^+_{wall}= 0.1, k_s^+ = 80,000.

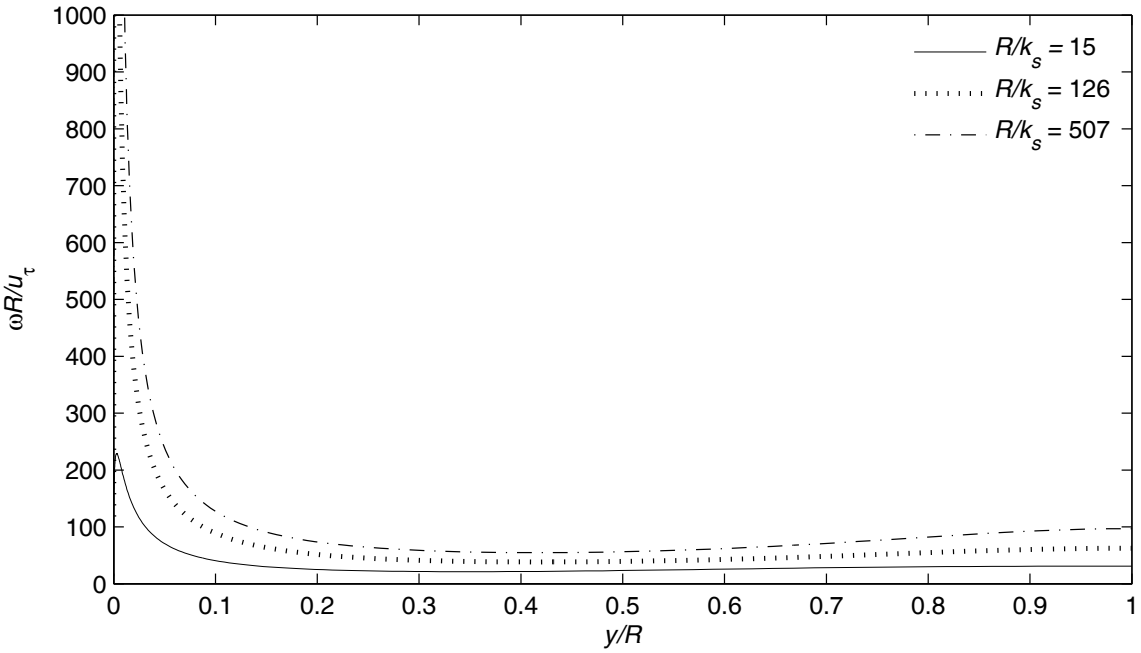

Fig. 4.7 Root-mean-square fluctuating vorticity distribution for different inverse roughness ratios, linear scale, $\sigma_k = 4$, $k^+_{wall} = 0.1$, $k_s^+ = 80{,}000$.

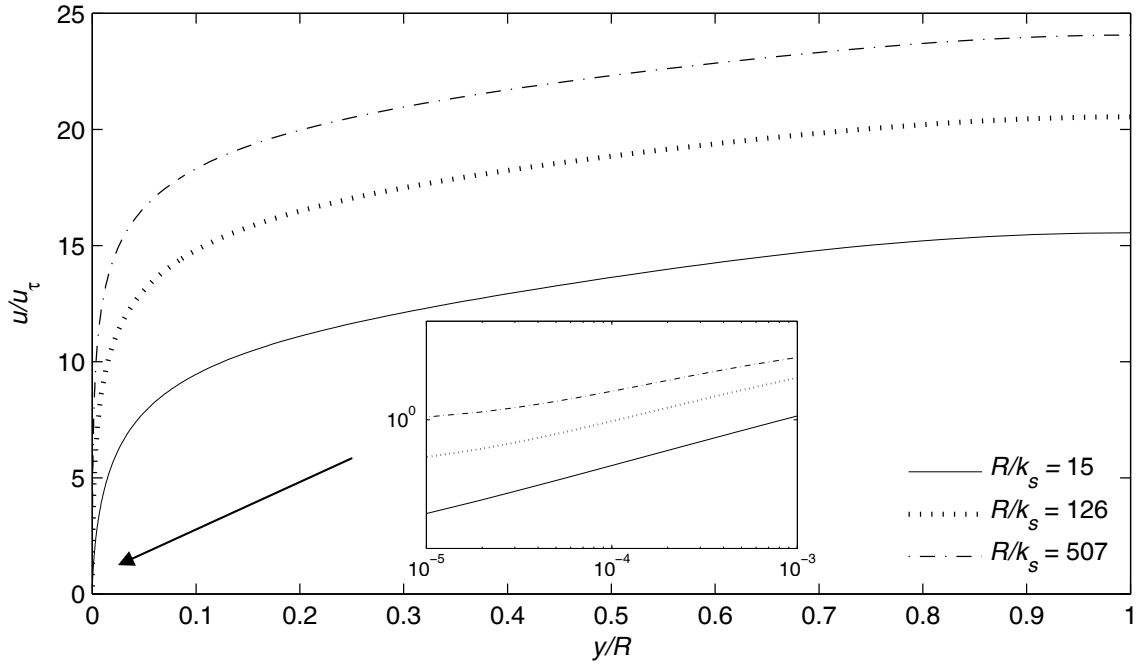

Fig. 4.8 Velocity distribution for different inverse roughness ratios, $\sigma_k = 4$, $k^+_{wall} = 0.1$, $k_s^+ = 80{,}000$.

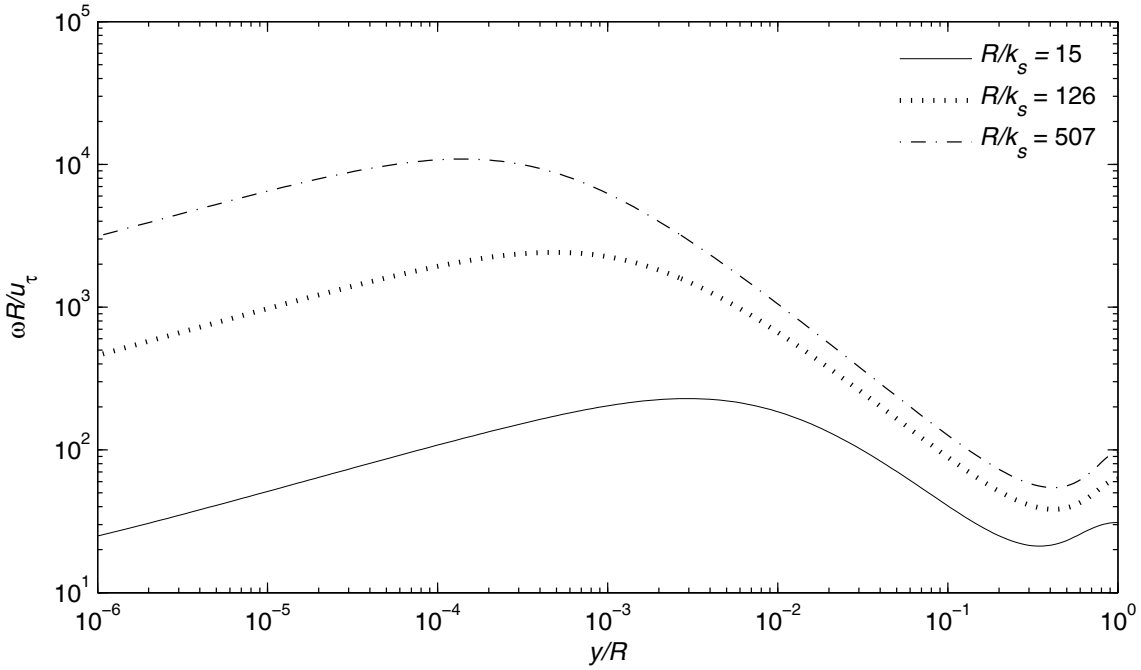

Fig. 4.9 Root-mean-square fluctuating vorticity distribution for different inverse roughness ratios, log scale, $\sigma_k = 4$, $k^+_{wall} = 0.1$, $k_s^+ = 80{,}000$.

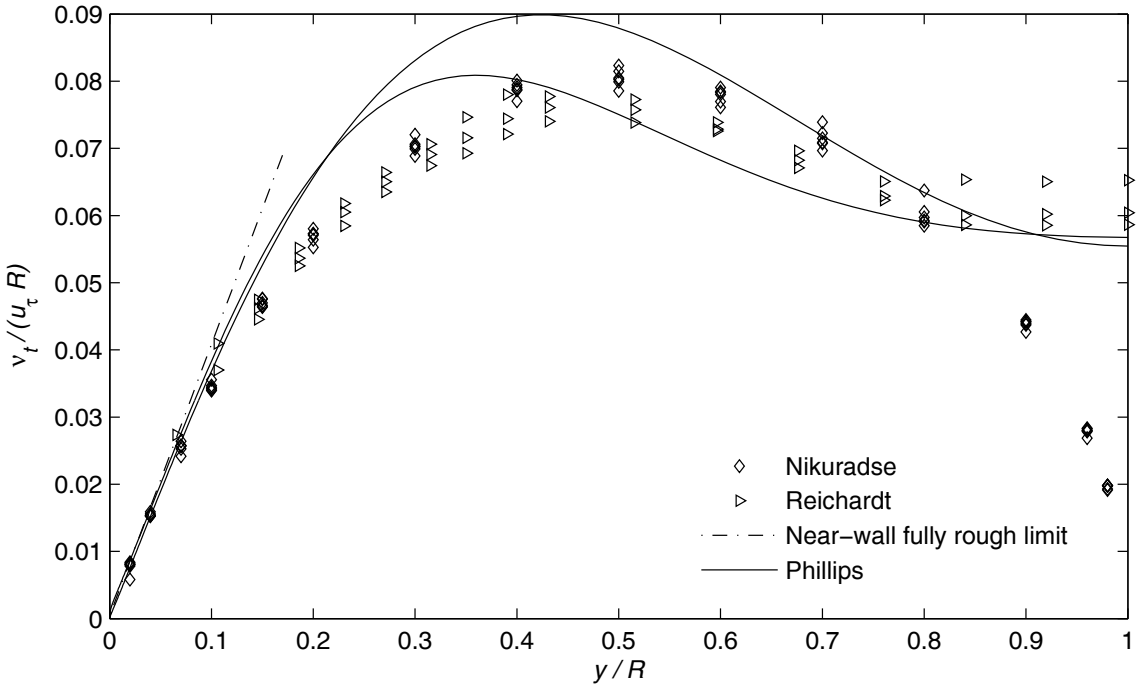

Fig. 4.10 Turbulent eddy viscosity distribution based on σ_k = 4, k^+_{wall}= 0.1, k_s^+ = 80,000 and R/k_s = [15-507] compared to experimental data.

From Fig. 4.6, the apparent value at the wall of the root-mean-square fluctuating vorticity appears to decrease as the inverse roughness ratio increases. As the pipe gets smoother, the apparent wall value of the root-mean-square fluctuating vorticity seems to originate at infinity. In fact, looking at the same data on a log-log scale proves that for any roughness number, the value of the vorticity at the wall is finite and increases slightly and decreases very sharply, as seen in Fig. 4.9. The value of the vorticity at the wall can be computed from the ratio of the turbulent kinetic energy at the wall and the square of the mean turbulent wavelength.

Figure 4.10 shows the turbulent eddy viscosity profile compared to values obtained by Nikuradse and Reichardt. The turbulent eddy viscosity obtained by Nikuradse and Reichardt is not expected to match that obtained from the current k-λ model. However, the order of magnitude should be close, which is the case. The data referring to Nikuradse and Reichardt is not obtained from experiments directly, but it is derived from the derivative of the experimental velocity values. Because taking the derivative of experimental points accumulates errors, the turbulent eddy viscosity obtained from Nikuradse and Reichardt is not expected to match that obtained from the current turbulence model. Note that the turbulent eddy viscosity stays fairly constant over the core of the pipe, as observed by Kays and Crawford [86] who suggested using a constant value over the central region of the pipe combined with an approximation from mixing-length theory near the wall.

The variations in the closure coefficient σ_k of the turbulent kinetic energy, the mean vortex wavelength, the turbulent eddy viscosity and the root-mean-square fluctuating vorticity for fully rough flow are shown in Figures 4.11–4.14. Because the turbulent-kinetic-energy transport equation depends on the closure coefficient σ_k, the distribution of the turbulent kinetic energy and the second turbulence variable also depend on this closure coefficient. The figures were generated using $k^+_{\text{wall}} = 0.1$, an inverse roughness ratio number of 252 and a roughness Reynolds number of 80000, which corresponds to fully rough flow. Changing the closure coefficient σ_k greatly impacts the amplitude of the turbulent kinetic energy, mean vortex wavelength and fluctuating vorticity. Increasing σ_k results in a decrease of the transport of the turbulent kinetic energy.

The variations in the turbulent kinetic energy, the mean vortex wavelength, the turbulent eddy viscosity and the root-mean-square fluctuating vorticity with the proportionality coefficient in the wall

boundary condition are shown in Figs. 4.15–4.19. Only slight variations with k^+_{wall} can be depicted on Figs. 4.15–4.19. Figure 4.15 shows that only the region very close to the wall is affected by a change in k^+_{wall}. Those figures were generated using 4 for the closure coefficient σ_k, an inverse roughness ratio of 252 and a roughness Reynolds number of 80,000. The proportionality coefficient in the wall boundary condition was only varied from 0.05 to 1.0.

The turbulent kinetic energy, mean vortex wavelength, turbulent eddy viscosity and root-mean-square fluctuating vorticity variations with the roughness Reynolds number are shown in Figs. 4.20–4.23. These figures were generated using k^+_{wall} = 0.1, an inverse roughness ratio of 252 and the closure coefficient σ_k set to 4.

There are only slight changes in the amplitude of the turbulent variable distributions between a roughness Reynolds number of 1000 and 80,000. Both of the roughness Reynolds numbers can be considered to represent fully rough flow. Changes are more noticeable for rough flow having a roughness Reynolds number of 100. At this roughness Reynolds number, the ratio of the turbulent eddy viscosity to the molecular viscosity is slightly greater than 1.

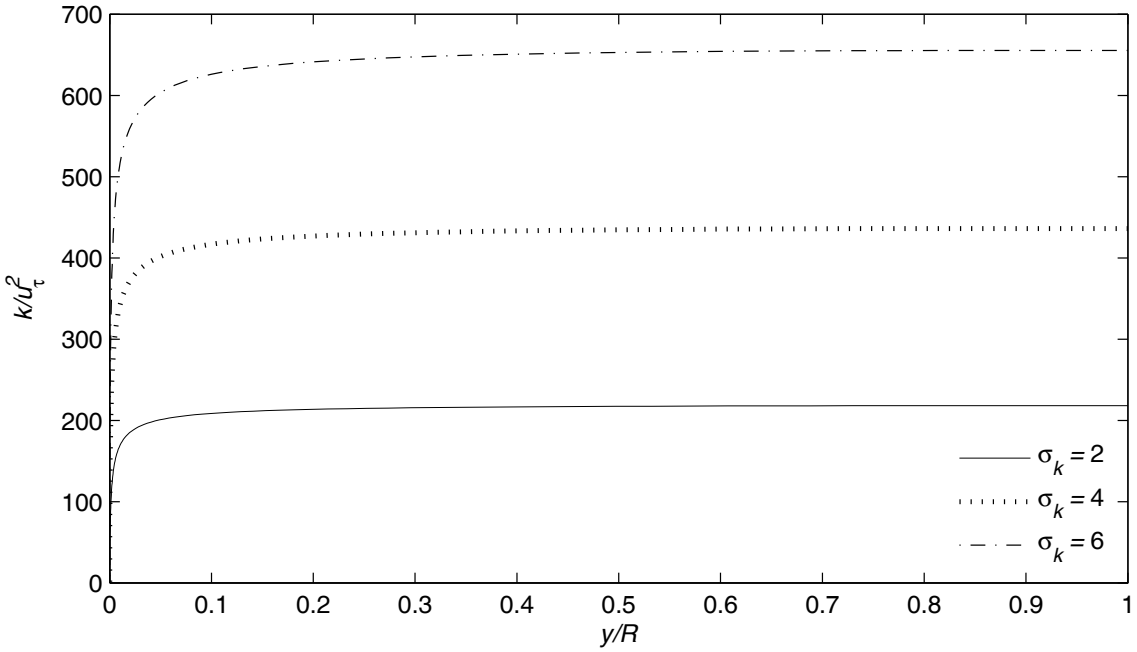

Fig. 4.11 Turbulent-kinetic-energy distribution for different σ_k values, k^+_{wall} = 0.1, k_s^+ = 80,000, R/k_s = 252.

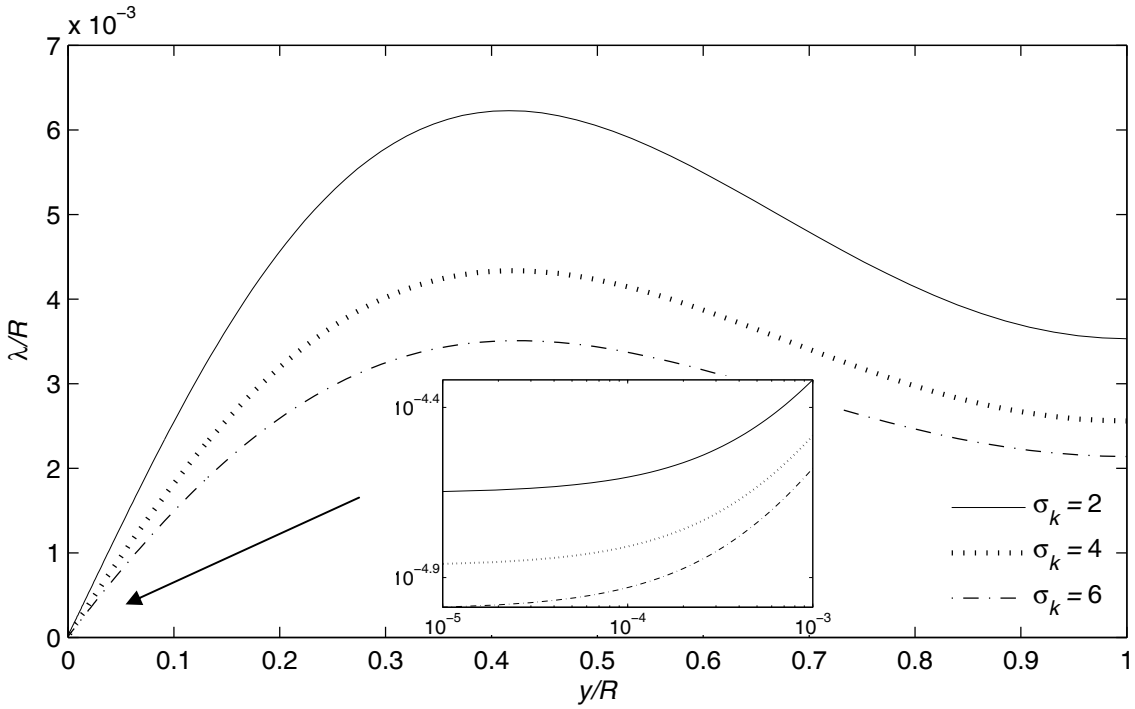

Fig. 4.12 Mean-vortex-wavelength distribution for different σ_k values, k^+_{wall}= 0.1, k_s^+ = 80,000, R/k_s = 252.

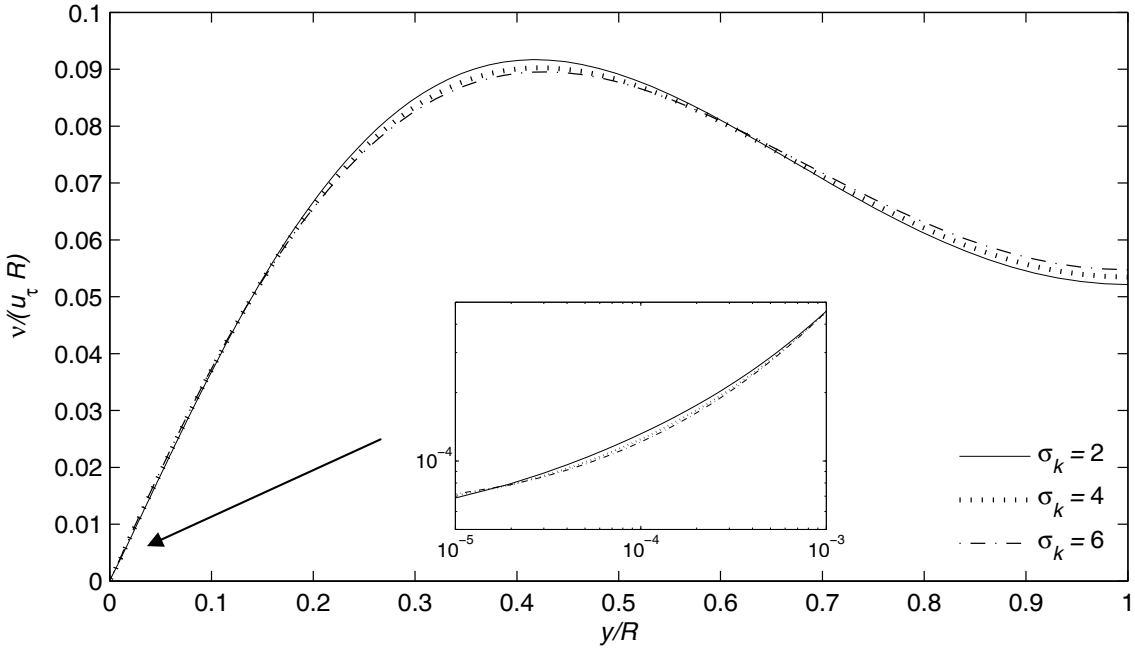

Fig. 4.13 Turbulent eddy viscosity distribution for different σ_k values, k^+_{wall}= 0.1, k_s^+ = 80,000, R/k_s = 252.

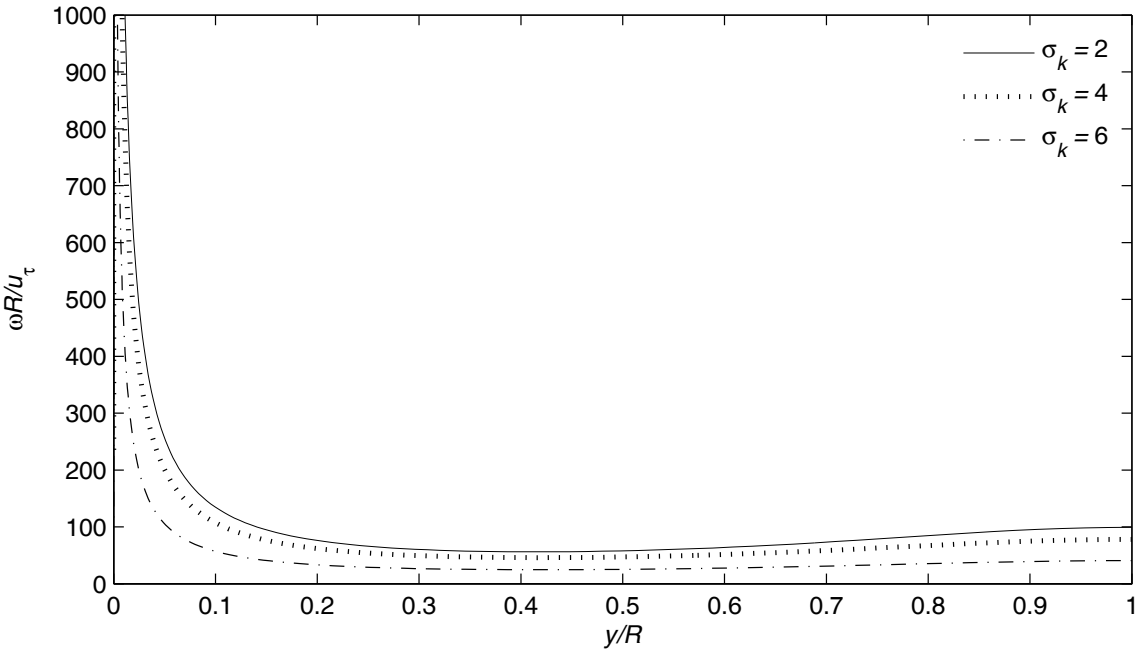

Fig. 4.14 Root-mean-square fluctuating vorticity distribution for different σ_k values, k^+_{wall} = 0.1, k_s^+ = 80,000, R/k_s = 252.

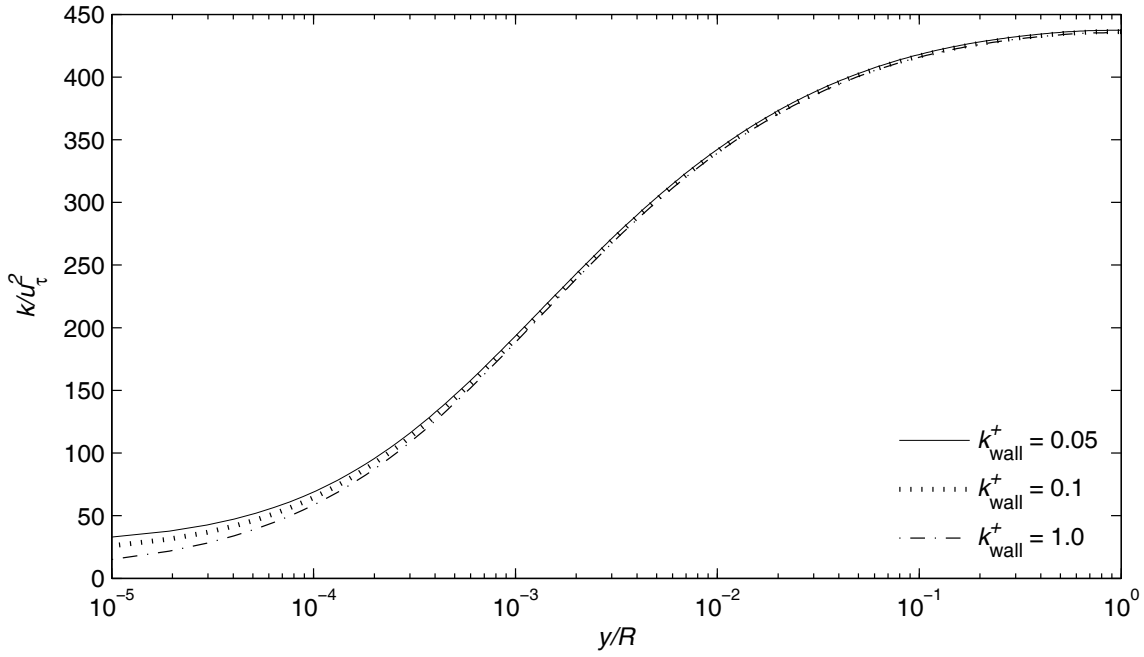

Fig. 4.15 Turbulent-kinetic-energy distribution for different k^+_{wall} values, logarithmic scale, σ_k = 4.0, k_s^+ = 80,000, R/k_s = 252.

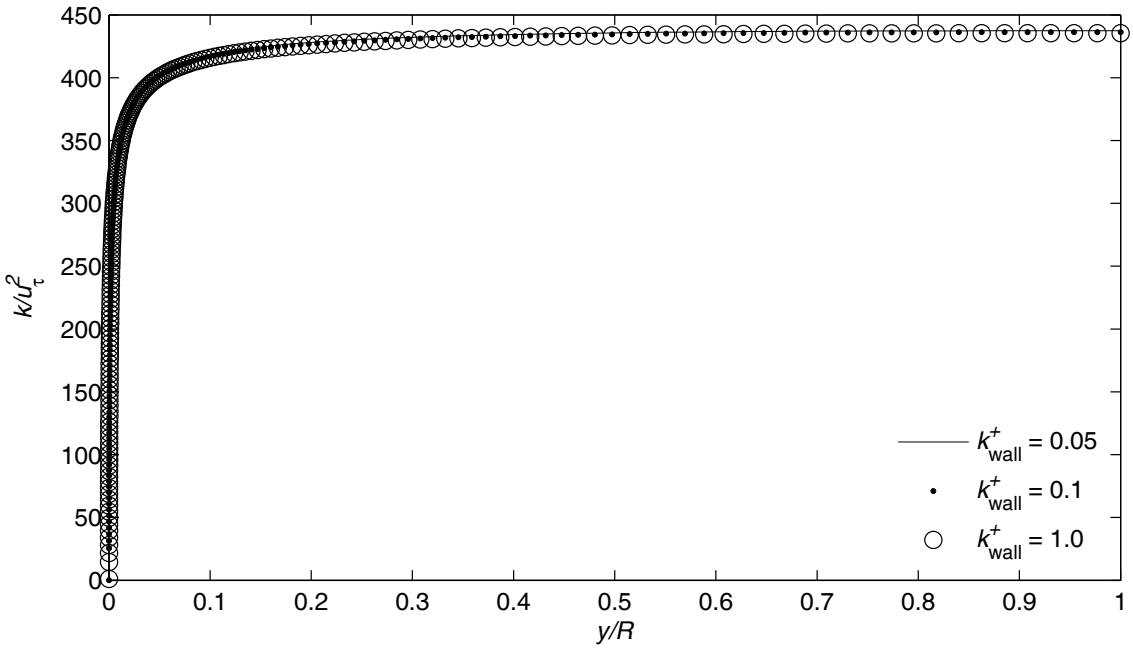

Fig. 4.16 Turbulent-kinetic-energy distribution for different k^+_{wall} values, linear scale, σ_k = 4.0, k_s^+ = 80,000, R/k_s = 252.

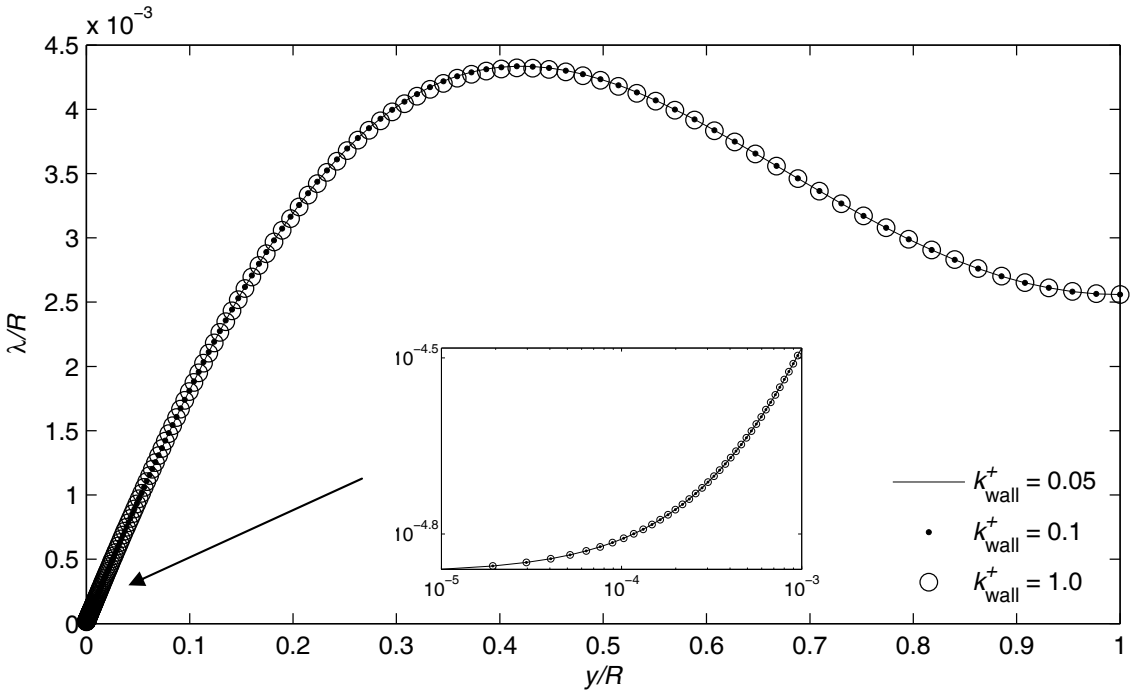

Fig. 4.17 Mean-vortex-wavelength distribution for different k^+_{wall} values, σ_k = 4.0, k_s^+ = 80,000, R/k_s = 252.

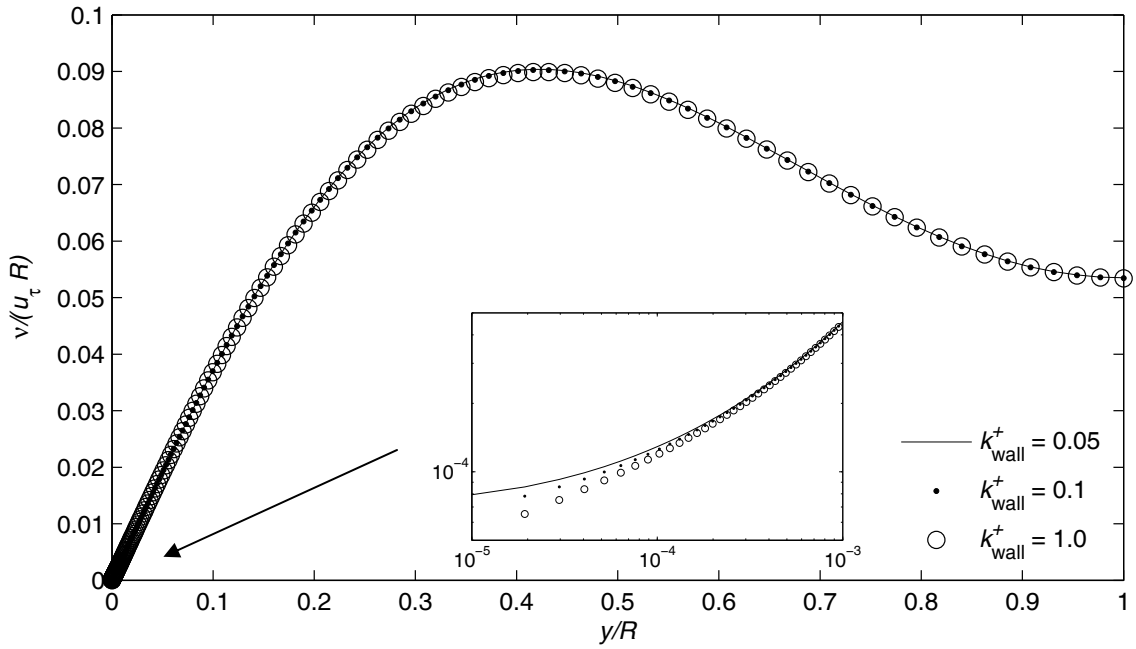

Fig. 4.18 Turbulent eddy viscosity distribution for different k^{+}_{wall} values, σ_k = 4.0, k_s^{+} = 80,000, R/k_s = 252.

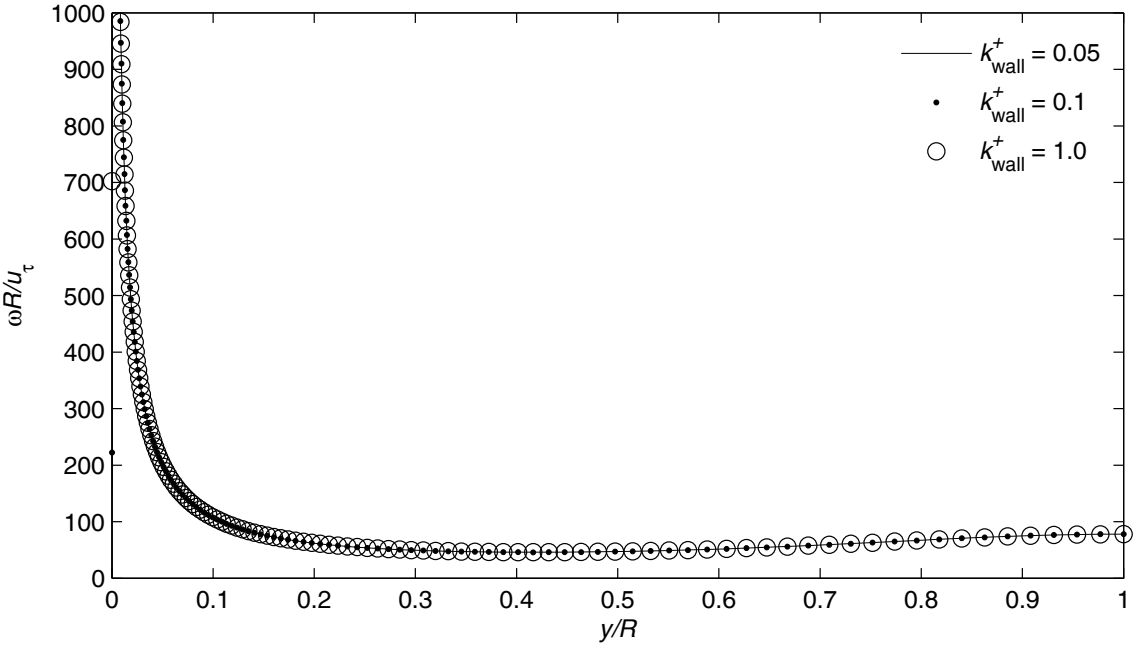

Fig. 4.19 Root-mean-square fluctuating vorticity distribution for different k^{+}_{wall} values, σ_k = 4.0, k_s^{+} = 80,000, R/k_s = 252.

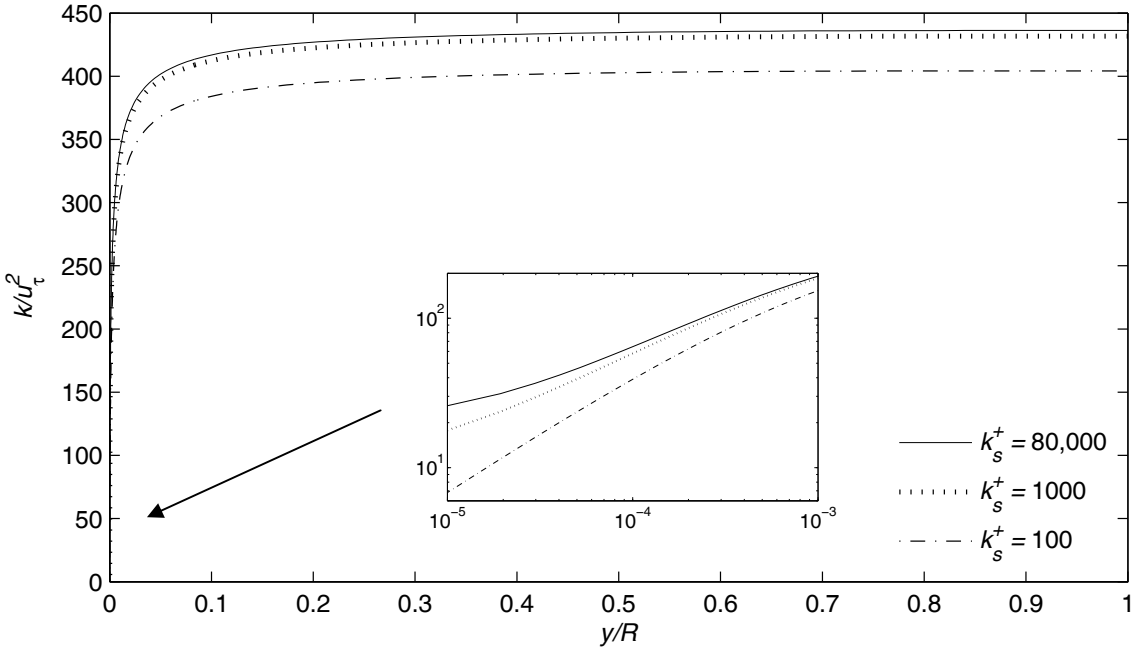

Fig. 4.20 Turbulent-kinetic-energy distribution for different k_s^+ values, σ_k = 4.0, k_{wall}^+= 0.1, R/k_s = 252.

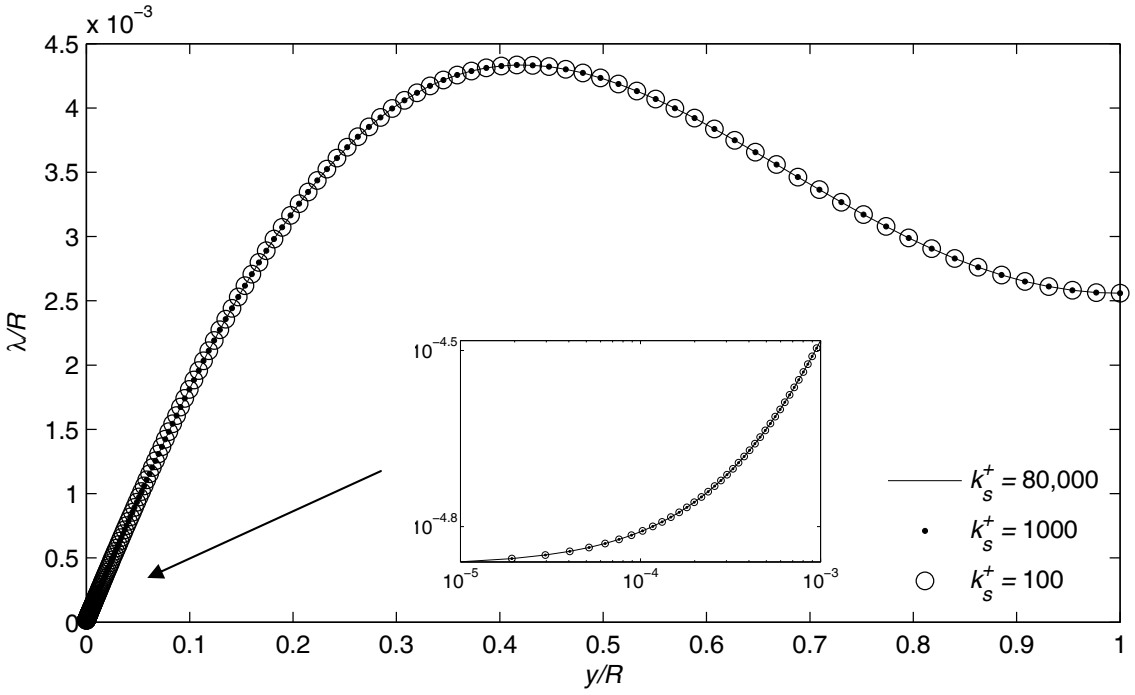

Fig. 4.21 Mean-vortex-wavelength distribution for different k_s^+ values, σ_k = 4.0, k_{wall}^+ = 0.1, R/k_s = 252.

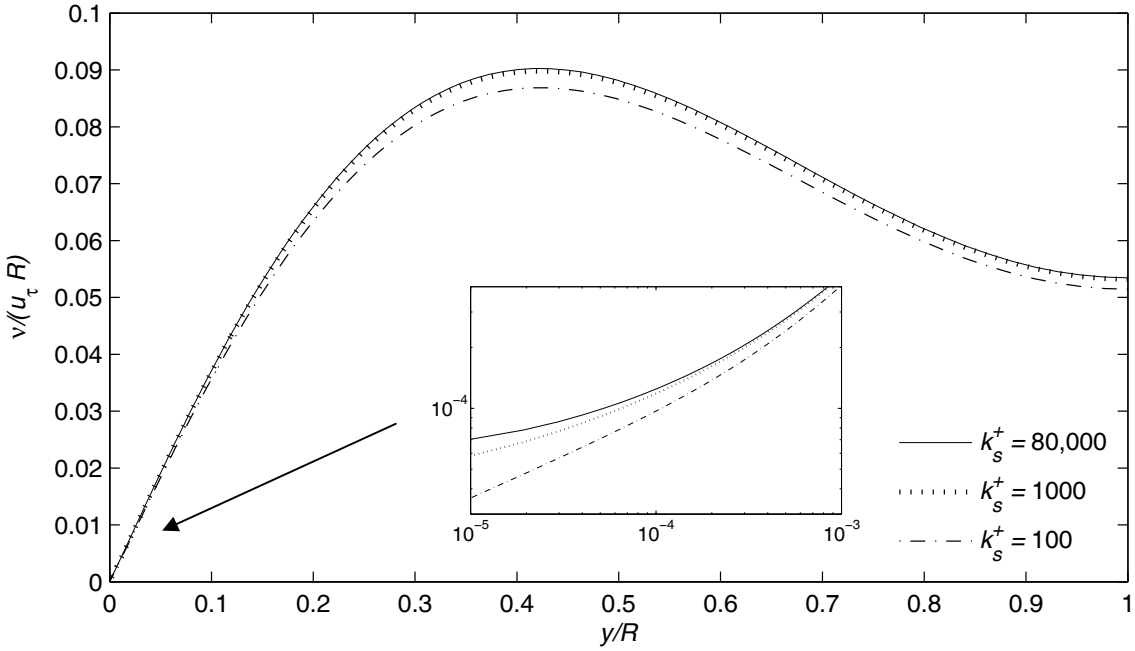

Fig. 4.22 Turbulent eddy viscosity distribution for different k_s^+ values, σ_k = 4.0, k^+_{wall}= 0.1, R/k_s = 252.

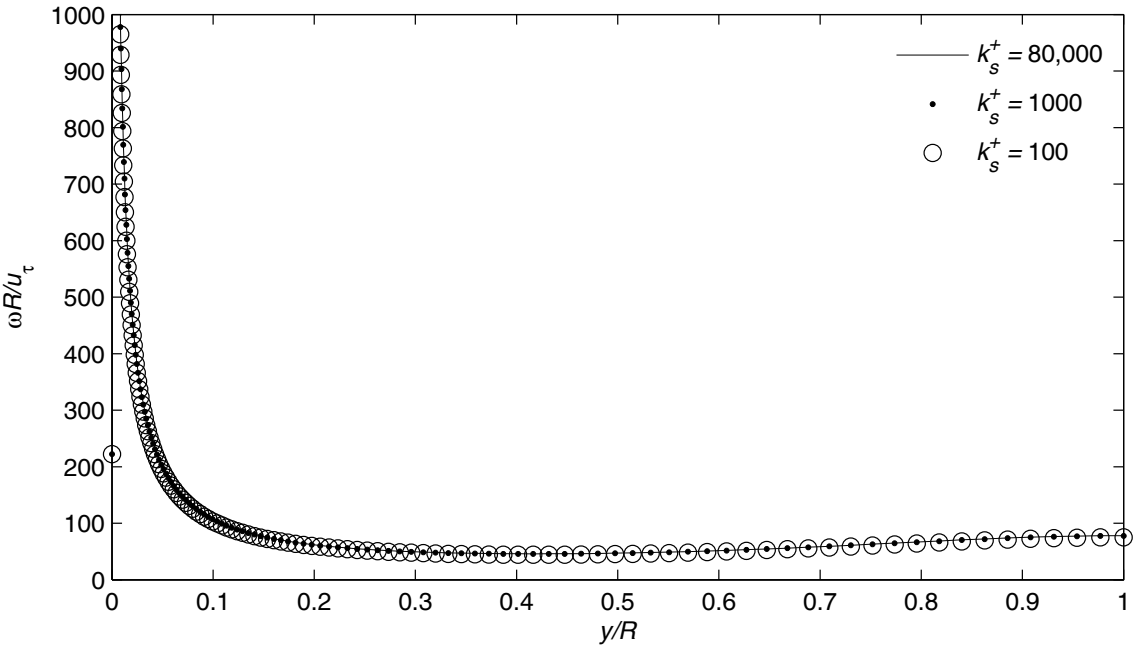

Fig. 4.23 Root-mean-square fluctuating vorticity distribution for different k_s^+ values, σ_k = 4.0, k^+_{wall} = 0.1, R/k_s = 252.

The flow is said to be fully rough when the molecular viscosity is negligible with regard to the turbulent eddy viscosity. Therefore changes in the distribution of the turbulent variables are expected to be between a roughness Reynolds number of 100 and one representing fully rough flow deeming a roughness Reynolds number of 1000 or more.

IV. Comparison to Other Models

A. Wilcox *k-ω* Model

Traditionally the *k-ω* model has been accepted as the model most capable of modeling rough-wall effects without implementing wall functions. The *k-ω* model was first attempted by Kolmogorov [8], and later revised by Saffman [87], Wilcox [88], Peng et al. [89], Kok [90] and Hellsten [91]. Wilcox 98 model [13] is the turbulent model implemented in Fluent [92]. Wilcox made a few modifications to his 98 model and proposed an improved model in 2006. This section compares the current turbulence model to both the Wilcox 98 and 2006 models for fully rough flow.

1. Wilcox 1998 k-ω Model

The Wilcox 1998 *k-ω* model [13] is likely the most widely used model for rough walls. The effects of surface roughness are commonly incorporated into the *k-ω* model by altering the surface boundary condition on ω. The Wilcox 1998 *k-ω* model is comprised of the following equations for incompressible flow. The algebraic equation for the eddy viscosity is

$$\nu_t = f_\mu \frac{k}{\omega} \tag{4.4}$$

The turbulent energy transport equation is

$$\frac{\partial k}{\partial t} + \overline{\mathbf{V}} \cdot \nabla k = 2\nu_t \bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) : \bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) - C_\mu f_k k\,\omega + \nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \nabla k \right] \tag{4.5}$$

The dissipation frequency transport equation is

$$\frac{\partial \omega}{\partial t} + \overline{\mathbf{V}} \cdot \nabla \omega = 2C_{\omega 1} f_1 \nu_t \frac{\omega}{k} \bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) : \bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) - C_{\omega 2} f_2 \, \omega^2 + \nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_\omega} \right) \nabla \omega \right] \tag{4.6}$$

The functions f_1, f_μ, f_k, f_2 are wall damping functions, also called low Reynolds number corrections, employed to force the model to agree more closely with near-wall experimental data. The near-wall damping functions used in Wilcox 1998 k-ω model are

$$R_t \equiv \frac{k}{\nu \omega}, \quad \chi_k \equiv \frac{\nabla k \cdot \nabla \omega}{\omega^3}, \quad f_\mu = \frac{0.024 + R_t / 6}{1 + R_t / 6},$$

$$f_k = \frac{4/15 + (R_t/8)^4}{1 + (R_t/8)^4} \begin{cases} 1, & \chi_k \le 0 \\ \dfrac{1 + 680\chi_k^2}{1 + 400\chi_k^2}, & \chi_k > 0 \end{cases}, \quad f_1 = \frac{1/9 + R_t/2.95}{f_\mu (1 + R_t/2.95)}, \quad f_2 = 1 \tag{4.7}$$

$$C_\mu = 0.09, \quad C_{\omega 1} = 0.52, \quad C_{\omega 2} = 0.072, \quad \sigma_k = 2.0, \quad \sigma_\omega = 2.0$$

Wilcox suggests using the following relation for ω^+ at a rough wall

$$\omega^+\Big|_{y^+=0} = \begin{cases} \left(\dfrac{200}{k_s^+} \right)^2, & k_s^+ \le 5 \\ \dfrac{100}{k_s^+} + \left[\left(\dfrac{200}{k_s^+} \right)^2 - \dfrac{100}{k_s^+} \right] \exp\left(5 - k_s^+\right), & k_s^+ > 5 \end{cases} \tag{4.8}$$

Along with his book on turbulence modeling [13], Wilcox incorporates a CD with a number of computer programs capable of running his model for various flows. One of the programs is a fully developed pipe flow code, which allows the user to model roughness effects by specifying the value of ω^+ at the wall. Multiple cases were run to cover the range of roughness ratios commonly displayed in the Moody diagram. The code requires the shear Reynolds number R_τ as input, which is a function of the roughness ratio

$$R_\tau = k_s^+ \frac{R}{k_s} \quad \text{or} \quad R_\tau = \frac{k_s^+}{2k_r} \tag{4.9}$$

The fully developed pipe flow code was used to compare the Wilcox 1998 model to the Colebrook equation and Nikuradse's experimental cases. For each R/k_s value, the code was run from $k_s^+ = 10^2$ to $k_s^+ = 10^6$ or until the code no longer converged.

Figure 4.24 shows the friction factor obtained from Wilcox 1998 k-ω model. For fully rough flow, at high Reynolds numbers, the friction factor is within the scatter of experimental data. However, as can be seen there is a positive slope for all roughness ratios at high Reynolds numbers. This figure is to be compared with Fig. 4.1, which displays the same diagram using data from the current one-equation k-λ model. Clearly, the Wilcox 1998 model does not match the Colebrook equation as well as the k-λ turbulence model does.

Figure 4.25 shows the velocity distribution obtained from the Wilcox 1998 k-ω model. It can be seen that the velocity distributions are systematical and slightly below the log law curve. Figure 4.25 shows that the bulk velocity will be under predicted, which corresponds to an over predicted friction factor at high roughness Reynolds numbers, as shown in Fig. 4.24. Figure 4.25 is to be correlated with Fig. 4.2 which displays the velocity profiles obtained from the current k-λ model. The current k-λ model predicts the velocity to a much higher accuracy and extrapolates better for lower roughness ratios.

2. Wilcox 2006 k-ω Model

In 2006, Wilcox published a revised version [13] of his 1998 model, which has not yet been as widely implemented as his 1998 model. The two-equation model is given by the following three equations. The algebraic kinematic eddy viscosity function is computed from

$$\nu_t = \frac{k}{\hat{\omega}} \quad , \qquad \hat{\omega} = \max\left(\omega, \frac{7}{8}\sqrt{\frac{2\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}) : \bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})}{C_\mu}} \right) \tag{4.10}$$

The turbulent energy transport equation is

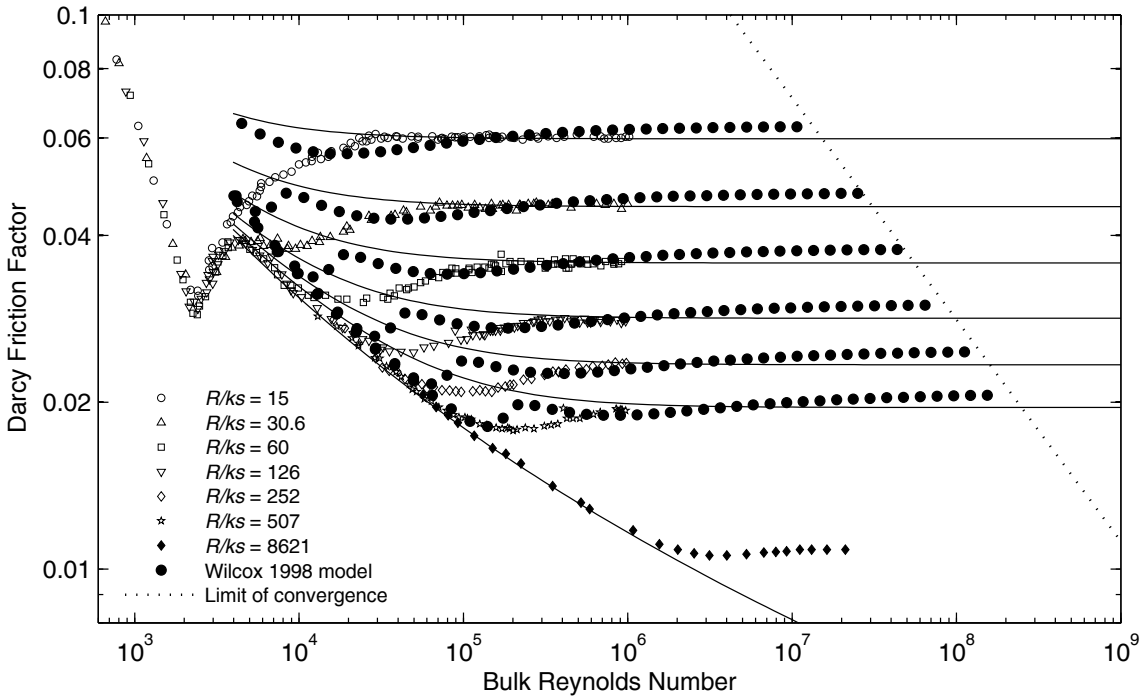

Fig. 4.24 Darcy friction factors obtained from the Wilcox 1998 *k-ω* model.

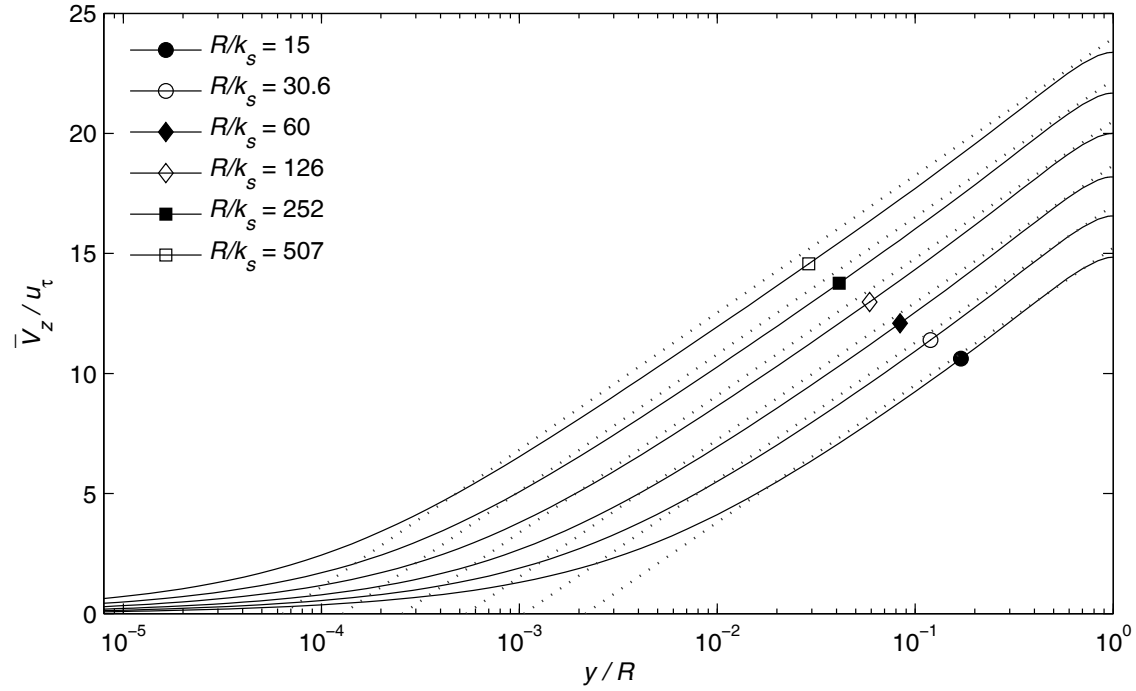

Fig. 4.25 Velocity profile obtained from the Wilcox 1998 *k-ω* model.

$$\frac{\partial k}{\partial t}+\overline{\mathbf{V}}\cdot\nabla k=2\nu_t\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}):\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})-C_\mu k\,\omega+\nabla\cdot\left[\left(\nu+\frac{1}{\sigma_k}\frac{k}{\omega}\right)\nabla k\right] \tag{4.11}$$

The dissipation frequency transport equation is

$$\frac{\partial \omega}{\partial t}+\overline{\mathbf{V}}\cdot\nabla\omega=2C_{\omega 1}\frac{\omega}{k}\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}}):\bar{\bar{\mathbf{S}}}(\overline{\mathbf{V}})+f_1\frac{\nabla k\cdot\nabla\omega}{\omega}-C_{\omega 2}f_2\,\omega^2+\nabla\cdot\left[\left(\nu+\frac{1}{\sigma_\omega}\frac{k}{\omega}\right)\nabla\omega\right] \tag{4.12}$$

The functions f_1 and f_k are wall damping functions defined as

$$R_t\equiv\frac{k}{\nu\omega},\quad \chi_k\equiv\frac{\nabla k\cdot\nabla\omega}{\omega^3},$$

$$f_1=\begin{cases}0, & \nabla k\cdot\nabla\omega\le 0\\ \dfrac{1}{8}, & \nabla k\cdot\nabla\omega>0\end{cases},\quad f_2=1, \tag{4.13}$$

$$C_\mu=0.09,\quad C_{\omega 1}=0.52,\quad C_{\omega 2}=0.0708,\quad \sigma_k=\frac{5}{3},\quad \sigma_\omega=2.0$$

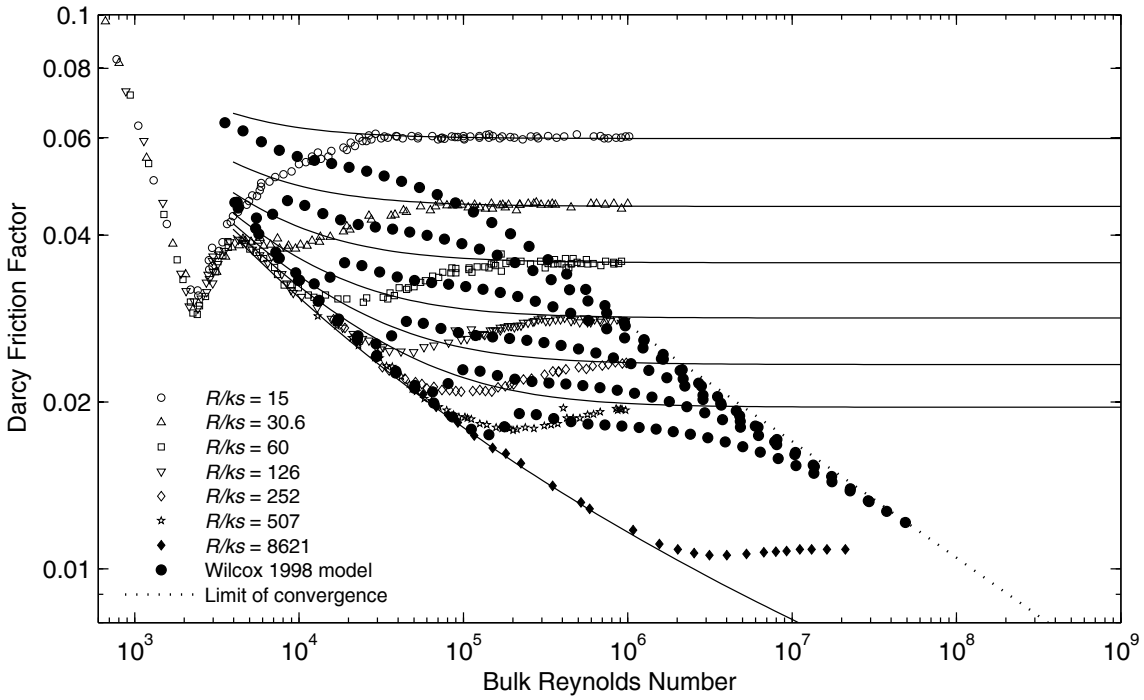

Fig. 4.26 Darcy friction factors obtained from the Wilcox 2006 *k-ω* model.

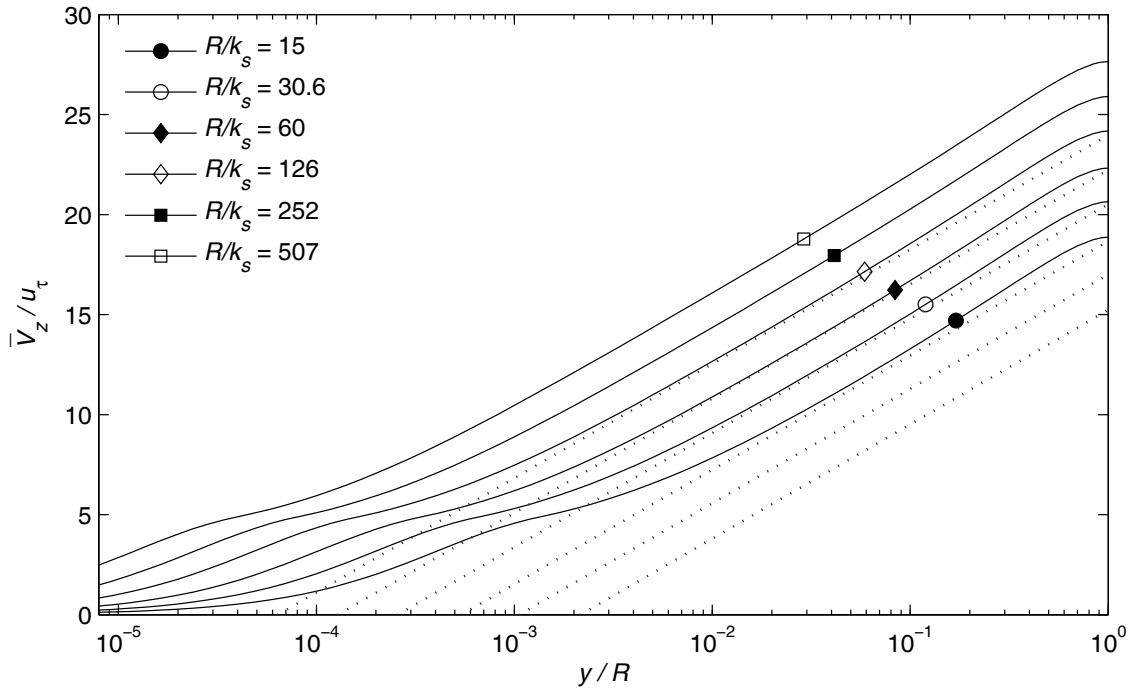

Fig. 4.27 Velocity profile obtained from the Wilcox 2006 *k-ω* model.

Figure 4.26 shows the friction factor obtained from his 2006 code compared to the Colebrook equation and the Nikuradse data. The code does not converge for k_s^+ values higher than about 1000 depending on the roughness ratio. Clearly, the model cannot handle fully rough flow. Figure 4.27 shows the velocity profile obtained from the Wilcox 2006 model. The velocity profiles do not follow the law of the wall. The bulk velocity is always over-predicted and corresponds to an under-predicted friction factor as was shown in Fig. 4.26.

A mean dissipation characteristic length can be defined from the turbulent eddy viscosity equation. The turbulent eddy viscosity is proportional to the product of a characteristic length times a characteristic velocity. The characteristic velocity associated with the vorticity-based turbulence model could be the square root of the turbulent kinetic energy. In the traditional *k-ϵ* model, the turbulent eddy viscosity is given by

$$\begin{aligned} \nu_t &= C_\mu \frac{k^2}{\varepsilon} \\ &= \left(C_\mu \frac{k^{3/2}}{\varepsilon} \right) k^{1/2} \end{aligned} \tag{4.14}$$

The characteristic length associated with the dissipation is the term in parathensis given in Eq. (4.14). Using Eq. (1.90), this characteristic length can be rewritten in terms of the dissipation frequency ω,

$$\nu_t = \left(\frac{k^{1/2}}{\omega} \right) k^{1/2} \tag{4.15}$$

This suggests that we could define a "mean dissipation wavelength" as

$$\lambda_\varepsilon = \frac{k^{1/2}}{\omega} \tag{4.16}$$

The mean dissipation wavelength, λ_ϵ, can be compared to the mean vortex wavelength, λ. In nondimensional coordinates, Eq. (4.15) becomes

$$\begin{aligned} \frac{\nu_t}{u_\tau R} &= \left[\frac{\left(\frac{k}{u_\tau^2} \right)^{1/2}}{\frac{\omega R}{u_\tau}} \right] \left(\frac{k}{u_\tau^2} \right)^{1/2} \\ &= \left[\frac{k^{1/2}}{\frac{\omega}{R}} \right] \left(\frac{k}{u_\tau^2} \right)^{1/2} \end{aligned} \tag{4.17}$$

Both terms in brackets of Eq. (4.17) represent the nondimensional mean dissipation length scale which can be directly compared to the nondimensional mean vortex wavelength λ/R .

$$\frac{\nu_t}{u_\tau R} = \left[\frac{\lambda}{R} \right] \left(\frac{k}{u_\tau^2} \right)^{1/2} \tag{4.18}$$

Hence, the "pipe-scaled dimensionless mean dissipation wavelength" is given by

$$\hat{\lambda}_{\varepsilon} \equiv \frac{\lambda_{\varepsilon}}{R} \tag{4.19}$$

A comparison between the pipe-scaled dimensionless mean dissipation and mean vortex wavelengths is given in Fig. 4.28. The figure is based on a closure coefficient σ_k = 4.0, a proportionality constant for the wall turbulent kinetic energy k^+_{wall} = 0.1, a roughness ratio of R/k_s = 252 and a roughness Reynolds number k_s^+ of 80,000. From Fig. 4.28, it can be seen that the two dimensionless wavelengths have the same order of magnitude.

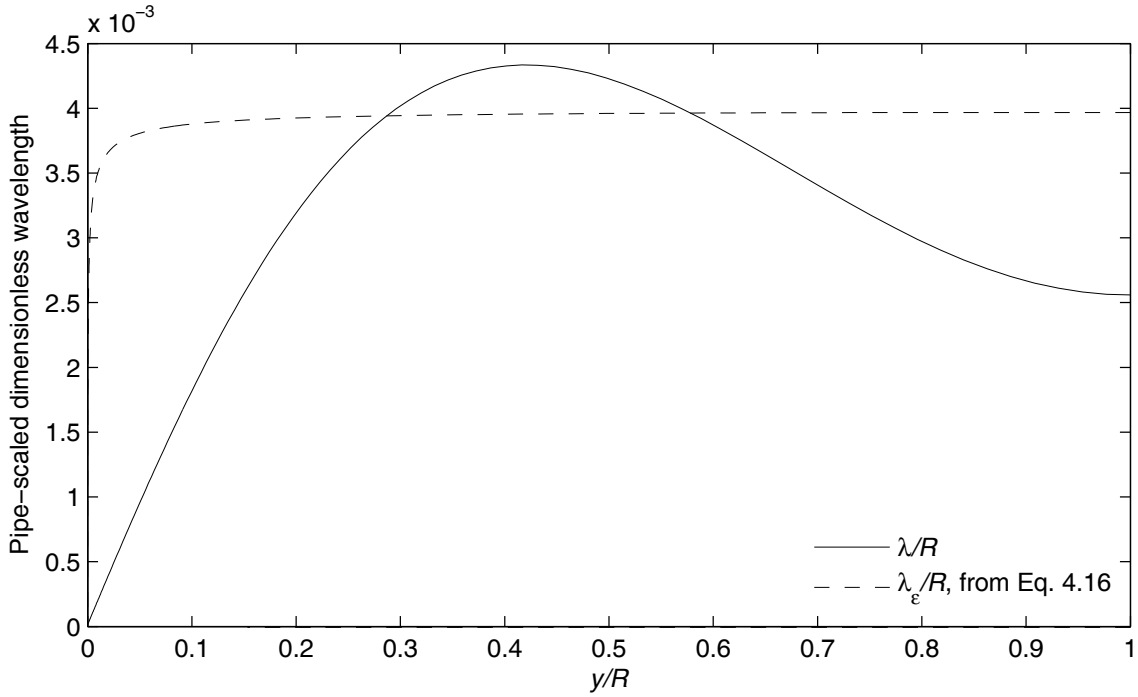

Fig. 4.28 Pipe-scaled dimensionless mean vortex and mean dissipation wavelengths.

B. Hunsaker 2011 Model

The one equation k-λ model requires a closure equation for the mean vortex wavelength. In 2011, Hunsaker [85] proposed a closure function. The resulting model could predict the friction factor very accurately, but the velocity distribution did not follow the log law. The current model based on a different closure function for the mean vortex wavelength is greatly improved as both the friction factor and velocity distribution are accurately predicted. Figure 4.30 compares the k-λ model resulting from the Hunsaker

closure function and the closure function given in Eqs. (3.17)–(3.35). The velocity distribution is greatly improved. The current model follows the log law.

The Hunsaker model gave "a %RMS error of 0.670" for the friction factor [85]. The current model gives a %RMS error in the range of 0.731 to 0.256 depending on the closure coefficient σ_k. As the closure coefficient is increased, the RMS error is decreased. The %RMS error of the new model is comparable to the 2011 version.

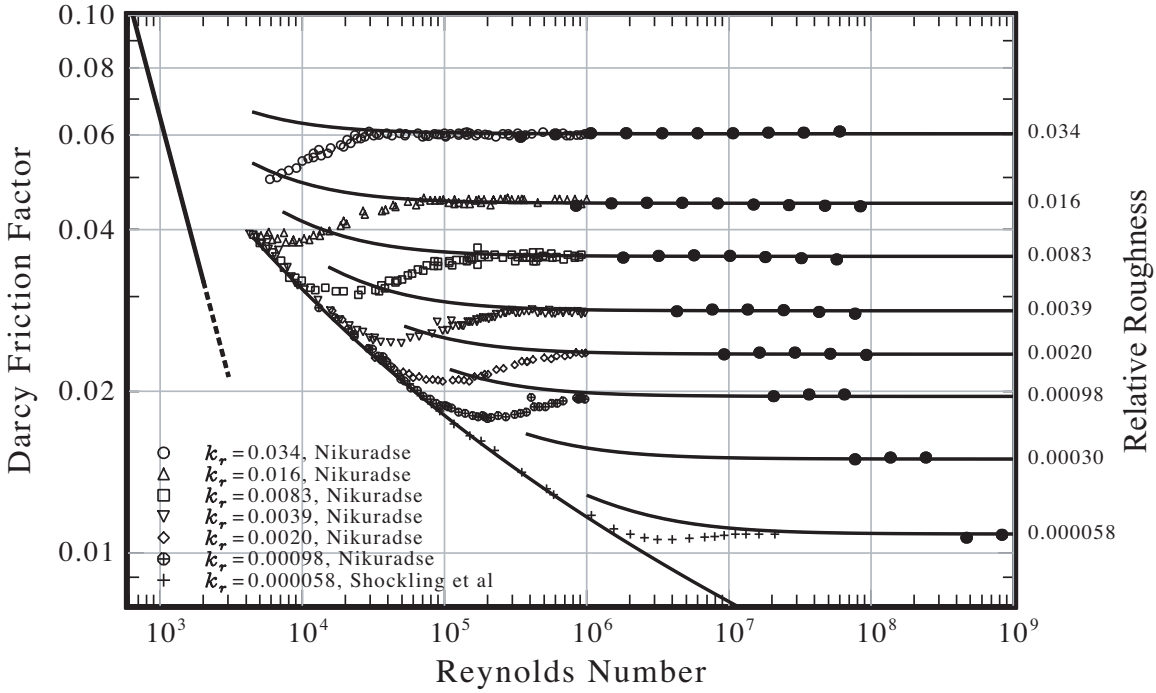

Fig. 4.29 Friction factor results, from the Hunsaker [85] model (reproduced with permission).

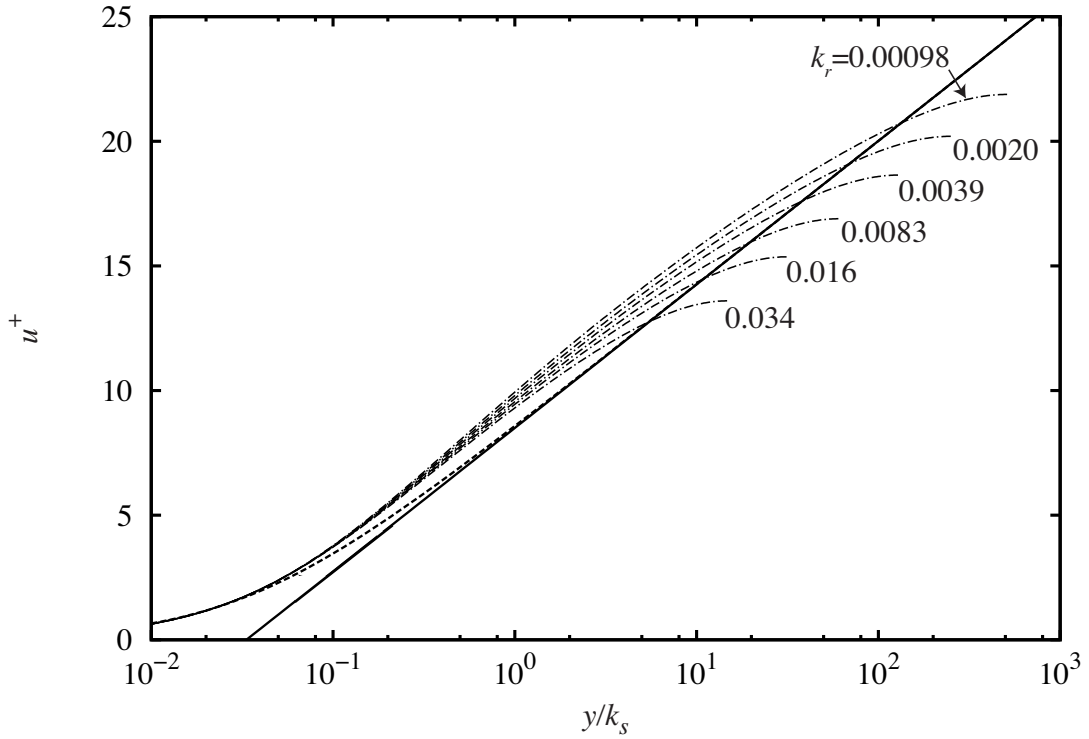

Fig. 4.30 Hunsaker [85] closure to the k-λ model, velocity-distribution comparison (reproduced with permission).

V. Summary and Conclusions

The focus of this work was to determine the distributions of the turbulence variables with the radial coordinate for fully rough pipe flows. These distributions were determined based on a one-equation *k*-λ turbulence model derived from a transport equation for the turbulent kinetic energy and an algebraic relation for the mean vortex wavelength. Many turbulence models in use today model the dissipation of turbulent kinetic energy as a transportable property, which is physically incorrect because dissipation is not a transportable property. Phillips' transport equation for the turbulent kinetic energy is presented in Chapter 1. An algebraic relation for a second turbulence variable, the mean vortex wavelength, was added to this transport equation to close the turbulence model for fully rough pipe flow. Chapter 2 presents the reasoning behind the choice of the mean-vortex-wavelength-profile function. This function features five unknown coefficients. The turbulent-kinetic-energy transport equation includes two closure coefficients and one unknown proportionality coefficient in the wall boundary condition. Chapter 3 derives functions for the unknown coefficients. Provided that certain algebraic relations between the model coefficients are

maintained, the model gives excellent agreement with well established relations obtained at fully rough flow. The evaluation of the coefficients was based on a rigorous optimization method. The algebraic relation for the mean vortex wavelength depends on the distance from the wall, the roughness Reynolds number, the proportionality coefficient in the wall boundary condition and one of the closure coefficients. Finally, Chapter 4 presents the results obtained from the current k-λ model along with a comparison to experimental data, well established empirical relations and other rough-wall models. The distribution of the turbulent kinetic energy, the mean vortex wavelength and the turbulent eddy viscosity are presented in Chapter 4.

The relations between the closure coefficients were developed such that the model gives excellent agreement for the friction factor and velocity distribution when compared to well established relations obtained for fully rough pipe flows. The friction factor was compared to the limit of the Colebrook equation from which the Moody diagram is based. The RMS error is less than 1% over the range of roughness ratios. The velocity distribution was compared to experimental data and the log law. The correlation to fully rough flow is excellent at any roughness ratio both in the core region and in the near-wall region. The model is significantly more accurate at predicting the friction factor and velocity distribution than the Wilcox 1998 and 2006 k-ω models. The closure that was developed here for Phillips' energy-vorticity turbulence model showed a significant improvement in evaluating the velocity distributions compared to that developed by Hunsaker [85]. From the mean-vortex-wavelength distribution and kinetic-energy-transport equation, the root-mean-square fluctuating vorticity and the turbulent eddy viscosity are calculated and presented.

The model needs to be extended to hydraulically smooth-wall pipe flow. The current model does not set a fixed value for one of the closure coefficients or for the proportionality coefficient in the wall boundary condition. Instead, a range of applicability is given for both of these parameters provided that the relations between the other coefficients are maintained. Extending the model to hydraulically smooth-wall flow might lead to a unique value for these parameters in order to match experimental velocity distributions and friction factors at any roughness Reynolds number. Even though the current empirical relation for the mean vortex wavelength might be modified for hydraulically smooth-wall flows, the complete model should converge to the current empirical function for fully rough pipe flow. The Wilcox 1998 and 2006

models predict the velocity distribution and friction factors fairly accurately for smooth-wall pipe flow. The resulting mean-vortex-wavelength function from Wilcox's model might be calculated. A function could be fit to this mean vortex wavelength and could be a starting point for developing an expression similar to the one currently used for fully rough pipe flow.

Once a robust model for flow in a pipe has been developed, a transport equation for the second turbulent variable might be developed. The present model, if extended to hydraulically smooth walls, could serve as a reference when developing a transport equation for the second turbulent variable. The vorticity is a transportable property and might be used as the second turbulent variable. This approach of deriving a transport equation for the second turbulent variable and comparing the results to the actual 1-D pipe flow model might lead to a more robust and accurate model. Deriving a transport equation would be one way to extend the model to 2- and 3-D flows.

REFERENCES

[1] Euler, L., "Principes Généraux de l'Etat d'Equilibre des Fluides," *Mémoire de l'Académie des Sciences de Berlin*, Vol. 11, 1757, pp. 217–273.

[2] Navier, C. L. M. H., "Mémoire sur les Lois du Mouvement des Fluides," *Mémoire de l'Académie Royales des Sciences*, Paris, Vol. 6, 1823, pp. 389–416.

[3] Stokes, G. G., "On the Theories of Internal Friction of Fluids in Motion," *Transactions of the Cambridge Philosophical Society*, Vol. 8, 1845, pp. 287–305.

[4] Reynolds, O., "On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion," *Philosophical transactions of the Royal Society of London*, Series A, Vol. 186, No. 4, 1895, pp. 123.

[5] Prandtl, L., "Über die ausgebildete Turbulenz," *Zeitschrift für Angewandte Mathematik und Mechanik,* Vol. 5, 1925, pp. 136–139.

[6] Boussinesq, J., "Essai sur la théorie des eaux courantes," *Mémoire présentés par divers savants à l'Académie des Sciences XXIII*, 1, 1877, pp. 1-680.

[7] Taylor, G. I., "The Spectrum of Turbulence," *Proceedings of the Royal Society of London*, Series A, Vol. 164, 1938, pp. 476–490.

[8] Kolmogorov, A. N., "Equations of Turbulent Motion of an Incompressible Fluid," *Izvestia Academy of Sciences*, USSR; Physics, Vol. 6, Nos. 1 and 2, 1942, pp. 56–58.

[9] Kraichnan, R., "Higher Order Interactions in Homogeneous Turbulence Theory," *Physics of Fluids,* Vol. 1, 1958, pp. 358.

[10] Prandtl, L., "Über ein neues Formelsystem für die ausgebildete Turbulenz," *Nachrichten Akademie der Wissenschaften zu Göttingen, Math-Phys. Klasse,* 1945, pp. 6–19.

[11] Launder, B. E., Morse, A., Rodi, W., and Spalding, D. B., "Prediction of Free Shear Flows – A Comparison of the Performance of Six Turbulence Models," *Proceedings of NASA Conference on Free Turbulent Shear Flows*, Langley Research Center, Hampton, VA, Vol. 1, 1972, pp. 361–426.

[12] Menter, F. R., "Eddy Viscosity Transport Equations and Their Relation to the k-ε Model," *Journal of Fluids Engineering*, Volume 119, Issue 4, 1997, pp. 876–884.

[13] Wilcox, D. C., *Turbulence Modeling for CFD*, Second Edition, DCW Industries, Inc., La Cañada CA, 1998, pp. 103-226.

[14] Wilcox D. C., "The *k-ω* Model," *Turbulence Modeling for CFD*, Third Edition, DCW Industries, Inc., La Cañada, CA, 2006, pp. 124–128.

[15] Menter, F. R., "Improved Two-Equation *k-ω* Turbulence Models for Aerodynamic Flows," NASA TM-103975, 1992.

[16] Wilcox, D. C., "Multiscale Model for Turbulent Flows," *AIAA Journal*, Vol. 26, No. 11, 1988, pp. 1311–1320.

[17] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA 92-439, 1992.

[18] Baldwin, B. S., and Barth, T. J., "A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows," NASA TM-102847, 1990.

[19] Launder, B. E. and Sharma, B. I., "Application of the Energy Dissipation Model of Turbulence to the Calculation of Flow Near a Spinning Disc," *Letters in Heat and Mass Transfer*, Vol. 1, No. 2, 1974, pp. 131–138.

[20] Speziale, C. G., Abid, R., and Anderson, E. C., "A Critical Evaluation of Two-Equation Models for Near Wall Turbulence," AIAA 90-1481, 1990.

[21] Wilcox D. C., "One-Equation and Two-Equation Models," *Turbulence Modeling for CFD*, Third Edition, DCW Industries, Inc., La Cañada, California, 2006, pp. 107–238.

[22] Bernard, P. S., and Wallace J. M., "Turbulent Flow Analyses," *Turbulent Flow*, Wiley, New-Jersey, 2002, pp. 14–16.

[23] Moin, P., and Mahesh, K., "Direct Numerical Simulation – A Tool in Turbulence Research," *Annual Review of Fluid Mechanics*, Vol. 30, 1998, pp. 539–578.

[24] Phillips, W. F., "Turbulent Flow in Newtonian Fluids," *Aerodynamics of Flight* DRAFT, Wiley, Hoboken, New Jersey, 2008, pp. 1–51.

[25] Hamilton, W. R., "On Quaternions: Or a New System of Imaginaries in Algebra," *Philosophical Magazine*, Vol. 25, No. 3, 1844, pp. 489–495.

[26] Hamilton, W. R., *Lectures on Quaternions,* Hodges and Smith, Dublin, 1853.

[27] Hamilton, W. R., *Elements of Quaternions*, Longmans, Green and Co., London, 1866.

[28] Boyer, C. B., and Merzback, U. C., *A History of Mathematics*, Second Edition, Wiley, New York, 1989, pp. 656–762.

[29] Wilcox, D. C., "One-Equation and Two-Equation Models," *Turbulence Modeling for CFD*, Third Edition, DCW Industries, Inc., La Cañada, California, 2006, pp. 34–38.

[30] Jones, W. P. and Launder, B. E., "The Prediction of Laminarization with a Two-Equation Model of Turbulence," *International Journal of Heat and Mass Transfer*, Vol. 15, 1972, pp. 301–314.

[31] Emmons, H. W., "Shear Flow Turbulence," *Proceedings of the Second U. S. Congress of Applied Mechanics*, ASME, 1954.

[32] Glushko, G., "Turbulent Boundary Layer on a Flat Plate in an Incompressible Fluid," *Izvestia Akademiya Nauk SSSR, Mekh.*, No. 4, 1965, pp. 13–30.

[33] Wolfshtein, M., "Convection Processes in Turbulent Impinging Jets," *Heat Transfer Section*, Imperial College, Report SF/R/2, 1967.

[34] Bradshaw, P., Ferriss, D. H., and Atwell, N. P., "Calculation of Boundary Layer Development Using the Turbulent Energy Equation," *Journal of Fluid Mechanics*, Vol. 28, Pt. 3, 1967, pp. 593–616.

[35] Nee, V. W., and Kovasznay, L. S. G., "The Calculation of the Incompressible Turbulent Boundary Layer by a Simple Theory," *Physics of Fluids*, Vol. 12, 1968, pp. 473.

[36] Sekundov, A. N., "Application of the Differential Equation for Turbulent Viscosity to the Analysis of Plane Non-Self-Similar Flows," *Akademiya Nauk SSSR*, Izvestiia, Mekhanika Zhidkosti i Gaza, 1971, pp. 114–127 (in Russian).

[37] Menter, F. R., “Eddy Viscosity Transport Equations and Their Relation to the *k-ε* Model,” *Journal of Fluids Engineering*, Volume 119, Issue 4, 1997, pp. 876–884.

[38] Rotta, J. C., “Statistische Theorie nichthomogener Turbulenz,” *Zeitschrift für Physik*, Vol. 129, 1951, pp. 547–572.

[39] Rotta, J. C., “Über eine Methode zur Berechnung turbulenter Scherströmungen,” *Aerodynamische Versuchanstalt Göttingen*, Rep. 69 A 14, 1968.

[40] Zeierman, S., and Wolfshtein, M., “Turbulent Time Scale for Turbulent Flow Calculations,” *AIAA Journal*, Vol. 24, No. 10, 1986, pp. 1606–1610.

[41] Huang, P. G., Coleman, G. H., and Bradhsaw, P., “Compressible Turbulent Channel Flows: DNS Results and Modeling,” *Journal of Fluid Mechanics*, Vol. 305, 1995, pp. 185–218.

[42] Taylor, G. I., “The Spectrum of Turbulence,” *Proceedings of the Royale Society of London*, Series A, Vol. 164, 1938, pp. 476-490.

[43] Hinze, J. O., “Isotropic Turbulence,” *Turbulence*, Second Edition, McGraw-Hill, New York, 1975, pp. 165–204.

[44] Hunsaker, D. F., *Evaluation of an Incompressible Energy-Vorticity Turbulence Model for Fully Rough Pipe Flow*, PhD dissertation, Utah State University, pp. 126-154.

[45] Prandtl, L. “Turbulenz und ihre entstehung,” *Journal of the Aeronautical Research Insitute*, Tokyo Imperial University, Nr . 65, 1930.

[46] Nikuradse, J., “Gesetzmassigkeiten der turbulenten stromung in glatten Rohren,” *VDI Forschungsheft* Vol. 356, 1932.

[47] Nikuradse, J., “Strömungsgesetze in rauhen Rohren,” *VDI Forschungsheft*, Vol. 361, 1933 [see also NACA TM-1292, 1950].

[48] Aupoix, B. and Spalart, P. R., "Extensions of the Spalart-Allmaras Turbulence Model to Account for Wall Roughness," *International Journal of Heat and Fluid Flow*, Vol. 24, 2003, pp. 454-462.

[49] Hellsten, A. and Laine, S., “Extension of the *k-ω* SST Turbulence Model for Flow over Rough Walls,” *AIAA Jounal*, Vol. 97, No. 6, 1997, pp. 3577.

[50] Launder, B. E., Reece, G. J. and Rodi, W. (1975), "Progress in the Development of a Reynolds-Stress Turbulent Closure," *Journal of Fluid Mechanics*, Vol. 68(3), pp. 537-566.

[51] White, F. M., “Viscous Fluid Flow,” *Viscous Fluid Flow*, Third Edition, McGraw-Hill, New York, 2006, pp. 433–440.

[52] Schlichting, H., “Rough Pipes and Equivalent Sand Roughness,” *Boundary Layer Theory*, Fourth Edition, McGraw-Hill, New York, 1960, pp. 519–527.

[53] Schlichting, H., *Grenzschicht-Theorie*, vormals G. Braunsche Hofbuchdruckerei und Verlag Gmbh, Karlsruhe, Germany, 1951.

[54] Robinson, D. E., Harris, J. E., and Hassan, H. A., “Unified Turbulence Closure Model for Axisymmetric and Planar Free Shear Flows,” *AIAA Journal*, Vol. 33, No. 12, 1995, pp. 2325–2331.

[55] Robinson, D. E. and Hassan, H. A., “Further Development of the *k-ζ* (Enstrophy) Turbulence Closure Model,” *AIAA Journal*, Vol. 36, No. 10, 1998, pp. 1825–1833.

[56] Darcy, H., "Recherches expérimentales relatives au mouvement de l'eau dans les tuyaux," *Mémoire de l'Académie des Sciences de l'Institut Impérial de France*, Bd. 15, 1858, p141.

[57] Blasius, H., "Das Ähnlichkeitsgesetz bei Reibungsvorgängen in Flüssigkeiten," *Forschungsarbeiten des Verein Deutscher Ingenieure,* No. 131, Berlin, 1913.

[58] Saph, V. and Schoder, E. H., "An Experimental Study of the Resistance to the Flow of Water in Pipes," *Transactions of the American Society of Civil Engineers*, Vol. 51, 1903, pp. 944.

[59] Nusselt, W., "Heat Transfer in Pipes," *Verein Deutscher Ingenieure Forschungsheft,* No. 89, Berlin, 1910.

[60] Ombeck, H., "Pressure Loss of Flowing Air in Straight Cylindrical Pipes*," Forschungsarbeiten des Verein Deutscher Ingenieure,* Vol. 158, Berlin, 1914.

[61] Stanton. T. E. and Pannell, J. R., "Similarity Motion in Relation to the Surface Friction in Fluids," *Proceedings of the Royal Society of London*, Vol. 214, 1914, pp. 199.

[62] Lees, Ch. H., "On the flow of Viscous Fluids through Smooth Circular Pipes," *Proceedings of the Royal Society of London*, Vol. 91, 1915, pp. 46.

[63] Jakob, M. and Erk, S., "The Pressure Drop in Smooth Pipes and the Coefficients of Flow of Normal Nozzles," *Verein Deutscher Ingenieure Forschungsheft,* No. 267, 1924, Berlin.

[64] Jimenez, J., "Turbulent Flow over Rough Walls," *Annual Review of Fluid Mechanics*, Vol. 36, 2004, pp. 173–196.

[65] Colebrook, C. F., "Turbulent Flow in Pipes with Particular Reference to the Transition Region Between the Smooth and Rough Pipe Laws," *Journal of the Institution of Civil Engineers*, Vol. 11, 1939, pp. 133–156.

[66] Moody, L. F., "Friction Factors for Pipe Flow," *Transactions of the A.S.M.E.,* Vol. 66, 1944, pp. 671–684.

[67] Schlichting, H., "Experimentelle Untersuchungen zum Rauhigkeitsproblem," *Ingenieure von der Architektur,* Vol. 7, 1936, pp. 1–34.

[68] Perry, A. E. and Abell, C. J., "Asymptotic Similarity of Turbulence Structures in Smooth and Rough Pipes," *Journal of Fluid Mechanics*, Vol. 79, 1977, pp. 785–799.

[69] McKeon, B. J., Li, J., Jiang, W., Morisson, J. F. and Smits, A. J., "Pitot Probe Corrections in the Fully-Developed Turbulent Pipe Flow," *Journal of Fluid Mechanics*, Vol 501, 2005, pp. 135–147.

[70] McKeon, B. J., Zaragola, M. V., and Smits, A. J., "A New Friction Factor Relationship for Fully Developed Pipe Flow," *Journal of Fluid Mechanics*, Vol 538, 2005, pp. 429–443.

[71] Zaragola, M. V., and Smits, A. J., "Mean-Flow Scaling of Turbulent Pipe Flow." *Journal of Fluid Mechanics*, Vol 373, pp. 33–79.

[72] Shockling, M. A., Allen, J. J., and Smits, A. J., "Roughness Effects in Turbulent Pipe Flow," *Journal of Fluid Mechanics*, Vol. 564, 2006, pp. 267–285.

[73] Ligrani, P. M. and Moffat, R. J., "Structure of Transitionally Rough and Fully Rough Turbulent Boundary Layers," *Journal of Fluid Mechanics,* Vol. 162, 1986, pp. 62–98.

[74] Perry, A. E., Hafez, S., and Chong, M. S., "A Theorical and Experimental Study of Wall Turbulence," *Journal of Fluid Mechanics*, Vol 165, 2001, pp. 395–401.

[75] Hama, F. R., "Boundary layer characteristics for smooth and rough surfaces," *Transactions of the Society of Naval Arcitects and Marine Engineers,* Vol. 62, 1954, pp. 333–358.

[76] Van Driest, E. R., "On Turbulent Flow Near a Wall," *Journal of Aerospace Sciences*, Vol. 23, 1956, pp. 1007–1011.

[77] Von Kármán, "Mechanische Ähnlichkeit und Turbulenz," *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen*, Fachgruppe Vol. 5, 1930, pp. 58–76.

[78] Anderson, P. S., Kays, W. M., and Moffat, R. J., "Experimental Results for the Transpired Turbulent Boundary Layer in an Adverse Pressure Gradient," *Journal of Fluid Mechanics*, Vol. 69, 1975, pp. 353–375.

[79] Reichardt, H., "Die Grundlagen des turbulenten Wärmeüberganges," *Architektur von der Gesellschaft Wärmetechnik 2*, 1951, pp. 129–142.

[80] Reichardt, H., *Zeitschrift fur angewandte Mathematik and Mechanik*, Vol. 20, No. 6, December 1940.

[81] Broyden, C. G., "The Convergence of a Class of Double-rank Minimization Algorithms," *Journal of the Institute of Mathematics and Its Applications,* Vol. 6, 1970, pp. 76–90.

[82] Fletcher, R., "A New Approach to Variable Metric Algorithms," *The Computer Journal*, Vol. 13, 1970, pp. 317–322.

[83] Goldfarb, D., "A Family of Variable Metric Methods Derived by Variational Means," *Mathematics of Computation*, Vol. 24, 1970, pp. 23–16.

[84] Shanno, D. F., "Conditioning of Quasi-Newton Methods for Function Minimization," *Mathematics of Computation*, Vol. 24, 1970, pp. 647–656.

[85] Hunsaker, D. F., *Evaluation of an Incompressible Energy-Vorticity Turbulence Model for Fully Rough Pipe Flow*, PhD dissertation, Utah State University, pp. 155-192.

[86] Kays, W. M. and Crawford, M. E., "Momentum Transfer: Turbulent Flow in Tubes," *Convective Heat and Mass Transfer*, Third Edition, McGraw-Hill, 1993, pp. 244–254.

[87] Saffman, P. G., "A Model for Inhomogeneous Turbulent Flow," *Proceedings of the Royal Society of London, Series A*, Vol. 317, 1970, pp. 417–433.

[88] Wilcox, D. C., "Reassessment of the Scale Determining Equation for Advanced Turbulence Models," *AIAA Journal*, Vol. 26, No. 11, 1988, pp. 1299–1310.

[89] Peng, S. H., Davidson, L., and Holmberg, S., "A Modified Low-Reynolds Number k-ω Model for Recirculating Flows," *Journal of Fluids Engineering*, Vol. 119, 1997, pp. 867–875.

[90] Kok, J. C., "Resolving the Dependence on Freestream Values for the k-ω Turbulence Model," *AIAA Journal*, Vol. 38, No. 7, 2000, pp. 1292–1295.

[91] Hellsten, A., "New Advanced k-ω Turbulence Model for High-Lift Aerodynamics," *AIAA Journal*, Vol. 43, No. 9, 2005, pp. 1857–1869.

[92] Fluent 6.3, "Standard k-ω Model," *FLUENT 6.3 User's Guide*, Fluent Inc., Lebanon, NH, Sept. 2006, pp. 12-26–12-31.

APPENDICES

APPENDIX A

MATHEMATICAL IDENTITIES

I. Vector Identities

$$\nabla \cdot (\mathbf{UV}) = (\mathbf{U} \cdot \nabla)\mathbf{V} + \mathbf{V}(\nabla \cdot \mathbf{U}) \tag{A.1}$$

$$2\nabla \cdot \bar{\bar{\mathbf{S}}}(\mathbf{V}) = \nabla^2 \mathbf{V} + \nabla(\nabla \cdot \mathbf{V}) = (\nabla \cdot \nabla)\mathbf{V} + \nabla(\nabla \cdot \mathbf{V}) \tag{A.2}$$

$$\nabla \cdot (S\mathbf{V}) = (\mathbf{V} \cdot \nabla)S + S(\nabla \cdot \mathbf{V}) \tag{A.3}$$

$$\mathbf{U} \cdot \left[\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{V} \cdot \nabla)\mathbf{U} \right] = \frac{\partial}{\partial t}(\tfrac{1}{2}U^2) + (\mathbf{V} \cdot \nabla)(\tfrac{1}{2}U^2) \tag{A.4}$$

$$\nabla \cdot \left(\bar{\bar{\boldsymbol{\sigma}}} \cdot \mathbf{V} \right) = \bar{\bar{\boldsymbol{\sigma}}} \cdot \left(\nabla \mathbf{V} \right) + \mathbf{V} \cdot \left(\nabla \cdot \bar{\bar{\boldsymbol{\sigma}}} \right) \tag{A.5}$$

II. Mathematical Operations

The double inner product is defined as the following

$$\begin{aligned} \bar{\bar{\boldsymbol{\tau}}} : \bar{\bar{\mathbf{J}}}(\mathbf{V}) = & \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix} : \begin{bmatrix} \frac{\partial V_x}{\partial x} & \frac{\partial V_x}{\partial y} & \frac{\partial V_x}{\partial z} \\ \frac{\partial V_y}{\partial x} & \frac{\partial V_y}{\partial y} & \frac{\partial V_y}{\partial z} \\ \frac{\partial V_z}{\partial x} & \frac{\partial V_z}{\partial y} & \frac{\partial V_z}{\partial z} \end{bmatrix} \\ = & \tau_{xx}\frac{\partial V_x}{\partial x} + \tau_{xy}\frac{\partial V_x}{\partial y} + \tau_{xz}\frac{\partial V_x}{\partial z} \\ & + \tau_{yx}\frac{\partial V_y}{\partial x} + \tau_{yy}\frac{\partial V_y}{\partial y} + \tau_{yz}\frac{\partial V_y}{\partial z} \\ & + \tau_{zx}\frac{\partial V_z}{\partial x} + \tau_{zy}\frac{\partial V_z}{\partial y} + \tau_{zz}\frac{\partial V_z}{\partial z} \end{aligned} \tag{A.6}$$

III. Flowfield Properties

$$\bar{\bar{\mathbf{S}}}(\mathbf{V})\cdot\bar{\bar{\mathbf{J}}}(\mathbf{V})=\bar{\bar{\mathbf{S}}}(\mathbf{V})\cdot\bar{\bar{\mathbf{S}}}(\mathbf{V}) \tag{A.7}$$

$$\mathbf{V}\cdot\left[\nabla\cdot\left(2\mu\bar{\bar{\mathbf{S}}}(\mathbf{V})\right)\right]=\nabla\cdot\left(\mu\left[\nabla\left(\tfrac{1}{2}V^2\right)+(\mathbf{V}\cdot\nabla)\mathbf{V}\right]\right)-2\mu\bar{\bar{\mathbf{S}}}(\mathbf{V}):\bar{\bar{\mathbf{S}}}(\mathbf{V}) \tag{A.8}$$

$$\left(\tilde{\mathbf{V}}\cdot\nabla\right)\tilde{\mathbf{V}}=\nabla\cdot\left(\tilde{\mathbf{V}}\tilde{\mathbf{V}}\right)-\tilde{\mathbf{V}}\left(\nabla\cdot\tilde{\mathbf{V}}\right) \tag{A.9}$$

$$\overline{\mathbf{V}}\cdot\left[\nabla\cdot\left(\tilde{\mathbf{V}}\tilde{\mathbf{V}}\right)\right]=\tilde{\mathbf{V}}\cdot\nabla\left(\overline{\mathbf{V}}\cdot\tilde{\mathbf{V}}\right)+\left(\overline{\mathbf{V}}\cdot\tilde{\mathbf{V}}\right)\left(\nabla\cdot\tilde{\mathbf{V}}\right)-\left(\tilde{\mathbf{V}}\tilde{\mathbf{V}}\right):\bar{\bar{\mathbf{J}}}\left(\overline{\mathbf{V}}\right) \tag{A.10}$$

$$\nabla\cdot\left(\rho\tilde{V}^2\tilde{\mathbf{V}}\right)=\rho\tilde{\mathbf{V}}\cdot\nabla\tilde{V}^2+\tilde{V}^2\nabla\cdot\left(\rho\tilde{\mathbf{V}}\right) \tag{A.11}$$

IV. Ensemble Averaging Identities

$$\overline{\overline{\tilde{\varphi}}}=0 \tag{A.12}$$

$$\begin{aligned}\overline{\varphi\psi}&=\overline{\varphi}\,\overline{\psi}+\overline{\varphi}\,\overline{\tilde{\psi}}+\overline{\tilde{\varphi}}\,\overline{\psi}+\overline{\tilde{\varphi}\tilde{\psi}}\\&=\overline{\varphi}\,\overline{\psi}+\overline{\tilde{\varphi}\tilde{\psi}}\end{aligned} \tag{A.13}$$

$$\overline{\varphi\psi\varsigma}=\overline{\varphi}\,\overline{\psi\varsigma}+\overline{\tilde{\varphi}\tilde{\psi}\varsigma}+\overline{\psi}\,\overline{\tilde{\varphi}\tilde{\varsigma}}+\overline{\varphi}\,\overline{\tilde{\psi}\tilde{\varsigma}}+\overline{\tilde{\varphi}\tilde{\psi}\tilde{\varsigma}} \tag{A.14}$$

$$\overline{\frac{\partial f}{\partial x}}=\frac{\partial\bar{f}}{\partial x} \tag{A.15}$$

V. Flowfield Tensors

The strain-rate tensor is given by

$$\bar{\bar{\mathbf{S}}}(\mathbf{V}) \equiv \frac{1}{2}\begin{bmatrix} \left(\frac{\partial V_x}{\partial x}+\frac{\partial V_x}{\partial x}\right) & \left(\frac{\partial V_x}{\partial y}+\frac{\partial V_y}{\partial x}\right) & \left(\frac{\partial V_x}{\partial z}+\frac{\partial V_z}{\partial x}\right) \\ \left(\frac{\partial V_y}{\partial x}+\frac{\partial V_x}{\partial y}\right) & \left(\frac{\partial V_y}{\partial y}+\frac{\partial V_y}{\partial y}\right) & \left(\frac{\partial V_y}{\partial z}+\frac{\partial V_z}{\partial y}\right) \\ \left(\frac{\partial V_z}{\partial x}+\frac{\partial V_x}{\partial z}\right) & \left(\frac{\partial V_z}{\partial y}+\frac{\partial V_y}{\partial z}\right) & \left(\frac{\partial V_z}{\partial z}+\frac{\partial V_z}{\partial z}\right) \end{bmatrix} \tag{A.16}$$

The Jacobian tensor of a vector field is

$$\bar{\bar{\mathbf{J}}}(\mathbf{V}) \equiv \begin{bmatrix} \frac{\partial V_x}{\partial x} & \frac{\partial V_x}{\partial y} & \frac{\partial V_x}{\partial z} \\ \frac{\partial V_y}{\partial x} & \frac{\partial V_y}{\partial y} & \frac{\partial V_y}{\partial z} \\ \frac{\partial V_z}{\partial x} & \frac{\partial V_z}{\partial y} & \frac{\partial V_z}{\partial z} \end{bmatrix} \tag{A.17}$$

The rotation tensor is

$$\bar{\bar{\mathbf{\Omega}}}(\mathbf{V}) \equiv \frac{1}{2}\begin{bmatrix} \left(\frac{\partial V_x}{\partial x}-\frac{\partial V_x}{\partial x}\right) & \left(\frac{\partial V_x}{\partial y}-\frac{\partial V_y}{\partial x}\right) & \left(\frac{\partial V_x}{\partial z}-\frac{\partial V_z}{\partial x}\right) \\ \left(\frac{\partial V_y}{\partial x}-\frac{\partial V_x}{\partial y}\right) & \left(\frac{\partial V_y}{\partial y}-\frac{\partial V_y}{\partial y}\right) & \left(\frac{\partial V_y}{\partial z}-\frac{\partial V_z}{\partial y}\right) \\ \left(\frac{\partial V_z}{\partial x}-\frac{\partial V_x}{\partial z}\right) & \left(\frac{\partial V_z}{\partial y}-\frac{\partial V_y}{\partial z}\right) & \left(\frac{\partial V_z}{\partial z}-\frac{\partial V_z}{\partial z}\right) \end{bmatrix} \tag{A.18}$$

APPENDIX B

EXPERIMENTAL DATA

I. Nikuradse Experimental Rough Flow Data

Nikuradse [47] studied the effect of coarse and fine pipe roughness over a wide range of Reynolds numbers in view of determining the laws governing the friction factor and the velocity profiles. Experimental data were obtained at six different degrees of relative roughness with Reynolds numbers ranging from $\mathrm{Re} = 10^4$ to 10^6. Nikuradse measured the pressure drop and the velocity distribution by means of pitot tubes. The experiments were conducted on three brass pipes of circular cross-section having different diameters. Uniform grain sizes were used to produce a uniform roughness throughout the pipe. The derivative du/dy was obtained graphically from the velocity distribution and was tabulated by Nikuradse [47].

1. Friction Factor Data

Table B.1 Nikuradse's friction factor data [47] for a range of roughness Reynolds number

$R/k_s = 507$											
$\log(\mathrm{Re})$	4.114	4.230	4.322	4.362	4.362	4.462	4.491	4.532	4.568	4.591	4.623
$\log(100 \times 4C_f)$	0.456	0.438	0.417	0.407	0.403	0.381	0.38	0.366	0.365	0.356	0.347
$\log(\mathrm{Re})$	4.672	4.69	4.716	4.763	4.806	4.851	4.898	4.94	4.973	5.009	5.025
$\log(100 \times 4C_f)$	0.333	0.324	0.320	0.307	0.303	0.292	0.286	0.278	0.274	0.274	0.272
$\log(\mathrm{Re})$	5.049	5.100	5.143	5.199	5.236	5.27	5.281	5.303	5.326	5.377	5.40
$\log(100 \times 4C_f)$	0.270	0.262	0.26	0.255	0.253	0.255	0.253	0.250	0.252	0.255	0.253
$\log(\mathrm{Re})$	5.493	5.534	5.574	5.608	5.63	5.668	5.709	5.756	5.792	5.833	5.940
$\log(100 \times 4C_f)$	0.258	0.26	0.262	0.290	0.272	0.272	0.272	0.278	0.279	0.283	0.286
$\log(\mathrm{Re})$	5.965	5.929	5.954	5.987							
$\log(100 \times 4C_f)$	0.288	0.289	0.288	0.286							

$R/k_s = 252$									
$\log(\mathrm{Re})$	4.210	4.279	4.465	4.507	4.549	4.597	4.644	4.778	4.820
$\log(100 \times 4C_f)$	0.4506	0.4349	0.3808	0.3636	0.3579	0.3562	0.3434	0.3257	0.3282
$\log(\mathrm{Re})$	4.916	4.987	5.057	5.100	5.173	5.210	5.283	5.366	5.494

$\log(100\times 4C_f)$	0.3222	0.3197	0.321	0.3228	0.3197	0.3276	0.3322	0.3416	0.3504
$\log(\mathrm{Re})$	5.580	5.623	5.702	4.708	5.305	5.544	5.787	4.748	4.869
$\log(100\times 4C_f)$	0.3562	0.3602	0.3636	0.3371	0.3328	0.3562	0.3661	0.3335	0.3228
$\log(\mathrm{Re})$	4.954	5.134	5.255	5.415	5.580	5.748	5.845	5.881	5.924
$\log(100\times 4C_f)$	0.321	0.321	0.3294	0.3434	0.3551	0.3608	0.3666	0.3688	0.3727
$\log(\mathrm{Re})$	5.967	5.991							
$\log(100\times 4C_f)$	0.3705	0.3716							

$R/k_s = 126$										
$\log(\mathrm{Re})$	3.630	3.675	3.715	3.760	3.810	3.833	3.895	3.925	3.950	3.965
$\log(100\times 4C_f)$	0.594	0.588	0.576	0.566	0.552	0.564	0.532	0.515	0.503	0.498
$\log(\mathrm{Re})$	4.015	4.111	4.196	4.265	4.330	4.386	4.425	4.470	4.496	4.511
$\log(100\times 4C_f)$	0.491	0.471	0.451	0.435	0.424	0.415	0.412	0.400	0.396	0.400
$\log(\mathrm{Re})$	4.550	4.620	4.697	4.76	4.820	4.910	4.985	5.057	5.121	5.164
$\log(100\times 4C_f)$	0.393	0.392	0.391	0.400	0.403	0.408	0.414	0.422	0.424	0.430
$\log(\mathrm{Re})$	5.591	5.616	5.655	5.675	5.708	5.736	5.756	5.775	5.798	5.831
$\log(100\times 4C_f)$	0.450	0.453	0.447	0.450	0.445	0.452	0.445	0.445	0.450	0.45
$\log(\mathrm{Re})$	5.835	5.874	5.894	5.935	5.961	5.97	5.987	4.950	5.049	5.021
$\log(100\times 4C_f)$	0.446	0.450	0.447	0.450	0.444	0.449	0.447	0.430	0.432	0.415
$\log(\mathrm{Re})$	5.10	5.130	5.179	5.196	5.225	5.225	5.250	5.274	5.290	5.310
$\log(100\times 4C_f)$	0.422	0.422	0.430	0.430	0.435	0.430	0.436	0.438	0.438	0.436
$\log(\mathrm{Re})$	5.330	5.350	5.366	5.393	5.423	5.432	5.455	5.476	5.501	5.525
$\log(100\times 4C_f)$	0.439	0.439	0.444	0.444	0.446	0.447	0.450	0.452	0.447	0.447
$\log(\mathrm{Re})$	5.560									
$\log(100\times 4C_f)$	0.450									

$R/k_s = 60$										
$\log(\mathrm{Re})$	3.653	3.700	3.740	3.785	3.851	3.869	3.909	3.849	3.996	4.057
$\log(100\times 4C_f)$	0.593	0.571	0.571	0.56	0.544	0.531	0.512	0.512	0.507	0.494
$\log(\mathrm{Re})$	4.090	4.161	4.236	4.290	4.391	4.412	4.512	4.54	4.553	4.580
$\log(100\times 4C_f)$	0.490	0.494	0.487	0.487	0.401	0.489	0.49	0.487	0.498	0.493
$\log(\mathrm{Re})$	4.609	4.694	4.665	4.699	4.740	4.769	4.813	4.849	4.93	4.954
$\log(100\times 4C_f)$	0.507	0.504	0.507	0.509	0.519	0.52	0.528	0.526	0.543	0.534
$\log(\mathrm{Re})$	5.034	5.155	5.083	5.185	5.231	4.875	4.924	4.954	5.052	5.033
$\log(100\times 4C_f)$	0.543	0.543	0.545	0.55	0.537	0.535	0.534	0.542	0.535	0.540
$\log(\mathrm{Re})$	5.130	5.170	5.196	5.23	5.258	5.283	5.312	5.35	5.408	5.47
$\log(100\times 4C_f)$	0.545	0.550	0.547	0.568	0.551	0.555	0.551	0.555	0.550	0.555
$\log(\mathrm{Re})$	5.497	5.515	5.549	5.554	5.750	5.600	5.621	5.625	5.641	5.655
$\log(100\times 4C_f)$	0.543	0.551	0.55	0.558	0.551	0.550	0.560	0.543	0.543	0.550
$\log(\mathrm{Re})$	5.659	5.668	5.691	5.714	5.748	5.757	5.789	5.836	5.865	5.914

$\log(100\times 4C_f)$	0.551	0.56	0.553	0.551	0.558	0.550	0.551	0.547	0.555	0.553
$\log(\mathrm{Re})$	5.916	5.945	5.962							
$\log(100\times 4C_f)$	0.550	0.551	0.555							

$R/k_s = 30.6$										
$\log(\mathrm{Re})$	3.672	3.708	3.748	3.763	3.785	3.826	3.869	3.881	3.929	3.935
$\log(100\times 4C_f)$	0.592	0.59	0.592	0.597	0.583	0.585	0.596	0.578	0.578	0.583
$\log(\mathrm{Re})$	3.978	4.009	4.049	4.079	4.124	4.130	4.390	4.270	4.290	4.309
$\log(100\times 4C_f)$	0.578	0.585	0.583	0.592	0.59	0.599	0.599	0.609	0.618	0.612
$\log(\mathrm{Re})$	4.584	4.653	4.799	4.900	4.965	5.029	5.068	5.134	5.176	4.425
$\log(100\times 4C_f)$	0.639	0.644	0.647	0.656	0.656	0.652	0.650	0.650	0.650	0.637
$\log(\mathrm{Re})$	4.44	4.56	4.636	4.740	4.830	4.855	4.990	5.100	5.240	5.275
$\log(100\times 4C_f)$	0.63	0.637	0.647	0.654	0.654	0.661	0.657	0.652	0.657	0.657
$\log(\mathrm{Re})$	5.323	5.473	5.655							
$\log(100\times 4C_f)$	0.647	0.657	0.652							

$R/k_s = 15$										
$\log(\mathrm{Re})$	3.770	3.820	3.855	3.905	3.955	4.000	4.041	4.076	4.079	4.114
$\log(100\times 4C_f)$	0.696	0.699	0.707	0.712	0.717	0.730	0.734	0.736	0.744	0.751
$\log(\mathrm{Re})$	4.133	4.179	4.196	4.270	4.290	4.314	4.34	4.366	4.386	4.410
$\log(100\times 4C_f)$	0.740	0.744	0.754	0.76	0.756	0.769	0.763	0.778	0.772	0.772
$\log(\mathrm{Re})$	4.425	4.466	4.520	4.590	4.630	4.725	4.811	4.865	4.885	4.965
$\log(100\times 4C_f)$	0.782	0.785	0.78	0.781	0.777	0.78	0.781	0.777	0.776	0.779
$\log(\mathrm{Re})$	5.000	5.042	5.098	5.155	5.179	5.285	4.440	4.500	4.540	4.596
$\log(100\times 4C_f)$	0.781	0.780	0.781	0.776	0.781	0.779	0.775	0.777	0.778	0.780
$\log(\mathrm{Re})$	4.685	4.722	4.845	4.869	4.929	4.949	5.002	5.005	5.097	5.139
$\log(100\times 4C_f)$	0.781	0.777	0.775	0.778	0.780	0.779	0.777	0.775	0.778	0.783
$\log(\mathrm{Re})$	5.156	5.220	5.236	5.310	5.360	5.410	5.446	5.455	5.515	5.567
$\log(100\times 4C_f)$	0.784	0.777	0.780	0.778	0.775	0.780	0.780	0.777	0.781	0.778
$\log(\mathrm{Re})$	5.613	5.690	5.834	5.882	5.959	6.008	5.793	5.857	5.93	5.987
$\log(100\times 4C_f)$	0.780	0.784	0.781	0.777	0.778	0.780	0.780	0.777	0.778	0.780

B. Velocity Profile Data

Table B.2 Nikuradse's velocity profile data

R/k_s	15	15	15	15	30.6	30.6	30.6	30.6
D (cm)	4.82	2.412	2.412	2.412	9.64	4.87	4.87	2.434

$\overline{V}_{bulk}$	873	956	524	215	734	796	420	459
ν	0.0098	0.0117	0.0117	0.0121	0.0111	0.01046	0.0105	0.0107
Re	430,000	197,000	108,000	43,000	638,000	372,000	195,000	104,000
u_τ	76.1	85	46.6	18.87	55.7	60	31.9	34.8
k_s^+	1230	579	319	124	805	458	245	130
y/R	$\overline{V}_z$							
0.00	0	0	0	0	0	0	0	0
0.02	450	532	270	107	430	461	215	258
0.04	549	628	320	136	500	535	270	328
0.07	643	710	367	158	570	610	325	367
0.10	702	767	406	174	616	660	351	395
0.15	773	846	456	191	670	718	380	424
0.20	830	910	492	206	715	766	404	447
0.30	917	1006	545	228	777	833	440	484
0.40	978	1080	585	243	819	880	467	509
0.50	1027	1143	617	253	853	918	489	531
0.60	1070	1195	647	263	881	950	507	551
0.70	1107	1237	660	271	904	976	519	568
0.80	1137	1267	689	277	925	994	530	580
0.90	1162	1290	701	283	940	1008	539	587
0.96	1172	1296	704	285	948	1016	541	590
0.98	1174	1298	706	286	951	1017	542	591
1.00	1176	1300	707	286.5	952	1018	543	592

R/k_s	60	60	60	126	126	126
D (cm)	9.8	9.8	9.8	9.92	9.92	9.92
$\overline{V}_{bulk}$	774	514	309	820	575	374
ν	0.0112	0.0115	0.0112	0.0085	0.0089	0.0089
Re	677,000	438,000	271,000	960,000	640,000	417,000
u_τ	51.9	34.4	20.72	49	34.3	22.28
k_s^+	370	239	146	230	154	100
y/R	$\overline{V}_z$					
0.00	0	0	0	0	0	0
0.02	475	316	193	557	383	250
0.04	557	370	224	619	436	284
0.07	624	411	245	676	476	312
0.10	666	442	266	718	506	331
0.15	714	474	286	760	536	350
0.20	753	500	300	798	564	366
0.30	800.5	536	323	852	597	390

0.40	848	562	338	887	619	407
0.50	879	582	353	916	640	419
0.60	904	600	362	939	657	431
0.70	925	616	370	959	667	439
0.80	941	628	378	979	678	448
0.90	955	636	384	988	690	453
0.96	962	640	387	994	696	456
0.98	964	641.4	387.5	996	698	457
1.00	966	642	388	998	701	458

Table B.3 Nikuradse's velocity gradient profile data

R/k_s	15		30.6		60	
D (cm)	4.82		9.64		9.8	
$\overline{V}_{bulk}$	873		734		774	
ν	0.0098		0.0111		0.0112	
Re	430,000		638,000		677,000	
u_τ	76.1		55.7		51.9	
k_s^+	1230		805		370	
y/R	u	$d\overline{V}_z/dy$	u	$d\overline{V}_z/dy$	u	$d\overline{V}_z/dy$
0	0		0		0	
0.02	450	3850	430	1444	475	1240
0.04	549	1985	500	728	557	640
0.07	643	1140	570	419	624	373
0.1	702	830	616	302	666	268
0.15	773	578	670	209	714	189
0.2	830	457	715	162	753	146
0.3	917	321	777	115	800.5	105
0.4	978	246	819	88	848	80
0.5	1027	201	853	72	879	65
0.6	1070	166	881	59	904	54
0.7	1107	136	904	49	925	43
0.8	1137	108	925	39	941	35
0.9	1162	74	940	26	955	24
0.96	1172	47	948	16.5	962	15
0.98	1174	33	951	11.7	964	10.7
1	1176		952		966	

R/k_s	126	252	507
D (cm)	9.92	4.924	9.94

$\overline{V}_{bulk}$	820		1127		838	
ν	0.0085		0.89		0.86	
Re	960,000		624		970	
u_τ	49		61		41.7	
k_s^+	230		70		49	
y / R	u	$d\overline{V}_z / dy$	u	$d\overline{V}_z / dy$	u	$d\overline{V}_z / dy$
0	0		0		0	
0.02	557	1670	532	2954	608	1022
0.04	619	616	794	1526	670	526
0.07	676	380	875	895	725	309
0.1	718	262	951	643	761	222
0.15	760	181	1000	443	803	153
0.2	798	138	1062	346	832	119
0.3	852	96	1107	247	874	84
0.4	887	74	1178	188	912	64
0.5	916	60	1225	154	940	52.5
0.6	939	50	1266	127	960	43.6
0.7	959	41	1303	104	978	35.5
0.8	979	31	1350	83	992	28.4
0.9	988	22.5	1366	56	1003	19.2
0.96	994	14.2	1369	35	1008	12
0.98	996	10	1372	25	1010	8.7
1	998		1373		1011	

The dimensionless velocity gradient is obtained from Nikuradse's tabulated pipe diameter, velocity gradient and shear velocity

$$\frac{du^+}{d\hat{y}} = \frac{d\overline{V}_z}{dy}\frac{D/2}{u_\tau} \tag{B.1}$$

Prandtl's mixing-length theory states that

$$\frac{\tau}{\rho} = \ell^2 \left|\frac{d\overline{V}_z}{dy}\right| \frac{d\overline{V}_z}{dy} \tag{B.2}$$

The mixing length is determined from the velocity profiles by

$$\ell = \sqrt{\frac{\tau / \rho}{\left(d\bar{V}_z / dy\right)^2}} \tag{B.3}$$

The shear stress τ at any point is linearly related to the wall shear stress according to

$$\tau = \tau_w \left(1 - \frac{y}{R}\right) \tag{B.4}$$

Using Eqs. (B.1), (B.3) and (B.4), the mixing length expressed only in terms of Nikuradse's tabulated data becomes

$$\ell = \frac{\sqrt{1 - \frac{y}{R}}}{\frac{d\bar{V}_z}{dy}\frac{D/2}{u_\tau}} \tag{B.5}$$

Similarly, the eddy viscosity expressed only in terms of Nikuradse's tabulated data is

$$\frac{\nu_t}{u_\tau R} = \frac{1 - \frac{y}{R}}{\frac{d\bar{V}_z}{dy}\frac{D/2}{u_\tau}} \tag{B.6}$$

II. Shockling Experimental Rough Flow Data

Table B.4 Shockling [72] friction factor data

$k_s / D = 0.000058$								
Re	56800	68210	80680	92550	116400	150800	181100	224200
$4C_f$	0.02064	0.01961	0.01894	0.01836	0.0174	0.01653	0.01614	0.0155
Re	349000	518900	586200	1079000	1556000	2016000	2535000	3138000
$4C_f$	0.01412	0.01318	0.01283	0.01174	0.01109	0.01084	0.01068	0.01058
Re	4005000	5270000	6726000	7955000	9265000	11130000	13160000	16290000
$4C_f$	0.01058	0.01064	0.01071	0.01074	0.01077	0.01084	0.01084	0.01084
Re	21110000							
$4C_f$	0.01084							

APPENDIX C

EMPIRICAL TURBULENT EDDY VISCOSITY FUNCTION

I. The Von Kármán Constant and the Nikuradse Number Determined from the Friction Factor

The Von Kármán constant and the Nikuradse number are optimized by comparing the Colebrook equation to the friction factor obtained from the near-wall fully rough limit of the law of the wall. Nikuradse [47] noticed that the friction factor is independent of the Reynolds number and depends only on the roughness height for fully rough flow. The most widely accepted form of the Nikuradse equation is

$$4C_f = \frac{1}{\left[2.0\log_{10}\left(\frac{3.7}{k_s / D}\right)\right]^2} \tag{C.1}$$

Equation (C.1) can be rewritten in terms of the pipe radius R in place of the pipe diameter D

$$4C_f = \frac{1}{\left[2.0\log_{10}\left(7.4\frac{R}{k_s}\right)\right]^2} \quad \text{or} \quad 4C_f = \frac{1}{\left[2.0\log_{10}\left(7.4\gamma\,\hat{R}_s\right)\right]^2} \quad \text{with} \quad \hat{R}_s = \frac{R}{\gamma\,k_s} \tag{C.2}$$

Based on Prandtl's mixing length theory [5], the near-wall fully rough limit formulation is

$$\frac{u_\tau \ell}{\nu}\frac{du^+}{dy^+} = 1\,, \qquad u^+\left(y^+ = 0\right) = 0 \tag{C.3}$$

In the near-wall region, the results obtained by Nikuradse for the fully rough limit are in excellent agreement with the empirical correlation

$$\ell = \kappa\left(y + \gamma\,k_s\right) \tag{C.4}$$

Substituting Eq. (C.3) into Eq. (C.4) and integrating the result yields the fully rough near-wall expression for the dimensionless velocity

$$u^+ = \frac{1}{\kappa}\ln\left(\frac{y^+}{\gamma\, k_s^+} + 1\right) \tag{C.5}$$

The wall-scaled dimensionless velocity u^+ may be rewritten in terms of the relative dimensionless velocity $\hat{u} \equiv u^+ / u_m^+$. Adding a function δ to satisfy the symmetry boundary condition at the pipe centerline and to provide better agreement with Nikuradse's velocity profile data yields

$$\hat{u}\left(\hat{y}, \hat{R}_s\right) = \frac{\hat{R}_s{}^2 \ln\left(\hat{R}_s \hat{y} + 1\right)}{\left(\hat{R}_s + 1\right)^2\left[\ln\left(\hat{R}_s + 1\right) - \frac{3}{2}\right] + 2\hat{R}_s + \frac{3}{2}} + \delta\left(\hat{y}, \hat{R}_s\right) \tag{C.6}$$

In terms of the wall-scaled variable u^+, this last equation may be rewritten as

$$u^+\left(\hat{y}, \kappa, \hat{R}_s\right) = \frac{\ln\left(\hat{R}_s \hat{y} + 1\right)}{\kappa} + \frac{\left(\hat{R}_s + 1\right)^2\left[\ln\left(\hat{R}_s + 1\right) - \frac{3}{2}\right] + 2\hat{R}_s + \frac{3}{2}}{\kappa \hat{R}_s{}^2}\delta\left(\hat{y}, \hat{R}_s\right) \tag{C.7}$$

The Darcy friction factor corresponding to this fully rough velocity profile is

$$4C_f = \frac{8\kappa^2 \hat{R}_s{}^4}{\left\{\left(\hat{R}_s + 1\right)^2\left[\ln\left(\hat{R}_s + 1\right) - \frac{3}{2}\right] + 2\hat{R}_s + \frac{3}{2}\right\}^2} \tag{C.8}$$

An optimization program is run to estimate the Von Kármán constant κ and the Nikuradse number γ by minimizing the difference between Eq. (C.1) and (C.8) over a range of roughness ratio R / k_s. This optimization code was developed by Hunsaker [85] and is derived from a BFGS method, named after the work of Broyden [81], Fletcher [81], Goldfarb [81] and Shanno [81] who each derived the same method independently in the same year. Nikuradse provides experimental data for the following six roughness ratio values $R / k_s = [15 \ ; \ 30.6 \ ; \ 60 \ ; \ 126 \ ; \ 252 \ ; \ 507]$. This same range was used during the optimization. The scalar to minimize is defined as the root mean square of the difference between the Darcy friction factors obtained from Eqs. (C.1) and (C.8).

$$RMS = \sqrt{\frac{1}{6}\sum_{i=1}^{6}\left[\text{Eq.}(B.1)_i - \text{Eq}(B.8)_i\right]^2} \tag{C.9}$$

This gave the following values for the Von Kármán constant κ and the Nikuradse number γ

$$\begin{aligned} \kappa &= 0.403120174922297 \\ \gamma &= 0.0323477786281198 \end{aligned} \tag{C.10}$$

The residual defined in Eq. (B.9) that minimizes the differences between Eq. (C.1) and Eq. (C.8) over the discrete range $R/k_s = [15\ ;\ 30.6\ ;\ 60\ ;\ 126\ ;\ 252\ ;\ 507]$ is found to be 0.0000410661935. However when only 3 significant digits are used for κ and γ, the minimum yields the following constants

$$\begin{aligned} \kappa &= 0.403 \\ \gamma &= 0.0324 \end{aligned} \tag{C.11}$$

For $\kappa = 0.403$ and $\gamma = 0.0324$, the residual defined in Eq. (C.9) is 0.0000412132063896745. For $\kappa = 0.403$ and $\gamma = 0.0323$, the residual defined in Eq. (C.9) is 0.000059565297143975. Therefore, truncated to 3 significant digits, the values for κ and γ that minimize the difference between the friction factors obtained from the Nikuradse equation given in Eq. (C.1) and from that given in Eq. (C.8) based on the fully rough mixing-length theory are $\kappa = 0.403$ and $\gamma = 0.0324$.

II. Deviation Between the Near-wall Fully Rough Limit and Nikuradse's Experimental Data

The function δ represents a velocity deviation added to the near-wall fully rough limit in order to match Nikuradse's experimental data [47] better. It satisfies the symmetry boundary condition at the centerline and obtains a friction factor that correlates to the Colebrook equation in the fully rough limit. The near-wall fully rough limit velocity is based on an empirical correlation derived from Prandtl's mixing length

$$u^+ = \frac{1}{\kappa}\ln\left(\frac{y^+}{\gamma\, k_s^+} + 1\right) \tag{C.12}$$

The nondimensional bulk velocity is obtained by integrating Eq. (C.12)

$$u_m^+ = \frac{\int_{\hat{r}=0}^{1} u^+ \hat{r} d\hat{r}}{\int_{r=0}^{1} \hat{r} d\hat{r}} = 2\int_{\hat{r}=0}^{1} u^+ \hat{r} d\hat{r} = \frac{1}{\kappa}\left(\frac{\gamma\, k_s}{R}\right)^2 \left\{\left(\frac{R}{\gamma\, k_s} + 1\right)^2 \left[\ln\left(\frac{R}{\gamma\, k_s} + 1\right) - \frac{3}{2}\right] + 2\frac{R}{\gamma\, k_s} + \frac{3}{2}\right\} \tag{C.13}$$

The Von Kármán constant κ and the Nikuradse number γ used in Eq. (C.12) that minimize the difference in the friction factor obtained from the Colebrook equation and that based on Eq. (C.13) are given in Eq. (C.11).

Although based on careful measurements, it was found that Nikuradse's tabulated results present some slight inconsistencies in the bulk velocity that had to be corrected in order to determine the deviation between the experimental results and the near-wall fully rough limit .Although Nikuradse does provide the value of the bulk velocity for each of his experiments $u_{bulk}^{\exp}$, this value is not used in the calculation of the function δ but is instead calculated u_{bulk}^{calc} based on a trapezoidal integration of the experimental data. Table C**.1** presents the deviations between Nikuradse's experimental velocities and the velocities obtained from the near-wall fully rough limit at a roughness ratio R/k_s=15. Table C.1 is based on the constants $\kappa = 0.403$ and $\gamma = 0.0324$. The dimensional velocity $u^{\exp}$ refers to the dimensional experimental velocity tabulated by Nikuradse. The column $\left(\frac{u^+}{u_m^+}\right)_{\text{F.R.}}$ refers to the near-wall fully rough limit and is the ratio of Eq. (C.12) with Eq. (C.13). The function δ is defined as the difference between the nondimensional experimental velocities and the nondimensional near-wall fully rough limit velocities obtained from the ratio of Eq. (C.12) with Eq. (C.13). The trapezoidal numerical integrals are second-order accurate. When $y/R = 1$, the trapezoidal integral of $\frac{u^{\exp}}{u_{bulk}^{calc}}$ and $\left(\frac{u^+}{u_m^+}\right)_{\text{F.R.}}$ does not give 1.0 but 0.901106438707911. Because both integrands are concave down functions without inflection points, both integral are underestimated because

there is a systematic area unaccounted for under the curve. A correction to the trapezoidal integration of these concave down functions needs to be applied. The three trapezoidal integrals are all corrected by dividing each column by the factor: 0.901106438707911.

Table C.1 Deviation between Nikuradse's experimental velocities and the near-wall fully rough limit

R/k_s	15
D (cm)	4.82
$u_{bulk}^{\exp}$	873
u_{bulk}^{calc}	871.2296
ν	0.0098
Re	430000
u_τ	76.1
k_s^+	$10^{3.9} = 1230.269$
u_{bulk}^+ from Eq. (C.13)	11.57369

$\frac{y}{R} = 1 - \hat{r}$	$u^{\exp}$	$\frac{u^{\exp}}{u_{bulk}^{calc}}$	$\left(\frac{u^+}{u_m^+}\right)_{\text{F.R.}}$	$\delta = \frac{u^{\exp}}{u_{bulk}^{calc}} - \left(\frac{u^+}{u_m^+}\right)_{\text{F.R.}}$
0.10	702	0.805758	0.826816	-0.02106
0.15	773	0.887252	0.912231	-0.02498
0.20	830	0.952677	0.973147	-0.02047
0.30	917	1.052535	1.059314	-0.00678
0.40	978	1.122551	1.120609	0.001942
0.50	1027	1.178794	1.16822	0.010573
0.60	1070	1.228149	1.207156	0.020993
0.70	1107	1.270618	1.240096	0.030522
0.80	1137	1.305052	1.268643	0.036409
0.90	1162	1.333747	1.293831	0.039916
0.96	1172	1.345225	1.307636	0.037589
0.98	1174	1.347521	1.312047	0.035474
1.00	1176	1.349816	1.316369	0.033447

Trapezoidal Numerical Integrals			
$\frac{y}{R} = 1 - \hat{r}$	$2\int_{\hat{r}=0}^{\hat{r}_i} u^{\exp}\hat{r}d\hat{r}$	$2\int_{\hat{r}=0}^{\hat{r}_i} \frac{u^{\exp}}{u_{bulk}^{calc}}\hat{r}d\hat{r}$	$2\int_{\hat{r}=0}^{\hat{r}_i} \left(\frac{u^+}{u_m^+}\right)_{\text{F.R.}} \hat{r}d\hat{r}$
0.10	18.95400	0.021755	0.022324
0.15	83.39650	0.095723	0.098301
0.20	149.4490	0.171538	0.175996
0.30	280.0390	0.321430	0.328000
0.40	402.9090	0.462460	0.469389
0.50	512.9390	0.588753	0.595036
0.60	607.0890	0.696819	0.701733
0.70	683.0990	0.784063	0.787222

0.80	739.0490	0.848283	0.849798
0.90	773.4090	0.887721	0.888109
0.96	783.1938	0.898952	0.899011
0.98	784.6010	0.900567	0.900582
1.00	785.0706	0.901106	0.901106

	Corrected Numerical Integrals		
$\frac{y}{R}=1-\hat{r}$	$2\int_{\hat{r}=0}^{\hat{r}_i} u^{\exp}\hat{r}d\hat{r}$	$2\int_{\hat{r}=0}^{\hat{r}_i} \frac{u^{\exp}}{u_{bulk}^{calc}}\hat{r}d\hat{r}$	$2\int_{\hat{r}=0}^{\hat{r}_i} \left(\frac{u^+}{u_m^+}\right)_{\text{F.R.}}\hat{r}d\hat{r}$
	corrected	corrected	corrected
0.10	21.03414	0.024143	0.024774
0.15	92.54900	0.106228	0.109089
0.20	165.8506	0.190364	0.195311
0.30	310.7724	0.356705	0.363997
0.40	447.127	0.513214	0.520902
0.50	569.2324	0.653367	0.660339
0.60	673.7151	0.773292	0.778746
0.70	758.0669	0.870112	0.873618
0.80	820.1573	0.941379	0.943061
0.90	858.2882	0.985146	0.985577
0.96	869.1468	0.997609	0.997674
0.98	870.7085	0.999402	0.999418
1.00	871.2296	1.000000	1.000000

Nikuradse's experimental velocities [47] for fully rough flow defined as $k_s^+ \geq 100$ are given in Table C.**2** along with the deviation from the near-wall fully rough limit velocities. The limit $k_s^+ \geq 100$ corresponds to the ratio of the turbulent eddy viscosity and molecular viscosity greater than 1.

Table C.2 Deviation between Nikuradse's experimental velocities and the near-wall fully rough limit

R/k_s	15		15		15		15	
D (cm)	4.82		2.412		2.412		2.412	
$u_{bulk}^{\exp}$	873		956		524		215	
u_{bulk}^{calc}	871.2295976		962.5939431		519.5674783		215.1930024	
ν	0.0098		0.0117		0.0117		0.0121	
Re	430000		197000		108000		43000	
u_τ	76.1		85		46.6		18.87	
k_s^+	1230.268771		579.4286964		319.1537855		124.4514612	
y/R	$u^{\exp}$	Deviation δ	$u^{\exp}$	Deviation δ	$u^{\exp}$	Deviation δ	$u^{\exp}$	Deviation δ
0.10	702	-0.021058	767	-0.03001	406	-0.0454	174	-0.01824
0.15	773	-0.024979	846	-0.03336	456	-0.03458	191	-0.02466
0.20	830	-0.020471	910	-0.02779	492	-0.02621	206	-0.01587
0.30	917	-0.006778	1006	-0.01422	545	-0.01036	228	0.0002
0.40	978	0.001942	1080	0.001359	585	0.005327	243	0.00861

0.50	1027	0.010573	1143	0.019196	617	0.019306	253	0.007468
0.60	1070	0.020993	1195	0.034281	647	0.03811	263	0.015002
0.70	1107	0.030522	1237	0.044973	660	0.030191	271	0.019238
0.80	1137	0.036409	1267	0.047592	689	0.05746	277	0.018574
0.90	1162	0.039916	1290	0.046298	701	0.055368	283	0.021267
0.96	1172	0.037589	1296	0.038726	704	0.047337	285	0.016756
0.98	1174	0.035474	1298	0.036393	706	0.046776	286	0.016993
1.00	1176	0.033447	1300	0.034149	707	0.044378	286.5	0.014994

R/k_s	30.6		30.6		30.6		30.6	
D (cm)	9.64		4.87		4.87		2.434	
$u_{bulk}^{\exp}$	734		796		420		459	
u_{bulk}^{calc}	738.29902		793.09555		420.9104		462.178	
ν	0.0111		0.01046		0.0105		0.0107	
Re	638000		372000		195000		104000	
u_τ	55.7		60		31.9		34.8	
k_s^+	805.37844		458.14189		244.9063		130.017	
y/R	$u^{\exp}$	Deviation δ	$u^{\exp}$	Deviation δ	$u^{\exp}$	Deviation δ	$u^{\exp}$	Deviation δ
0.10	616	-0.015314	660	-0.01748	351	-0.01576	395	0.004985
0.15	670	-0.017095	718	-0.01927	380	-0.02178	424	-0.00719
0.20	715	-0.009438	766	-0.01204	404	-0.01806	447	-0.01072
0.30	777	-0.000708	833	-0.00281	440	-0.00777	484	-0.00591
0.40	819	0.002722	880	0.002992	467	0.002916	509	-0.00528
0.50	853	0.007281	918	0.009412	489	0.01369	531	0.000831
0.60	881	0.011289	950	0.015844	507	0.022538	551	0.010187
0.70	904	0.013757	976	0.019941	519	0.022362	568	0.018284
0.80	925	0.017347	994	0.017784	530	0.023642	580	0.019394
0.90	940	0.015737	1008	0.01351	539	0.023098	587	0.012614
0.96	948	0.014558	1016	0.011581	541	0.015834	590	0.007089
0.98	951	0.014782	1017	0.009003	542	0.014371	591	0.005414
1.00	952	0.012375	1018	0.006502	543	0.012985	592	0.003816

R/k_s	60		60		60	
D (cm)	9.8		9.8		9.8	
$u_{bulk}^{\exp}$	774		514		309	
u_{bulk}^{calc}	771.5091217		512.9097565		309.0379227	
ν	0.0112		0.0115		0.0112	
Re	677000		438000		271000	
u_τ	51.9		34.4		20.72	
k_s^+	369.8281798		238.7811283		145.881426	
y/R	$u^{\exp}$	Deviation δ	$u^{\exp}$	Deviation δ	$u^{\exp}$	Deviation δ
0.10	666	-0.00325	442	-0.00474	266	-0.00576
0.15	714	-0.00795	474	-0.00927	286	-0.00796
0.20	753	-0.00495	500	-0.00613	300	-0.0102
0.30	800.5	-0.01045	536	-0.00301	323	-0.00285

0.40	848	0.003501	562	6.56E-05	338	-0.00193
0.50	879	0.006734	582	0.002111	353	0.009663
0.60	904	0.008942	600	0.007009	362	0.00859
0.70	925	0.010628	616	0.01267	370	0.008943
0.80	941	0.009245	628	0.013945	378	0.012709
0.90	955	0.007878	636	0.010028	384	0.01261
0.96	962	0.006258	640	0.007134	387	0.011624
0.98	964	0.005434	641.4	0.006447	387.5	0.009826
1.00	966	0.004679	642	0.004269	388	0.008097

R/k_s	126		126		126	
D (cm)	9.92		9.92		9.92	
$u_{bulk}^{\exp}$	820		575		374	
u_{bulk}^{calc}	816.4728933		571.950574		374.448781	
ν	0.0085		0.0089		0.0089	
Re	960000		640000		417000	
u_τ	49		34.3		22.28	
k_s^+	229.6148648		154.1700453		100	
y/R	$u^{\exp}$	Deviation δ	$u^{\exp}$	Deviation δ	$u^{\exp}$	Deviation δ
0.10	718	-0.00181	506	0.003488	331	0.002763
0.15	760	-0.01013	536	-0.00382	350	-0.00626
0.20	798	-0.00602	564	0.002702	366	-0.00596
0.30	852	0.000289	597	0.000572	390	-0.00169
0.40	887	0.000695	619	-0.00342	407	0.001246
0.50	916	0.003273	640	0.000352	419	0.000352
0.60	939	0.004525	657	0.003157	431	0.005481
0.70	959	0.00626	667	-0.00212	439	0.004086
0.80	979	0.011039	678	-0.0026	448	0.008404
0.90	988	0.00467	690	0.000985	453	0.004365
0.96	994	0.002488	696	0.001945	456	0.002847
0.98	996	0.001893	698	0.002397	457	0.002473
1.00	998	0.001359	701	0.004659	458	0.00216

The δ function is represented using a seventh-order polynomial function whose coefficients were selected based on mathematical and physical constraints and in order to match Nikuradse experimental data for fully rough flow. The three coefficients C_0, $\hat{\nu}_t\big|_{\hat{y}=1}$ and C_7 are not based on mathematical or physical constraints and were tuned to provide the best visual fit for the δ function through the discretized deviation given in Table C.2. Only the points close to the centerline ($y/R \geq 0.5$) were taken into consideration. The closure coefficients were found to be

$$C_0 = 1.2\left(\frac{k_s}{R}\right)^{1.4} \quad , \quad \hat{\nu}_t\big|_{\hat{y}=1} = 0.056 \text{ and } \quad C_7 = -0.65 \tag{C.14}$$

Figure C.1 shows the deviation between Nikuradse's experimental velocities and the near-wall fully rough velocity at different roughness ratios and also displays the δ function. It can be seen that the δ function is within the experimental scatter. The δ function is defined as

$$\delta = C_0 + C_1\hat{r} + C_2\hat{r}^2 + C_3\hat{r}^3 + C_4\hat{r}^4 + C_5\hat{r}^5 + C_6\hat{r}^6 + C_7\hat{r}^7 \tag{C.15}$$

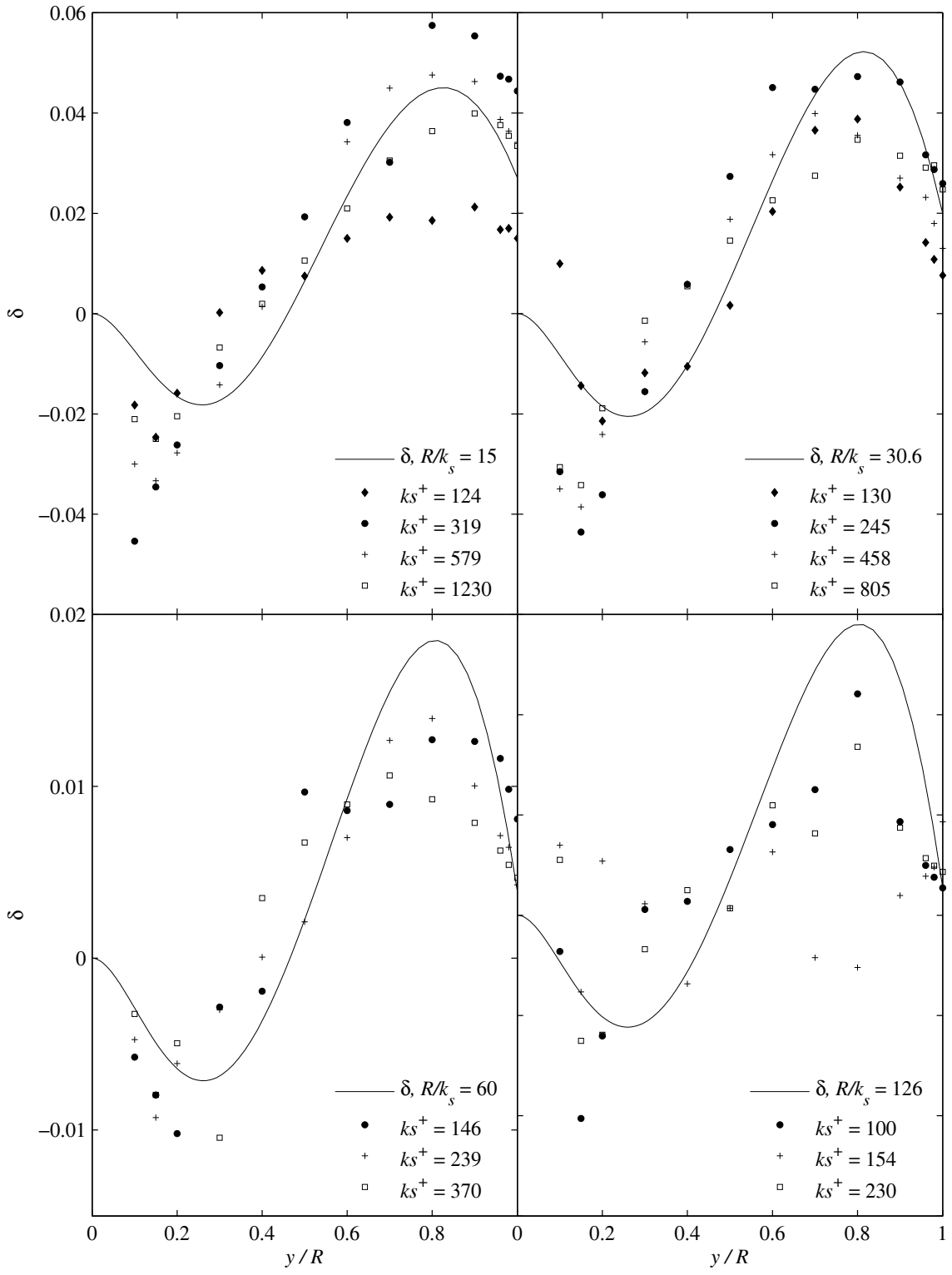

Fig. C.1 Experimental deviation between Nikuradse's velocities and the near-wall fully rough velocity along with the δ function at different roughness ratio.

III. Turbulent Eddy Viscosity Empirical Function

A δ function is added to the near-wall fully rough velocity correlation in order to match Nikuradse's experimental data, satisfy the symmetry boundary condition and provide a friction factor, which corresponds to the Colebrook fully rough limit. The near-wall fully rough velocity equation yields a friction factor that matches the experimental pressure drop observed relatively well. However, this velocity equation does not satisfy the centerline symmetry boundary condition. A function δ is added to the near-wall fully rough limit equation such that the bulk velocity is conserved and the velocity distribution is slightly modified to satisfy the centerline boundary condition and match experimental data for a range of roughness ratio and Reynolds number corresponding to fully rough flow. A seventh-order polynomial is used for the δ function, as shown in Eq. (C.15).

The δ function features eight coefficients that need to be evaluated. Five of them (C_1, C_3, C_4, C_5, C_6) are obtained from constraints imposed to satisfy the wall and symmetry boundary conditions, the near-wall fully rough limit behavior, the bulk velocity and the eddy viscosity symmetry boundary condition. The remaining three coefficients, not determined by mathematical or physical constraints, are C_0, C_2 and C_7. It is possible to express C_2 in terms of the centerline dimensionless eddy viscosity $\hat{\nu}_{tc}$. Therefore, the three coefficients that needs to be determined based on Nikuradse's experimental data [47] are C_0, C_7 and $\hat{\nu}_{tc}$.

The first coefficient evaluated based on Nikuradse experimental data is C_0. From Eq. (C.15), the coefficient C_0 is determined from the value of the δ function at the centerline for which $\hat{r} = 0$. Because Nikuradse's centerline data depends on the roughness ratio k_s / R, the coefficient C_0 is proposed to be

$$C_0 = A\left(\frac{k_s}{R}\right)^e + B \tag{C.16}$$

where the three coefficients A, B and e need to be evaluated. Table C.3 gives the centerline deviation δ between Nikuradse's experimental data and the near-wall fully rough limit. Nikuradse's experimental centerline velocities are given in Table B.2.

Table C.3 Centerline Deviation between the near-wall fully rough limit and Nikuradse's experimental velocities.

R/k_s	$\gamma k_s/R$	δ
15.0	2.16000E-03	3.34474E-02
15.0	2.16000E-03	3.41485E-02
15.0	2.16000E-03	4.43782E-02
15.0	2.16000E-03	1.49940E-02
30.6	1.05882E-03	1.23749E-02
30.6	1.05882E-03	6.50244E-03
30.6	1.05882E-03	1.29853E-02
30.6	1.05882E-03	3.81617E-03
60.0	5.40000E-04	4.67867E-03
60.0	5.40000E-04	4.26937E-03
60.0	5.40000E-04	8.09654E-03
126.0	2.57143E-04	1.35941E-03
126.0	2.57143E-04	4.65895E-03
126.0	2.57143E-04	2.15980E-03

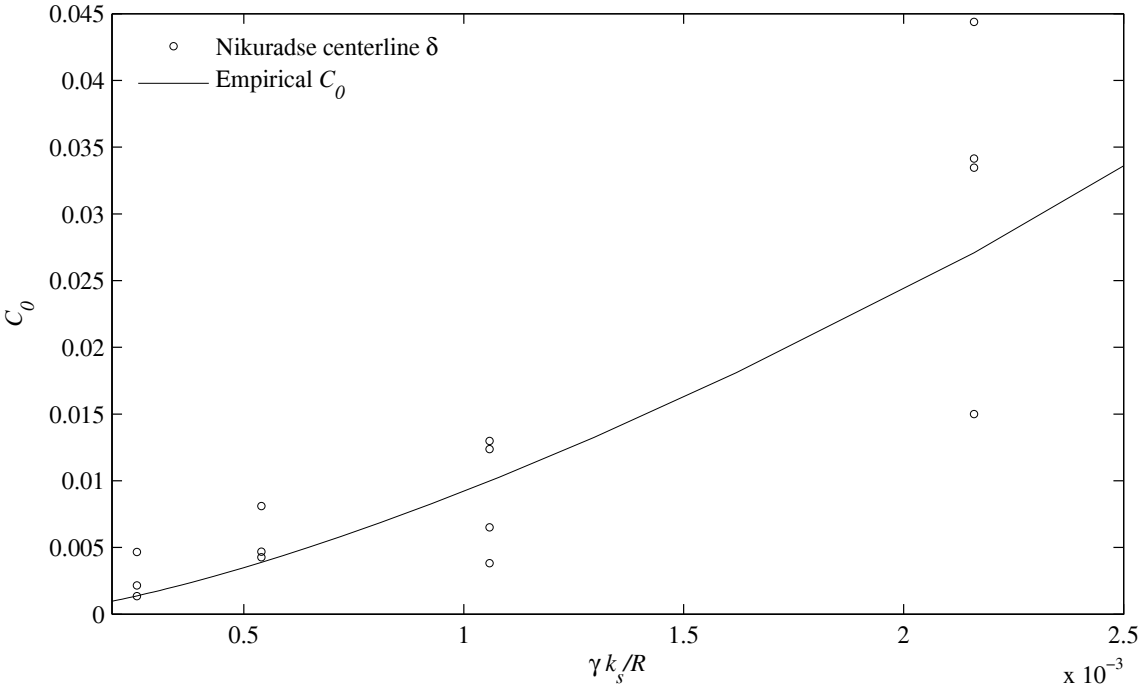

Fig. C.2 Nikuradse's experimental data deviation at the pipe centerline along with the empirical function C_0 for $A = 1.2$, $B = 0.0$ and $e = 1.4$.

Figure C.2 shows the coefficient C_0 obtained from Eq. (C.16) having $A = 1.2$, $B = 0.0$ and $e = 1.4$. This set of coefficients fits the data points with high roughness ratio (15 and 30.6) relatively well and is in the range of uncertainty for the lowest Reynolds roughness ratio (60 and 126).

The two other coefficients from the empirical δ function given in Eq. (C.15) that need to be determined from Nikuradse's experimental data [47] are the centerline eddy viscosity $\hat{\nu}_{tc}$ and the coefficient C_7. An exhaustive search is used in place of an optimization code for the remaining two coefficients $\hat{\nu}_{tc}$ and C_7 due to the numerous local minimum present in the function minimizing the difference between Nikuradse's experimental velocities and the near-wall fully rough limit velocity profile. The dimensionless eddy viscosity $\hat{\nu}_{tc}$ is discretized every 0.005 ranging from 0.03 to 0.06. For each of those points, the coefficient C_7 is varied from -12.0 to 2.0 with a step of 0.01. The empirical eddy viscosity and velocity may now be computed at each of these points. The nondimensional velocity $\hat{u}$ is then compared with Nikuradse's corrected nondimensional velocity profiles at each of the 14 cases available for fully rough flow (4 cases for $R/k_s = 15$, 4 for $R/k_s = 30.6$, 3 for $R/k_s = 60$ and 3 for $R/k_s = 126$). These corrected velocities are given in Table C.2. A fitness function is defined to be the square of the difference between the corrected experimental velocities with the velocities obtained from the empirical eddy viscosity and discretized at the eight available experimental points located at the relative coordinate $y/R \geq 0.5$. The points closer to the wall tabulated by Nikuradse present a lot more experimental uncertainty and were ignored during this optimization process. For each of the discretized turbulent eddy viscosities $\hat{\nu}_{tc}$, the value of C_7 that gives the minimum fitness is selected by fitting a parabola through three of the discretized C_7 points, which surround the minimum. The fitness function is then plotted against $\hat{\nu}_{tc}$ over the range 0.03 to 0.06. A fourth order function is fit through the points and features a systematic minimum in the neighborhood of $\hat{\nu}_{tc} = 0.04$. The coefficient C_7 is plotted as a function of the centerline eddy viscosity $\hat{\nu}_{tc}$ and a cubic spline is fit through the data. It should be noted that the point of minimum fitness features several inflection points in the corresponding nondimensional turbulent eddy viscosity and is therefore ignored. The C_7 coefficient is increased from its minimum value so that the turbulent eddy viscosity features only one inflection point in the core region over the range of roughness

ratio $R/k_s = [15-126]$ and even extended to smooth pipes featuring a roughness ratio $R/k_s = 10{,}000$. Table C.4 gives the optimum C_7 coefficients obtained at each discretized centerline turbulent eddy viscosity along with its respective fitness value.

Table C.4 Deviation function coefficients.

A	*B*	*e*	$\hat{\nu}_{tc}$	C_7	Fitness
1.2	0	1.4	0.03	-11.19015	0.00920869
1.2	0	1.4	0.035	-7.938638	0.00840393
1.2	0	1.4	0.04	-5.500003	0.00815821
1.2	0	1.4	0.045	-3.603288	0.00820237
1.2	0	1.4	0.05	-2.085915	0.00838489
1.2	0	1.4	0.055	-0.844429	0.00862570
1.2	0	1.4	0.06	0.190144	0.00888425

Figures C.3 and C.4 show the variations of the coefficients C_7 and $\hat{\nu}_{tc}$ given in Table C.4. Even though the fitness shows a minimum value at $\hat{\nu}_{tc} = 0.041$, the turbulent eddy viscosity shows several inflection points, as shown in Fig. C.5. The value of $\hat{\nu}_{tc}$ that gives satisfactory results over the range of roughness ratios $R/k_s = 15$ to $R/k_s = 126$ is $\hat{\nu}_{tc} = 0.056$.

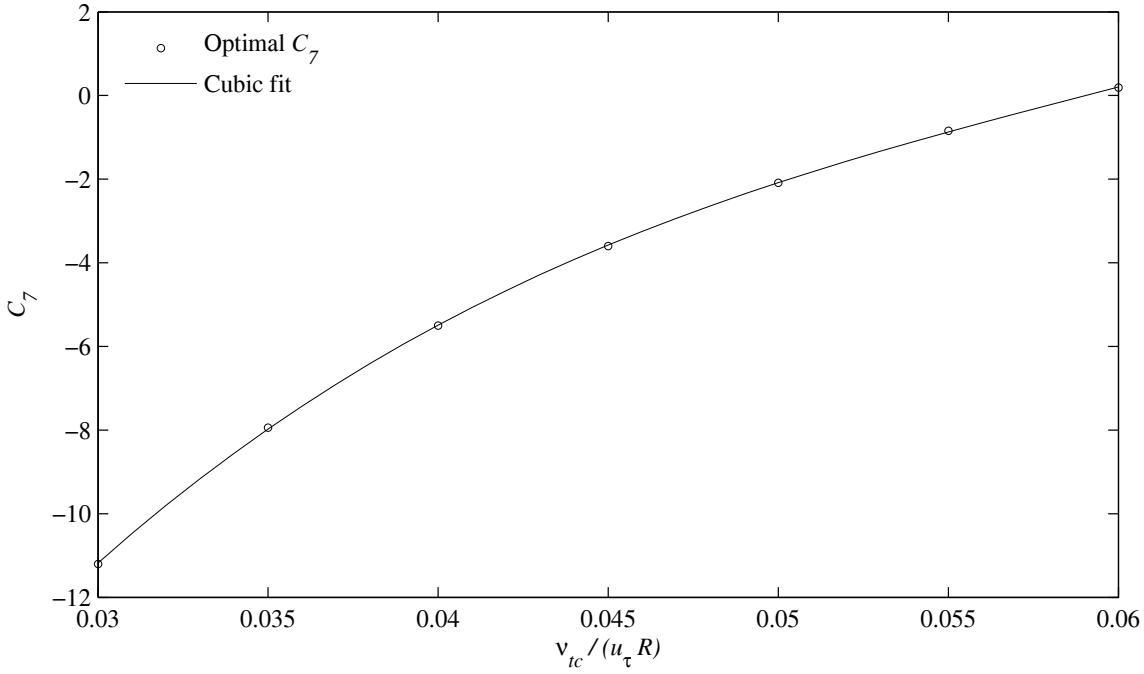

Fig. C.3 Coefficient C_7 variations as a function of the centerline turbulent eddy viscosity.

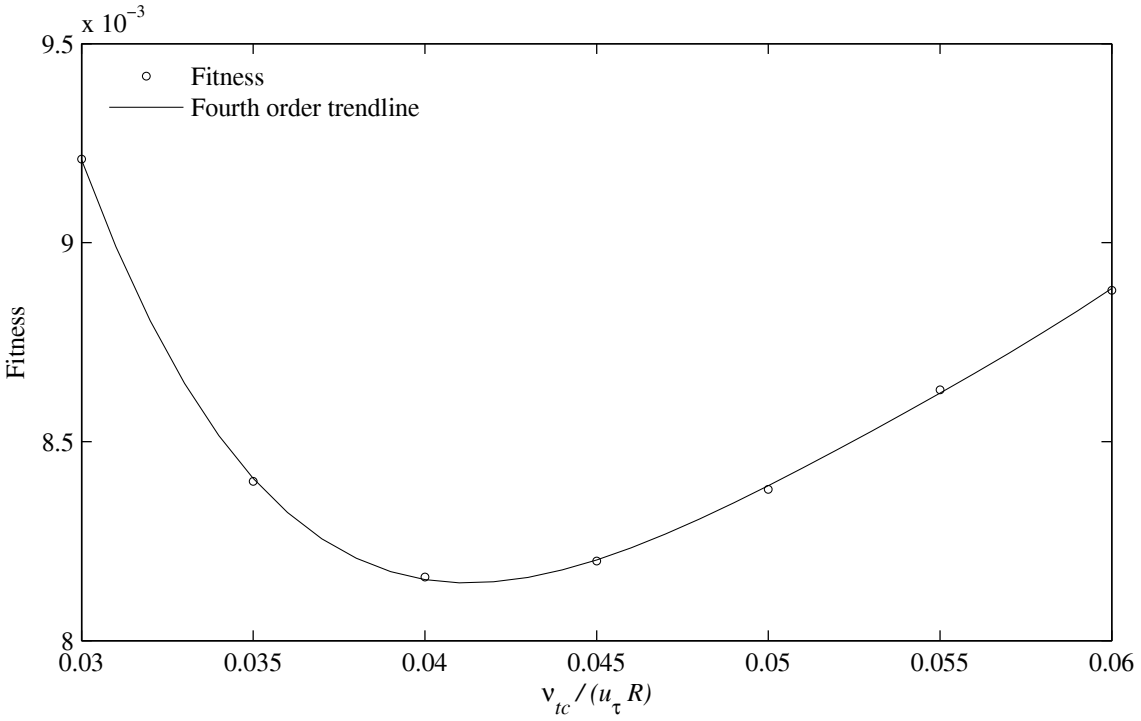

Fig. C.4 Fitness function variations with the centerline turbulent eddy viscosity.

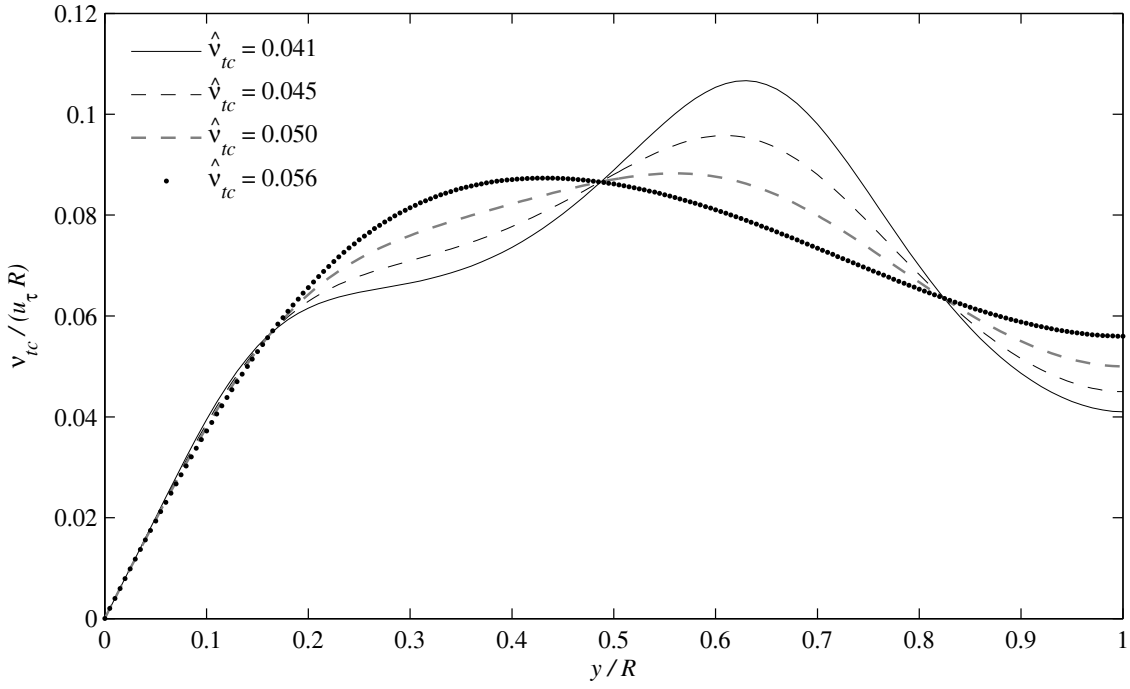

Fig. C.5 Turbulent eddy viscosity profiles when varying the centerline turbulent eddy viscosity.

This process of finding the optimum $\hat{\nu}_{tc}$ and C_7 coefficients, which give satisfactory results for the turbulent eddy viscosity was repeated over a range of A, B and e values . Because the coefficient B, which gave the best fitness was quite small, the coefficient B was set to 0.

APPENDIX D

COMPUTER CODE

I. Code to Solve the *k-λ* Model for Fully Rough Pipe Flow

```
program main
   use solver
   implicit none
   CHARACTER*(50):: filename
   integer :: case_num,ierror,i
   real :: case_fitness,case_fitness_Moody,fitness,weighting,fitness_uprofile,fitness_Moody
   real :: Roverks_(8)

   Roverks_ = (/ 15.,30.6,60.,126.,252.,507.,0.5/0.0003,0.5/0.000058 /)
   call set_solver_vals()

   call solver_allocate()
   call create_grid()


   write(*,*)Fully Rough Flow k-Lambda Model'

   write(*,*) 'Roverks (',Roverks,' ):'
   read(5,'(a)') rec
   if(rec .ne. ' ') read(rec,*) Roverks
   if( Roverks==1667) Roverks=0.5/0.0003
   if( Roverks==8621) Roverks=0.5/0.000058
   RoverksUser = Roverks

   weighting = 0.5
   kwall = 0.05
   A0 = 0.0025
   A10 = 0.012
   A11 = 0.05
   A12 = 0.09
   A13 = 0.39
   B00 = 0.0713244976729399
   B01 = 0.25
   B10 = 1.37
   B11 = -14.0
   B20 = -0.7
   B21 = 40.

!    write(*,*) 'Cnu (',Cnu ,' ):'
!    read(5,'(a)') rec
!    if(rec .ne. ' ') read(rec,*) Cnu

   write(*,*) 'sigmak (',sigmak ,' ):'
   read(5,'(a)') rec
   if(rec .ne. ' ') read(rec,*) sigmak

   write(*,*) 'kwall (',kwall ,' ):'
   read(5,'(a)') rec
```

```
      if(rec .ne. ' ') read(rec,*) kwall


!     write(*,*) 'A0 (,A0 ,' ):'
!     read(5,'(a)') rec
!     if(rec .ne. ' ') read(rec,*) A0
!
!     write(*,*) 'A10 (,A10 ,' ):'
!     read(5,'(a)') rec
!     if(rec .ne. ' ') read(rec,*) A10
!
!     write(*,*) 'A11 (,A11 ,' ):'
!     read(5,'(a)') rec
!     if(rec .ne. ' ') read(rec,*) A11

      write(*,*) 'B00 (,B00 ,' ):'
      read(5,'(a)') rec
      if(rec .ne. ' ') read(rec,*) B00

!     write(*,*) 'A13 (,A13 ,' ):'
!     read(5,'(a)') rec
!     if(rec .ne. ' ') read(rec,*) A13

!     write(*,*) 'B00 (,B00 ,' ):'
!     read(5,'(a)') rec
!     if(rec .ne. ' ') read(rec,*) B00

!     write(*,*) 'B01 (,B01 ,' ):'
!     read(5,'(a)') rec
!     if(rec .ne. ' ') read(rec,*) B01
!
!     write(*,*) 'B02 (,B02 ,' ):'
!     read(5,'(a)') rec
!     if(rec .ne. ' ') read(rec,*) B02

!     write(*,*) 'B03 (,B03 ,' ):'
!     read(5,'(a)') rec
!     if(rec .ne. ' ') read(rec,*) B03

!     write(*,*) 'B10 (,B10 ,' ):'
!     read(5,'(a)') rec
!     if(rec .ne. ' ') read(rec,*) B10
!
!     write(*,*) 'B11 (,B11 ,' ):'
!     read(5,'(a)') rec
!     if(rec .ne. ' ') read(rec,*) B11
!
!     write(*,*) 'B20 (,B20 ,' ):'
!     read(5,'(a)') rec
!     if(rec .ne. ' ') read(rec,*) B20
!
!     write(*,*) 'B21 (,B21 ,' ):'
!     read(5,'(a)') rec
!     if(rec .ne. ' ') read(rec,*) B21
```

```
   write(*,*) 'weighting (',weighting ,' ):'
   read(5,'(a)') rec
   if(rec .ne. ' ') read(rec,*) weighting


   call update_variable_vRoverks()


   case_num = 0
   write(*,*) 'Enter integer for case (',case_num,'):'
   read(5,'(a)') rec
   if(rec .ne. ' ') read(rec,*) case_num



   fitness_uprofile = case_fitness(case_num)  ! done at ksp = 80000

   if (weighting/=0.0) then
      fitness_Moody = case_fitness_Moody(case_num)  ! ksp varies
   endif

   fitness = weighting*fitness_Moody+(1.-weighting)*fitness_uprofile

   write(filename,*) case_num
   filename = 'fitness_'//trim(adjustl(filename))//'.txt'
   open(unit = 20, File = filename, status="replace", action = "write", iostat = ierror)
   write(20,*) fitness,' = case fitness'
   close(20)
   write(*,*)


   write(*,*)
   write(*,*) ' ----------------------------------------'
   write(*,*) ' Fitness u+ =', fitness_uprofile
   write(*,*) ' Fitness Moody =', fitness_Moody
   write(*,*) ' Fitness TOTAL =', fitness
   write(*,*) ' ----------------------------------------'
   write(*,*)

!    open(unit=15,file = 'fitness.txt')
!       write(15,'(20ES24.14)')Roverks,kwall,A0,A1,B0,B1,B2,weighting,fitness
!    close(15)

   if (RoverksUser<0) then
      open(unit=10,file = 'indiv_fitness.txt')
      do i=1,8
         write(10,*)    Roverks_(i),fitness_Moody_(i),fitness_uprofile_(i),weighting*fitness_Moody_(i)+(1.-
weighting)*fitness_uprofile_(i)
      enddo
      write(10,*)
      write(10,*)
RoverksUser,sum(fitness_Moody_)/8.,sum(fitness_uprofile_)/8.,weighting*sum(fitness_Moody_)/8.+(1.-
weighting)*sum(fitness_uprofile_)/8.
      close(10)
   else
```

```
      open(unit=10,file = 'indiv_fitness.txt')
         write(10,*) Roverks,fitness_Moody,fitness_uprofile,fitness
      close(10)
   endif

end program main



!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

real function case_fitness_Moody(case_num)
   use solver
   integer :: ngood,total,ierror,i,jj,case_num
   real :: results(7),local_Roverks,Roverks_(7)
   character*(50) :: filename
   results(:) = 0.0
   total = 0
   local_Roverks = Roverks

   !Roverks_ = (/ 15.,30.6,60.,126.,252.,507.,0.5/0.0003,0.5/0.000058 /)
   Roverks_ = (/ 15.,30.6,60.,126.,252.,507.,0.5/0.000058 /)


   100 FORMAT (1X, 1000ES22.14)
   write(filename,*) case_num
   filename = 'caseMoody_'//trim(adjustl(filename))//'.txt'
!    open(unit = 10, File = filename, status="replace", action = "write", iostat = ierror)
!    write(10,*) '  Roverks             Re_bulk             4CF                 Colebrook           '

   if(RoverksUser.gt.0.0) then
      Roverks = RoverksUser
      call run_case( Roverks,  ngood,results(1)); total = total + ngood
!       do i=1,ngood
!          write(10,100) Roverks,vRebulk(i),vCF(i),func_Colebrook(vRebulk(i))
!       end do
!       write(10,*)
      fitness_Moody_(1) = sqrt(results(1)/ngood)
      case_fitness_Moody = fitness_Moody_(1)
   else
      do jj=1,7
         Roverks = Roverks_(jj)
         call update_variable_vRoverks()
         call run_case( Roverks,  ngood,results(jj)); total = total + ngood
!          do i=1,ngood
!                                                                                                  write(10,100)
Roverks,vRebulk(i),vCF(i),func_Colebrook(vRebulk(i)),(func_Colebrook(vRebulk(i)) - vCF(i))**2
!          end do
!          write(10,*)
         fitness_Moody_(jj) = (results(jj)/ngood)
      enddo

      case_fitness_Moody = sum(fitness_Moody_)/7.
   end if

   !case_fitness_Moody =  sqrt(sum(results)/real(total))
```

```
!    write(10,*) '  kappa = ',kappa
!    write(10,*) '  gamma = ',gamma
!    write(10,*)
!    do jj=1,8
!       write(10,*) Roverks_(jj),results(jj)
!    enddo
!    write(10,*)
!    write(10,*) case_fitness_Moody, ' = case fitness'
!    close(10)

end function case_fitness_Moody


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

subroutine run_case(Roverks_val,ngood,error)
    use solver
    integer :: ngood
    real :: Roverks_val,error

   Roverks = Roverks_val
   call update_variable_vRoverks()


    call p_vksp(1,ngood,error)
    write(*,*) 'ngood = ',ngood
    write(*,*) 'fitness Moody chart = ',error
end subroutine run_case

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


real function case_fitness(case_num)
    use solver
    integer :: ngood,ierror,i,jj,case_num
    real,dimension(14) :: results,local_Roverks
    real :: Roverks_(8)
    character*(50) :: filename
    results(:) = 0.0
    Roverks_ = (/ 15.,30.6,60.,126.,252.,507.,0.5/0.0003,0.5/0.000058 /)

    100 FORMAT (1X, 1000ES22.14)
!    write(filename,*) case_num
!    filename = 'caseUprofile_'//trim(adjustl(filename))//'.txt'
!    open(unit = 10, File = filename, status="replace", action = "write", iostat = ierror)
!    write(10,*) '  R/ks               Fitness'

    if(RoverksUser.gt.0.0) then
       Roverks = RoverksUser
       call update_variable_vRoverks()
       call run_case_Velocity( Roverks,1,results(1))
!       write(10,100) Roverks,results(1)
!       write(10,*)
       total = 1
```

```
        fitness_uprofile_ = results(1)
    else
        do jj=1,8
            Roverks = Roverks_(jj)
            call update_variable_vRoverks()
            call run_case_Velocity( Roverks,1,results(jj))
!             write(10,100) Roverks,results(jj)
!             write(10,*)
            fitness_uprofile_(jj) = results(jj)
        enddo
        total = 8
    end if

    case_fitness = sum(results)/real(total)

    !write(*,*) 'fitness Velocity TOTAL = ',case_fitness

!    write(10,*) '        A0 = ',A0
!    write(10,*) '        A1 = ',A1
!    write(10,*) '        B0 = ',B0
!    write(10,*) '        B1 = ',B1
!    write(10,*) '        B2 = ',B2
!    write(10,*) '     kwall = ',kwall
!!    write(10,*) 'lambdaWall = ',lambdaWall
!    !write(10,*) 'lambdaWallSlope = ',lambdaWallSlope
!!    write(10,*) 'R/ks used =',local_Roverks(1:nb_of_sets)
!    write(10,*)
!    write(10,*) case_fitness, ' = case fitness (Nikuradse u+ exp data)'
!    close(10)


end function case_fitness

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

subroutine run_case_Velocity(Roverks_val,case_number,error)
    use solver
    integer, intent(in) :: case_number ! can only be 1 through 6
    real :: Roverks_val,error

    call update_variable_vRoverks()

    call p_getu(case_number,kshat,error)

end subroutine run_case_Velocity

!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
subroutine set_solver_vals()
    use solver
    real :: RC_ltxt(50)
    real :: a,b,c,d,e,f

      Roverks = 15.0
        kr = 0.5/Roverks
```

```
      Cnu = 0.006
     Clam = Cnu**2
   sigmak = 2.0
      ksp = 80000.0

    kappa = 0.403
    gamma = 0.0324
     beta = 1.000002
        n = 3201
    nplot = 51

    nuhat = 2.*kr/ksp
     Rtau = 1./nuhat
     kshat = 2.*kr


     a = 0.00233
     b = 0.659
     c = 0.0307
     d = 0.00551
     e = 0.209
     f = 0.0421



end subroutine set_solver_vals
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!


module solver
   IMPLICIT NONE
   integer :: n,nb_of_sets,nb_pts_per_set,nplot,nstat
   integer :: FullyRough,FullySmooth

   real :: Rtau,kr,ksp,beta,nuhat,kshat,Rshat,Roverks,qkguess
   real :: kappa,RoverksUser,fitness_Moody_(7),fitness_uprofile_(8)
   real :: Cnu, sigmak,Clam
   real :: kwall,lambdaWallSlope,lambdaWall,lambdaCenter
   real :: Re,Df,um,km
   real :: gamma
   real                                                                    ::
A0,A1,B0,B1,B2,A10,A11,A12,A13,B00,B01,B02,B03,B10,B11,B12,B20,B21,B22,A1_exp,A1_cst

   CHARACTER*(80):: rec,init,file_r
   CHARACTER(LEN=100)::fn

   real, allocatable, dimension(:)::r_hat,yplus
   real, allocatable, dimension(:)::k,kprime
   real, allocatable, dimension(:)::qk,qkprime
   real, allocatable, dimension(:)::u,uprime
   real, allocatable, dimension(:)::nu
   real, allocatable, dimension(:)::h,lambda

   real, allocatable, dimension(:)::vCF,vRebulk
```

```
contains

subroutine solver_allocate()
   !Allocate Memory
   ALLOCATE(r_hat(n)); ALLOCATE(yplus(n))
   ALLOCATE(k(n));     ALLOCATE(kprime(n));
   ALLOCATE(qk(n));     ALLOCATE(qkprime(n));
   ALLOCATE(u(n));     ALLOCATE(uprime(n));
   ALLOCATE(nu(n));
   ALLOCATE(h(n)); ALLOCATE(lambda(n));
   ALLOCATE(vRebulk(nplot)); ALLOCATE(vCf(nplot))
end subroutine solver_allocate

subroutine solver_deallocate()
   !Deallocate Memory
   DEALLOCATE(r_hat); DEALLOCATE(yplus);
   DEALLOCATE(k);     DEALLOCATE(kprime);
   DEALLOCATE(qk);     DEALLOCATE(qkprime);
   DEALLOCATE(u);     DEALLOCATE(uprime);
   DEALLOCATE(nu);
   DEALLOCATE(h); DEALLOCATE(lambda);
   DEALLOCATE(vRebulk);DEALLOCATE(vCf);
end subroutine solver_deallocate


subroutine create_grid()
   integer :: j
   real :: eta,cbeta,dzeta

   if (beta>0) then
      dzeta = 1.0/real(n-1)
      !Create Grid
      do j=1,n,1
         eta = real(n-j)/real(n-1)
         cbeta = ((beta+1.0)/(beta-1.0))**(1.0-eta)
         if(beta .eq. 0.0) then
            r_hat(j) = 1.0 - real(j-1)*dzeta
         else
            r_hat(j) = 1.0 - (beta+1.0 - (beta-1.0)*cbeta)/(1.0 + cbeta)
         end if
      end do
      r_hat(1) = 0.0
   else ! uniform grid, same spacing
      do j=1,n
         r_hat(j) = real(j-1) / real(n-1)
      enddo
   endif

end subroutine create_grid

subroutine update_variable_vRoverks()
   real :: C_0,C_1,C_2,C_3,C_4,C_5
   real :: sigma_k, k_wall
```

```
   real                                                                                    ::
A_lam0,A_lam10,A_lam11,A_lam12,A_lam13,B_lam00,B_lam01,B_lam02,B_lam03,B_lam10,B_lam11,
B_lam20,B_lam21


   kshat = 1./Roverks
   kr = kshat/2.
   nuhat = 2.*kr/ksp
   Rtau = 1./nuhat
   Rshat = 1./(gamma*kshat)
   sigma_k = sigmak
   k_wall = kwall

      Cnu    =   1.1430803817E-04*sigma_k**4 - 2.1568032682E-03*sigma_k**3                &
             + 1.5454411906E-02*sigma_k**2 - 5.3147683089E-02*sigma_k + 8.3648609370E-02      !
FINAL NO CHANGE FROM V8.1
!
      A_lam0  = - 5.4806699600E-06*sigma_k**4 + 1.0829889892E-04*sigma_k**3                &
             - 5.8824903497E-04*sigma_k**2 + 6.4266598213E-05*sigma_k + 7.0559295507E-03      !
FINAL NO CHANGE FROM V8.1
!
      A_lam10 =   6.0548405176E-06*sigma_k**4 - 1.7464041639E-04*sigma_k**3                &
             + 1.7083135016E-03*sigma_k**2 - 7.8015697237E-03*sigma_k + 2.3620283331E-02      !
FINAL NO CHANGE FROM V8.1
!
      A_lam11 =   1.6430210539E-04*sigma_k**4 - 3.2876048062E-03*sigma_k**3                &
             + 2.3668508652E-02*sigma_k**2 - 7.4817705811E-02*sigma_k + 1.3742572373E-01      !
FINAL NO CHANGE FROM V8.1
!
        C_4 = - 3.6249854255E-04*k_wall**2  - 2.3771783095E-04*k_wall  + 3.1811391789E-04      !
<<<<<<<<<<
        C_3 =   6.7456577268E-03*k_wall**2  + 3.5031152723E-03*k_wall  - 7.6039769524E-03      !
<<<<<<<<<<
        C_2 = - 4.3056163634E-02*k_wall**2  - 2.0050955122E-02*k_wall  + 5.7370834148E-02      !
<<<<<<<<<<
        C_1 =   1.1098738361E-01*k_wall**2  + 5.7323021287E-02*k_wall  - 1.5770938564E-01      !
<<<<<<<<<<
        C_0 = - 9.7888362645E-02*k_wall**2  - 7.6772775598E-02*k_wall  + 1.6397877418E-01      !
<<<<<<<<<<
      A_lam12 =     C_4*sigma_k**4 + C_3*sigma_k**3 + C_2*sigma_k**2 + C_1*sigma_k + C_0
! FINAL CHANGED FROM V8.1 <<<<<<<<<<
!
      A_lam13 =   2.1775071643E-04*sigma_k**4 - 4.6125648020E-03*sigma_k**3                &
             + 3.2004745006E-02*sigma_k**2 - 7.6059412219E-02*sigma_k + 2.6786227707E-01      !
FINAL NO CHANGE FROM V8.1
!
      !B_lam00 =   1.5571555961E-03*sigma_k**4 - 3.7455015562E-02*sigma_k**3                &
      !         + 3.0085113473E-01*sigma_k**2 - 9.7875053089E-01*sigma_k + 1.1002011812E+00      !
V 2.5 NO CHANGE FROM V8.1
B_lam00 = B00
!
      B_lam01 = - 1.7611480539E-03*sigma_k**4 + 3.3472445299E-02*sigma_k**3                &
             - 2.1562199190E-01*sigma_k**2 + 4.9483981773E-01*sigma_k + 4.4081967221E-01      !
FINAL NO CHANGE FROM V8.1
!
        C_4 =   4.5958241000E-03*k_wall**2  + 2.0834333483E-03*k_wall  - 1.3277370846E-02
```

```
        C_3 = - 9.3116503886E-02*k_wall**2  - 2.8569790820E-02*k_wall  + 2.7587164438E-01
        C_2 =   6.0395589241E-01*k_wall**2  + 1.8761243453E-01*k_wall  - 2.0079111202E+00
        C_1 = - 1.4946016082E+00*k_wall**2  - 7.3780887177E-01*k_wall  + 6.0340918034E+00
        C_0 =   1.2142113765E+00*k_wall**2  + 1.2696368571E+00*k_wall  - 7.2873561012E+00
      B_lam02 =     C_4*sigma_k**4  +  C_3*sigma_k**3  +  C_2*sigma_k**2  +  C_1*sigma_k  +  C_0
! FINAL NO CHANGE FROM V8.1
!
      B_lam03 =   6.3453858956E-04*sigma_k**4 - 1.7156286122E-02*sigma_k**3                    &
           + 1.4904850864E-01*sigma_k**2 - 5.3187100685E-01*sigma_k + 7.9782349433E-01         !
FINAL NO CHANGE FROM V8.1
!
      B_lam10 = - 1.0134626360E-03*sigma_k**4 + 1.9147367622E-02*sigma_k**3                    &
           - 1.2222170912E-01*sigma_k**2 + 2.5530412815E-01*sigma_k + 1.4355452292E+00         !
FINAL NO CHANGE FROM V8.1
!
      B_lam11 =                              - 6.6206887937E-03*sigma_k - 1.3830728894E+01       ! V 2   NO
CHANGE FROM V8.1
        C_4 = - 2.0201876364E-03*k_wall**2  + 4.3279392861E-03*k_wall  - 1.3949382336E-03
        C_3 =   2.6570830671E-02*k_wall**2  - 6.4136620678E-02*k_wall  + 2.3621757343E-02
        C_2 = - 1.1989031734E-01*k_wall**2  + 3.3373664338E-01*k_wall  - 1.3864710895E-01
        C_1 =   2.1942375688E-01*k_wall**2  - 7.0814005373E-01*k_wall  + 3.0315634574E-01
        C_0 = - 1.4066477454E-01*k_wall**2  + 4.5837394149E-01*k_wall  - 8.0853545298E-01
      B_lam20 =     C_4*sigma_k**4  +  C_3*sigma_k**3  +  C_2*sigma_k**2  +  C_1*sigma_k  +  C_0
! FINAL NO CHANGE FROM V8.1
!
      B_lam21 =                              - 2.6815510578E-03*sigma_k + 4.0037296283E+01       ! V 2   NO
CHANGE FROM V8.1


    A0 = A_lam0
    A1     =      A_lam11   +   A_lam12*kshat   +   (A_lam10   -   A_lam11   -   A_lam12*kshat)*exp(-
(kshat**A_lam13))
    B0  =   B_lam01 + B_lam02*kshat + (B_lam00 - B_lam01 - B_lam02*kshat)*exp(-(kshat**B_lam03))
    B1  =   B_lam10 + B_lam11*kshat
    B2  =   B_lam20 + B_lam21*kshat


end subroutine



!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!       Compare      velocity      with      Nikuradse      exp      data
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
subroutine p_getu(case_number,ksoverR_in,error)
   integer :: iprint, niter,icompare,i,j
   integer, intent(in) :: case_number
   real :: error,junk,pRMS,RMS
   real, intent(in) :: ksoverR_in
   real :: uplus_reference(12),uhat_reference(12),r_hat_discretized(12),u_fromEmpLambda(12)
   real :: a,b,u1,u2,r1,r2


   do i = 1,n
      lambda(i) = lambdahat(r_hat(i))
   enddo
```

```
   call integrate_k_fromWall()

   write(*,*)
   write(*,*) '  Re (um) = ' , Re
   write(*,*) ' 4Cf      = ' , Df
   write(*,*) 'Colebrook = ', func_Colebrook(Re)
   write(*,*)
   write(*,*) '        B0 = ',B0
   write(*,*) '        B1 = ',B1
   write(*,*) '        B2 = ',B2
   write(*,*) '        A0 = ',A0
   write(*,*) '        A1 = ',A1
   write(*,*) 'lambdaWall = ', lambdaWall
   write(*,*) '     kwall = ',kwall
   write(*,*)


!    open(unit=10,file='data.txt')
!       do i=n,1,-1
!          write(10,*) r_hat(i),1.-r_hat(i),u(i),k(i),lambda(i),nu(i), r_hat(i)/( nuhat+nu(i) )
!       enddo
!    close(10)


    ! Compare u+ ( u+ from empirical nut vs. u+ from RK4 with empirical lambda_hat)
    r_hat_discretized = (/ 0.0,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.95 /)


   j=1
   u_fromEmpLambda(1) = u(1)
   call get_uplus_uhat(0.0,uplus_reference(1),junk)


   do icompare=2,12
      do while ((j<=n).and. (r_hat(j)-r_hat_discretized(icompare)<=0.)  )
         j=j+1
      enddo
      j=j-1

      if        (r_hat_discretized(icompare)-r_hat(j)>         r_hat(j+1)-r_hat_discretized(icompare))         then
!r_hat_discretized(icompare) closer to r_hat(j+1)
         call get_uplus_uhat(r_hat(j+1),uplus_reference(icompare),junk)
         u_fromEmpLambda(icompare) = u(j+1)
         pRMS                                    =                                    abs((uplus_reference(icompare)-
u_fromEmpLambda(icompare))/uplus_reference(icompare))
      else
         call get_uplus_uhat(r_hat(j),uplus_reference(icompare),junk)
         u_fromEmpLambda(icompare) = u(j)
         pRMS                                    =                                    abs((uplus_reference(icompare)-
u_fromEmpLambda(icompare))/uplus_reference(icompare))
      endif

   enddo


   error = sum( abs(uplus_reference-u_fromEmpLambda)/uplus_reference)/12.
```

```
end subroutine p_getu



! ---------------------------------------------------------------------------------------
! uplus/uhat when nut = Eq. 106
! ---------------------------------------------------------------------------------------

subroutine get_uplus_uhat(r_hat,u_plus,u_hat)
   real :: yhat,r_hat
   real :: delta,deltap
   real :: u_plus,u_hat
   real :: D0,D1,D2,D3,D4,D5,C0,C1,C2,C3,C4,C5,C6,C7,Acoeff,Bcoeff,ecoeff,nutc

      Acoeff = 1.2
      Bcoeff = 0.0
      ecoeff = 1.4
      nutc = 0.053
      C7 = -1.3

      D0 = (Rshat+1.)**2*(log(Rshat+1.)-3./2.)+2.*Rshat+3./2.
      C0 = Acoeff*kshat**ecoeff+Bcoeff
      C1 = Rshat**3./((Rshat+1.)*D0)
      C2 = Rshat**2.*(Rshat**2/((Rshat+1.)**2)-kappa/nutc)/(2.*D0)
      C3 = Rshat**5./(3.*(Rshat+1.)**3*D0)
      D1 = -C0-C1-C2-C3-C7
      D2 = -C1-2.*C2-3.*C3-7.*C7
      D3 = -C0/2.-C1/3.-C2/4.-C3/5.-C7/9.
      C4 = -39.*D1+3.*D2+168.*D3
      C5 = 84.*D1-7.*D2-336.*D3
      C6 = -44.*D1+4.*D2+168.*D3


      yhat = 1.-r_hat

      delta = C0 + C1*r_hat + C2*r_hat**2 + C3*r_hat**3 + C4*r_hat**4 + C5*r_hat**5 + C6*r_hat**6
+ C7*r_hat**7
      deltap = -(C1 + 2.*C2*r_hat + 3.*C3*r_hat**2 + 4.*C4*r_hat**3 + 5.*C5*r_hat**4 +
6.*C6*r_hat**5 + 7.*C7*r_hat**6)

      u_hat = Rshat**2*log(Rshat*yhat+1.)/D0+delta
      u_plus = log(Rshat*yhat+1.)/kappa+D0*delta/(kappa*Rshat**2)

end subroutine



! ---------------------------------------------------------------------------------------
! Lambda_hat Empirical Function
! ---------------------------------------------------------------------------------------

function lambdahat(r_hat)
   real :: r_hat,lambdahat,yhat

   yhat = 1.-r_hat
```

```
   lambdahat = (A0*kshat+A1*yhat)*(1.-.5*yhat)*(B0+B1*r_hat**2+B2*r_hat**4+(1.-B0-B1-
B2)*r_hat**6)

end function


! ------------------------------------------------------------------------------------
!   INTEGRATE from WALL
! ------------------------------------------------------------------------------------

subroutine integrate_k_fromWall()
   integer :: i,niter,nitermax
   real :: t0,y0(5),dt,y(5),qwall(3),qcenter(3)

   ! integrates from the wall, shoots to the centerline to have Q+ = 0 at the centerline
   !kwall = 6120.1*kshat**3-1503.6*kshat**2+144.43*kshat-0.0309

   !write(*,*) 'kwall = ', kwall

   niter = 1
   nitermax = 200
   nstat = 0
   qkguess = 2.0

   !write(*,*) 'Iterate on Q...'


   do while ((qkguess>=-1.).and.(nstat==0) )
   niter = 1

      ! GUESS 1 ON Q
      ! ---------------
      t0 = 0.
      k(n) = kwall
      qk(n) = qkguess   ! initial guess for q(wall)
      qwall(1) = qk(n)
      y0(1) = k(n)
      y0(2) = qk(n)
      !write(*,*) i,k(n),qk(n)

      do i = n-1,1,-1
         dt = r_hat(i+1)-r_hat(i)  ! dt is equivalent of dy, not dr_hat
         call RK4_fromWall(2,t0,y0,dt,y)
         t0 = t0+dt
         y0 = y
         k(i) = y(1)
         qk(i) = y(2)
      enddo
      qcenter(1) = qk(1)

      !write(*,*) niter, qk(1)


      ! GUESS 2 ON Q
      ! ---------------
      niter = niter + 1
```

```
t0 = 0.
k(n) = kwall
if (qkguess<0) then
   qk(n) = qkguess*.95   ! initial guess for q(wall)
else
   qk(n) = qkguess*1.05
endif
qwall(2) = qk(n)
y0(1) = k(n)
y0(2) = qk(n)

do i = n-1,1,-1
   dt = r_hat(i+1)-r_hat(i)
   call RK4_fromWall(2,t0,y0,dt,y)
   t0 = t0+dt
   y0 = y
   k(i) = y(1)
   qk(i) = y(2)
enddo
qcenter(2) = qk(1)
!write(*,*) niter, qk(1)


! ITERATE ON Q (secant method)
! ----------------------------
do while ((abs(qk(1))>1.0e-12).and.(niter<nitermax))
   niter = niter + 1
   t0 = 0.
   k(n) = kwall
   qk(n) = qwall(2)-qcenter(2)*( qwall(2)-qwall(1) ) / ( qcenter(2)-qcenter(1) )
   qwall(3) = qk(n)
   y0(1) = k(n)
   y0(2) = qk(n)

   do i = n-1,1,-1
      dt = r_hat(i+1)-r_hat(i)
      call RK4_fromWall(2,t0,y0,dt,y)
      t0 = t0+dt
      y0 = y
      k(i) = y(1)
      qk(i) = y(2)
   enddo
   qcenter(3) = qk(1)
   !write(*,*) niter, qk(1),qk(n)

   qwall(1) = qwall(2)
   qwall(2) = qwall(3)
   qcenter(1) = qcenter(2)
   qcenter(2) = qcenter(3)

enddo


niter = niter + 1
t0 = 0.
k(n) = kwall
```

```
      qk(n) = qwall(2)-qcenter(2)*( qwall(2)-qwall(1) ) / ( qcenter(2)-qcenter(1) )
      qwall(3) = qk(n)
      y0 = 0.
      y0(1) = k(n)
      y0(2) = qk(n)
      u(n) = 0.

      do i = n-1,1,-1
         dt = r_hat(i+1)-r_hat(i)
         call RK4_fromWall(5,t0,y0,dt,y)
         t0 = t0+dt
         y0 = y
         k(i) = y(1)
         qk(i) = y(2)
         u(i) = y(3)
      enddo
      qcenter(3) = qk(1)
      km = y(5)
      um = y(4)
      !write(*,*) niter, Roverks,qk(1),qwall(3)
      nu = lambda*sqrt(k)
      h = Cnu*k /nuhat
      Df =  8./um**2.
      Re = 2.*um/nuhat

      if (isnan(Re)) then
         qkguess = qkguess-0.1
         nstat = 0
      else
         nstat =1
      endif

   enddo


   if ((niter>=nitermax).or.(isnan(Re))) then
      nstat = 0
   else
      nstat = 1
      write(*,*) niter, Roverks,qk(1),qwall(3)
   endif


end subroutine




subroutine RK4_fromWall(nn,t0,y0,dt,y)
   ! solve a system of n first order differential equations
   ! dy(i)/dt = f(i,t,y)
   ! calls the function f

   integer, intent(in) :: nn ! number of equation to be solved
   real, dimension(nn,4) :: k_ ! table of coefficient (k1, k2, k3 k4); 4 coeffs per equation
   real, intent(inout) :: t0, y0(nn),dt
```

```
    real :: coeff(4),Rtau
    real, intent(out) :: y(nn)
    integer :: i,j

    if (t0+dt>1.) dt=1.-t0
    coeff = (/ 1./6., 1./3., 1./3., 1./6. /)
    do j=1,nn
        k_(j,1)=f_fromWall(j,t0,y0)*dt
        y(j)=y0(j)+k_(j,1)/2.
    end do
    do j=1,nn
        k_(j,2)=f_fromWall(j,t0+dt/2.,y)*dt
    end do
    do j=1,nn
       y(j)=y0(j)+k_(j,2)/2.
    end do
    do j=1,nn
       k_(j,3)=f_fromWall(j,t0+dt/2.,y)*dt
    end do
    do j=1,nn
       y(j)=y0(j)+k_(j,3)
    end do
    do j=1,nn
       k_(j,4)=f_fromWall(j,t0+dt,y)*dt
    end do
    do j=1,nn
       y(j)=y0(j)
       do i=1,4
          y(j)=y(j)+coeff(i)*k_(j,i)
       end do
    end do

end subroutine


function f_fromWall(i,zeta,y)
    integer, intent(in) :: i
    real, intent(in) :: zeta,y(*)   ! y(1) = k, y(2) = q
    real :: f_fromWall,r_hat
    ! input zeta = 0 at wall, 1 at centerline
    !if (zeta<0.) zeta=0.

    r_hat = 1. - zeta ! r_hat = 0 at centerline, 1 at wall
    if (i==1) then  ! dk/dr_hat
       if (r_hat <= 10**(-16./6.)) then   !(r_hat <= 10**(-16./6.)) then  ! at the centerline, exception /0
          f_fromWall = 0.
       else
          f_fromWall = -y(2)/r_hat*3./( nuhat+5.*(lambdahat(r_hat)*sqrt( abs(y(1)) ) )/sigmak) ! corresponds
to dk/dr_hat
          !write(*,*) nut(r_hat),y(2)
       endif
    elseif (i==2) then   ! dq/dr_hat
       f_fromWall                =                                (lambdahat(r_hat)*sqrt(abs(y(1))))*r_hat**3./(nuhat+
(lambdahat(r_hat)*sqrt(abs(y(1))))                                )**2.                                -
Cnu**2.*nuhat/(lambdahat(r_hat)*sqrt(abs(y(1))))**2.*y(1)**2.*r_hat
       !if  ((nut(r_hat)==0.).and.(r_hat==0) )  f_fromWall=0.
```

```
   elseif (i==3) then   ! du/dr_hat
      f_fromWall = -r_hat/( nuhat+(lambdahat(r_hat)*sqrt(y(1))) )
   elseif (i==4) then   ! dum/dr_hat
      f_fromWall = -y(3)*r_hat*2.
   elseif (i==5) then   ! dkm/dr_hat
      f_fromWall = -y(1)*r_hat*2.
   end if
   f_fromWall = -f_fromWall  ! gives the variables in terms of d_/dy
   !if (f_fromWall=='NaN') pause
end function



subroutine p_vksp(iprint,ngood,error)
   integer :: iprint,ngood,niter
   real :: error,power

   power = 2.0
   ksp = 10**power
   Re = 0.0
   ngood = 1
   error = 0.0
   Rshat = Roverks/(gamma)

   if(iprint.eq.1) write(*,*) '     ngood   R/ks                ksp                 Re                 4CF        &
                        &          4Cf Reference          error                ' !num_iter'
   do while (((Re<1.0e8).and.(ngood<=nplot)).or.(ngood<5))
      nuhat = 2.*kr/ksp
      Rtau = 1./nuhat

      call integrate_k_fromWall()
      vRebulk(ngood) = Re
      vCF(ngood) = Df

      if((Re<1.0e8).or.(ngood<4))  error  =  error  +  abs(reference_4Cf(Re,Roverks)  -
vCF(ngood))/reference_4Cf(Re,Roverks)
!                       if((Re<1.0e8).or.(ngood<4))  error  =  error  +  ((reference_4Cf(Re,Roverks)  -
vCF(ngood))/func_Colebrook(Re))**2 ! % RMS
      if(iprint.eq.1) write(*,*) ngood,Roverks,ksp,Re,Df,reference_4Cf(Re,Roverks),error !,niter
      ngood = ngood + 1
      power = power + 0.25
      ksp = 10.0**power
   end do
   ngood = ngood - 2
end subroutine p_vksp

real function func_Colebrook(Re)
   implicit none
   real :: Re,old,new
   if(Re<100.0) then
      func_Colebrook = 64.0/Re
   else
      new = (-1.8*log10((kr/3.7)**(1.11) + 6.9/Re))**(-2.0)
      old = 10.0
      do while(abs((new-old)/new) .gt. 1.0e-5)
         old = new
```

```
            new = (-2.0*log10(kr/3.7 + 2.51/(Re*sqrt(old))))**(-2.0)
        end do
        func_Colebrook = new
    end if
return; end function func_Colebrook


real function reference_4Cf(Re,Roverks)
    real :: Re,Roverks
    integer :: i,indix

    reference_4Cf = (2.0*log10(Roverks)+1.74)**(-2.0)

end function

end module solver
```

II. Optimization Code

```
program main
    use bfgs
    implicit none
    character*(50) :: rec,fn
    integer :: i,ierror

    !Defaults
    iter = 0
    default_alpha = 1.0e-8
    diff_delta = 1.0e-8
    diff_scheme = 1
    stop_delta = 1.0e-12
    nsearch = 8

    fn = 'none'
!    write(*,*) 'Enter filename (none=use interactive console instead of file) (',fn,' ) :'
!    read(5,'(a)') rec
!    if(rec .ne. ' ') read(rec,*) fn
    fn = 'bfgs_start.txt'

    if(fn.ne.'none') then
        open(unit = 10, File = fn, action = "read", iostat = ierror)
        read(10,*) nvars
        read(10,*) !names of variables

        call opt_allocate()

        read(10,*) vars(1:nvars)
        read(10,*) opton(1:nvars)
        read(10,*) default_alpha
        read(10,*) diff_delta
        read(10,*) diff_scheme
        read(10,*) stop_delta
```

```
      read(10,*) nsearch
      close(10)
   else
      write(*,*) 'Enter number of variables :'
      read(5,'(a)') rec
      if(rec .ne. ' ') read(rec,*) nvars

      call opt_allocate()

      do i=1,nvars
         write(*,*) 'Enter variable ',i,' :'
         read(5,'(a)') rec
         if(rec .ne. ' ') read(rec,*) vars(i)
         write(*,*) 'Optimize this variable? (1=yes,0=no) (',opton(i),') :'
         read(5,'(a)') rec
         if(rec .ne. ' ') read(rec,*) opton(i)
      end do

      write(*,*) 'Enter default line search alpha (',default_alpha,') :'
      read(5,'(a)') rec
      if(rec .ne. ' ') read(rec,*) default_alpha

      write(*,*) 'Enter delta step size used for gradient calculations (',diff_delta,') :'
      read(5,'(a)') rec
      if(rec .ne. ' ') read(rec,*) diff_delta

      write(*,*) 'Enter differencing scheme (1=central diff, 0=forward diff) (',diff_scheme,') :'
      read(5,'(a)') rec
      if(rec .ne. ' ') read(rec,*) diff_scheme

      write(*,*) 'Enter stop delta (',stop_delta,') :'
      read(5,'(a)') rec
      if(rec .ne. ' ') read(rec,*) stop_delta

      write(*,*) 'Enter number of simultaneous cases in the line search (',nsearch,') :'
      read(5,'(a)') rec
      if(rec .ne. ' ') read(rec,*) nsearch
   end if

   fn = 'bfgs_start.txt'; call write_bfgs_file(fn)
   call opt_run()
   fn = 'bfgs_end.txt'; call write_bfgs_file(fn)

   call opt_deallocate()
end program main

module bfgs
   implicit none
   integer :: nvars
   integer :: diff_scheme !1 = central difference. 0 = forward difference
   integer :: iter
   integer :: nsearch

   real :: default_alpha
   real :: diff_delta
   real :: fitness_curr
```

```
    real :: stop_delta
    real :: alpha

    integer,allocatable :: opton(:)
    real,allocatable :: grad(:)
    real,allocatable :: vars(:)
    real,allocatable :: s(:)

contains

!-----------------------------------------------------------------------------------------------------------------------
-

subroutine opt_allocate()
    allocate(opton(nvars))
    allocate(grad(nvars))
    allocate(vars(nvars))
    allocate(s(nvars))
    opton = 0; vars = 0.0; grad = 0.0; s = 0.0
end subroutine opt_allocate

!-----------------------------------------------------------------------------------------------------------------------
-

subroutine opt_deallocate()
    deallocate(opton)
    deallocate(grad)
    deallocate(vars)
    deallocate(s)
end subroutine opt_deallocate

!-----------------------------------------------------------------------------------------------------------------------
-

subroutine opt_run()
    integer :: i_iter,o_iter,i,ierror
    real :: vars_orig(nvars),vars_old(nvars),grad_old(nvars)
    real :: dx(nvars,1),NG(nvars,1),N(nvars,nvars),gamma(nvars,1)
    real :: mag_dx,denom
    character(LEN=50)::fn,command
    110 format (1X, I10, 100ES22.13)

    fn = 'optimization.txt'
    open(unit = 1001, File = fn, action = "write", iostat = ierror)
    write(1001,*) 'iter o_it i_it   R/ks              sigmak              kwall              &
                          &coef              &
                          &weighting          &
                          &Fitness            alpha              mag(dx)'
    close(1001)
    open(unit = 1001, File = 'gradient.txt', action = "write", iostat = ierror)
    write(1001,*) 'iter o_it i_it   R/ks              sigmak              kwall              &
                          &coef              &
                          &weighting          &
                          &Fitness            alpha              mag(dx)'
    close(1001)
```

```
   command = 'rm input_* output_* case_* fitness_*'
   call system(command)

   write(*,*) 'Beginning Optimization Routine'
   write(*,*) 'Optimization Variables: '
   do i=1,nvars
      write(*,110) opton(i),vars(i)
   end do
   write(*,*) '      default alpha : ',default_alpha
   write(*,*) 'differenceing delta : ',diff_delta
   write(*,*) 'differencing scheme : ',diff_scheme
   write(*,*) '     stopping delta : ',stop_delta
   write(*,*) 'simultaneous search : ',nsearch

   o_iter = 0
   mag_dx = 1.0
   do while(mag_dx > stop_delta)
      vars_orig = vars
      i_iter = 0

      do while(mag_dx > stop_delta)
         call gradient()
         call append_file(fn,o_iter,i_iter,mag_dx)

         if(i_iter .eq. 0) then !set N=identity
            N = 0.0
            do i=1,nvars
               N(i,i) = 1.0 !N = identity matrix
            end do
         else
            dx(:,1) = vars(:) - vars_old(:)
            gamma(:,1) = grad(:) - grad_old(:)
            NG(:,1) = matmul(N,gamma(:,1))
            denom = dot_product(dx(:,1),gamma(:,1))
      !           N = N + matmul(dx-NG,transpose(dx-NG))/dot_product(dx(:,1)-NG(:,1),gamma(:,1)) !Rank
One Hessian Inverse Update
            N = N + (1.0+dot_product(gamma(:,1),NG(:,1))/denom)*(matmul(dx,transpose(dx))/denom) &
!BFGS Update
               & - ( matmul(dx,matmul(transpose(gamma),N)) + matmul(NG,transpose(dx)))/denom
         end if
         s(:) = -matmul(N,grad)
         vars_old = vars
         grad_old = grad

         call line_search()

         dx(:,1) = vars(:) - vars_old(:)
         mag_dx = sqrt(dot_product(dx(:,1),dx(:,1)))
         i_iter = i_iter + 1
         iter = iter + 1
      end do

      call append_file(fn,o_iter,i_iter,mag_dx)

      dx(:,1) = vars(:) - vars_orig(:)
      mag_dx = sqrt(dot_product(dx(:,1),dx(:,1)))
```

```
      o_iter = o_iter + 1
   end do

   call sleep(1)
   fitness_curr = case_fitness_single(0)
   call append_file(fn,o_iter,i_iter,mag_dx)
end subroutine opt_run

!-----------------------------------------------------------------------------------------------------------------------
-

subroutine write_bfgs_file(fn)
   character(50) :: fn
   integer :: ierror
   110 format (100ES22.13)
   120 format (100I22)
   fn = trim(adjustl(fn))
   write(*,*) 'writing ',fn
   open(unit = 200, File = fn, status = "replace", action = "write", iostat = ierror)
   write(200,*) nvars,'       num vars'
   write(200,*) '   R/ks                sigmak              kwall               &
                           &coef               &
                           &weighting           '

   write(200,110) vars(:)
   write(200,120) opton(:)
   write(200,*) default_alpha, '   default line search alpha'
   write(200,*) diff_delta, '   delta step size used for gradient calculations'
   write(200,*) diff_scheme,'   differencing scheme (1=central diff, 0=forward diff)'
   write(200,*) stop_delta,'   stop delta'
   write(200,*) nsearch,'     number of simultaneous cases in the line search'
   close(200)
end subroutine write_bfgs_file

!-----------------------------------------------------------------------------------------------------------------------
-

subroutine append_file(fn,o_iter,i_iter,mag_dx)
   character(50) :: fn
   real :: mag_dx
   integer :: o_iter,i_iter,ierror
   110 format (3I5, 100ES22.13)
   write(* ,110) iter,o_iter,i_iter,vars(:),fitness_curr,alpha,mag_dx
   open(unit = 1001, File = fn, status = "OLD", access = "append", iostat = ierror)
   write(1001,110) iter,o_iter,i_iter,vars(:),fitness_curr,alpha,mag_dx
   close(1001)
   open(unit = 1001, File = 'gradient.txt', status = "OLD", access = "append", iostat = ierror)
   write(1001,110) iter,o_iter,i_iter,grad(:),fitness_curr,alpha,mag_dx
   close(1001)
end subroutine append_file

!-----------------------------------------------------------------------------------------------------------------------
-

real function case_fitness(case_num)
   integer :: case_num,ierror
```

```
    character(50)::filename

    write(filename,*) case_num
    filename = 'fitness_'//trim(adjustl(filename))//'.txt'
    open(unit = 10, File = filename, action = "read", iostat = ierror)
    read(10,*) case_fitness
    close(10)
end function case_fitness

!-----------------------------------------------------------------------------------------------------------------------------
-

subroutine forward_diff()
    integer :: i
    character(50) :: command
    real :: vars_orig(nvars)
    vars_orig(:) = vars(:)

    grad = 0.0
    call start_case(0)
    do i=1,nvars
       if(opton(i).eq.1) then
          vars(i) = vars(i) + diff_delta
          call start_case(i)
          vars(:) = vars_orig(:)
       end if
    end do

    do while(.not.all_done())
       call sleep(1)
    end do
    call sleep(1)   !one more time to ensure all files are totally written

    fitness_curr = case_fitness(0)
    do i=1,nvars
       if(opton(i).eq.1) grad(i) = (case_fitness(i) - fitness_curr)/diff_delta
    end do

    call sleep(1)
    command = 'mv case_0.txt curr_case.txt'
    call system(command)
    command = 'rm input_* output_* case_* fitness_*'
    call system(command)

end subroutine forward_diff

!-----------------------------------------------------------------------------------------------------------------------------
-

subroutine gradient()
    real :: temp(nvars)
    call forward_diff()
    if(diff_scheme.eq.1) then
       temp(:) = grad(:)
       diff_delta = -diff_delta
       call forward_diff()
```

```
      grad(:) = 0.5*(grad(:) + temp(:))
      diff_delta = -diff_delta
   end if
end subroutine gradient

!-----------------------------------------------------------------------------------------------------------------------
-

subroutine line_search()
   real :: local_fitness,f1,f2,f3,a1,a2,a3,da
   real :: xval(0:nsearch),yval(0:nsearch),vars_orig(nvars)
   integer :: i,j,mincoord
   write(*,*) 'line search ----------------------------------------------------------------------------------------'

   alpha = max(default_alpha,1.1*stop_delta/sqrt(dot_product(s(:),s(:))))
   vars_orig(:) = vars(:)

   xval(0) = 0.0; yval(0) = fitness_curr
   do
      call run_mult_cases(nsearch,alpha,vars_orig,xval(1:nsearch),yval(1:nsearch))
      do j=0,nsearch
         write(*,*) j,xval(j),yval(j)
      end do
      if(yval(1)>yval(0)) then
         if(alpha*sqrt(dot_product(s(:),s(:))) < stop_delta) then
            write(*,*) 'Line search within stopping tolerance : alpha = ',alpha
            return
         end if
         write(*,*) 'Too big of a step. Reducing Alpha'
         alpha = 0.5*alpha
      else
         mincoord = minimum_coordinate(nsearch+1,yval)-1
         write(*,*) 'mincoord = ',mincoord
         if(mincoord.ne.nsearch) exit
         alpha = 2.0*alpha
      end if
   end do
   a1 = xval(mincoord-1)
   a2 = xval(mincoord)
   a3 = xval(mincoord+1)
   f1 = yval(mincoord-1)
   f2 = yval(mincoord)
   f3 = yval(mincoord+1)

   da = a2-a1
   alpha = a1+da*(4.0*f2-f3-3.0*f1)/(2.0*(2.0*f2-f3-f1))
   if((alpha > a3).or.(alpha < a1)) then !For parabolas whose min is not in bounds
      alpha = a2
      if(f2 > f1) alpha = a1
   end if
   vars(:) = vars_orig(:) + alpha*s(:)
   write(*,*) 'final alpha = ',alpha
end subroutine line_search

!-----------------------------------------------------------------------------------------------------------------------
-
```

```
integer function minimum_coordinate(num,vals)
   integer :: num,i
   real :: vals(num),minval
   minval = vals(1)
   minimum_coordinate = 1
   do i=2,num
      if(vals(i)<minval) then
         minval = vals(i)
         minimum_coordinate = i
      else
         exit
      end if
   end do
end function minimum_coordinate

!-----------------------------------------------------------------------------------------------------------------------------
-

subroutine run_mult_cases(ncases,start_alpha,vars_orig,x,y)
   integer :: ncases,i
   real :: start_alpha,vars_orig(nvars)
   real ::x(ncases),y(ncases)
   character(50) :: command

   do i=1,ncases
      x(i) = real(i)*start_alpha
      vars(:) = vars_orig(:) + x(i)*s(:)
      call start_case(i)
   end do
   vars(:) = vars_orig(:)
   do while(.not.mult_done(ncases))
      call sleep(1)
   end do
   call sleep(1)   !one more time to ensure all files are totally written

   do i=1,ncases
      y(i) = case_fitness(i)
   end do
   call sleep(1)

   command = 'rm input_* output_* case_* fitness_*'
   call system(command)
end subroutine run_mult_cases

!-----------------------------------------------------------------------------------------------------------------------------
-

logical function mult_done(ncases)
   implicit none
   integer :: ncases,ios(ncases),i
   character(50)::filename

   do i=1,ncases
      write(filename,*) i
      filename = 'fitness_'//trim(adjustl(filename))//'.txt'
```

```
      open(i*100,file=filename,status='old',iostat=ios(i))
   end do
   if(count(ios==0)==size(ios)) then
      mult_done = .true.
   else
      mult_done = .false.
   end if

   do i=1,ncases
      if(ios(i)/=0) cycle
      close(i*100)
   end do
end function mult_done

!-----------------------------------------------------------------------------------------------------------------------
-

real function case_fitness_single(case_num)
   integer :: case_num,ierror
   character(50)::filename,command

   call start_case(case_num)
   do while(.not.one_done(case_num))
      call sleep(1)
   end do
   call sleep(1)   !one more time to ensure file is totally written

   case_fitness_single = case_fitness(case_num)
   command = 'rm input_* output_* case_* fitness_*'
   call system(command)
end function case_fitness_single

!-----------------------------------------------------------------------------------------------------------------------
-

subroutine start_case(case_num)
   implicit none
   integer :: case_num,ierror,i
   character(50)::file_i,file_o,command

   write(file_o,*) case_num
   file_i = 'input_'//trim(adjustl(file_o))//'.txt'
   file_o = 'output_'//trim(adjustl(file_o))//'.txt'
   open(unit = 10, File = file_i, status="replace", action = "write", iostat = ierror)
   do i=1,nvars
      write(10,*) vars(i)
   end do
   write(10,*) case_num
   close(10)
   command = './a.out < '//trim(adjustl(file_i))//' > '//trim(adjustl(file_o))//' &'
!    write(*,*) command
   call system(command)
end subroutine start_case

!-----------------------------------------------------------------------------------------------------------------------
-
```

```
logical function one_done(case_num)
   implicit none
   integer::case_num,ios
   character(50)::filename

   write(filename,*) case_num
   filename = 'fitness_'//trim(adjustl(filename))//'.txt'
   open(100,file=filename,status='old',iostat=ios)
   if(ios==0) then
      one_done = .true.
      close(100)
   else
      one_done = .false.
   end if
end function one_done

!-----------------------------------------------------------------------------------------------------------------------------
-

logical function all_done()
   implicit none
   integer ::ios(nvars+1),i
   character(50)::filename

   !Check for 0_ file
   open(100,file="fitness_0.txt",status='old',iostat=ios(nvars+1))
   if(ios(nvars+1)==0) close(100)

   do i=1,nvars
      if(opton(i).eq.1) then
         write(filename,*) i
         filename = 'fitness_'//trim(adjustl(filename))//'.txt'
         open(i*100,file=filename,status='old',iostat=ios(i))
      else
         ios(i)=0
      end if
   end do
   if(count(ios==0)==size(ios)) then
      all_done = .true.
   else
      all_done = .false.
   end if

   do i=1,nvars
      if(ios(i)/=0) cycle
      close(i*100)
   end do
end function all_done

end module bfgs
```

III. Code Iterate Over Different Variables

```
program main
   use solver
   implicit none
   CHARACTER*(50):: filename
   integer :: case_num,ierror,i,j
   real :: case_fitness,case_fitness_Moody,fitness,weighting,fitness_uprofile,fitness_Moody
   real :: Roverks_(8)
   real :: sigmak_(5),kwall_(4)

   Roverks_ = (/ 15.,30.6,60.,126.,252.,507.,0.5/0.0003,0.5/0.000058 /)
   kwall_ = (/ 1.0,0.5,0.1,0.05 /)
   sigmak_ = (/ 2.,3.,4.,5.,6. /)
   call set_solver_vals()

   call solver_allocate()
   call create_grid()


   write(*,*) 'Phillips Rough Flow k-lambda Model'

!   write(*,*) 'Roverks (',Roverks,' ):'
!   read(5,'(a)') rec
!   if(rec .ne. ' ') read(rec,*) Roverks
!   if( Roverks==1667) Roverks=0.5/0.0003
!   if( Roverks==8621) Roverks=0.5/0.000058
!   RoverksUser = Roverks

   weighting = 0.5
   kwall = 0.05
!   A0 = 0.0025
!   A10 = 0.012
!   A11 = 0.05
!   A12 = 0.09
!   A13 = 0.39
!   B00 = 0.0713244976729399
!   B01 = 0.25
!   B10 = 1.37
!   B11 = -14.0
!   B20 = -0.7
!   B21 = 40.


!   write(*,*) 'sigmak (',sigmak ,' ):'
!   read(5,'(a)') rec
!   if(rec .ne. ' ') read(rec,*) sigmak
!
!   write(*,*) 'kwall (',kwall ,' ):'
!   read(5,'(a)') rec
!   if(rec .ne. ' ') read(rec,*) kwall

!   write(*,*) 'weighting (',weighting ,' ):'
!   read(5,'(a)') rec
!   if(rec .ne. ' ') read(rec,*) weighting
```

```
!    case_num = 0
!    write(*,*) 'Enter integer for case (',case_num,'):'
!    read(5,'(a)') rec
!    if(rec .ne. ' ') read(rec,*) case_num

open(unit = 20, File = 'fitness.txt')
   write(20,*) '      sigmak                kwall                fitness              fitness_uprofile        fitness_4Cf'
close(20)


   do i = 1,5

      sigmak = sigmak_(i)

      do j=1,4
         kwall = kwall_(j)
         fitness_uprofile = case_fitness(case_num)  ! done at ksp = 80000
         fitness_Moody = case_fitness_Moody(case_num)  ! ksp varies
         fitness = weighting*fitness_Moody+(1.-weighting)*fitness_uprofile

         open(unit = 20, File = 'fitness.txt', status = "OLD", access = "append", iostat = ierror)
         write(20,'(100ES25.12)') sigmak,kwall,fitness,fitness_uprofile,fitness_Moody
         close(20)

      enddo
      write(*,*) 'sigmak = ', sigmak
   enddo


!    open(unit=15,file = 'fitness.txt')
!       write(15,'(20ES24.14)')Roverks,kwall,A0,A1,B0,B1,B2,weighting,fitness
!    close(15)

!    if (RoverksUser<0) then
!       open(unit=10,file = 'indiv_fitness.txt')
!       do i=1,8
!            write(10,*) Roverks_(i),fitness_Moody_(i),fitness_uprofile_(i),weighting*fitness_Moody_(i)+(1.-
weighting)*fitness_uprofile_(i)
!       enddo
!       write(10,*)
!                                                                                                  write(10,*)
RoverksUser,sum(fitness_Moody_)/8.,sum(fitness_uprofile_)/8.,weighting*sum(fitness_Moody_)/8.+(1.-
weighting)*sum(fitness_uprofile_)/8.
!       close(10)
!    else
!       open(unit=10,file = 'indiv_fitness.txt')
!          write(10,*) Roverks,fitness_Moody,fitness_uprofile,fitness
!       close(10)
!    endif

end program main



!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
real function case_fitness_Moody(case_num)
   use solver
   integer :: ngood,total,ierror,i,jj,case_num
   real :: results(7),local_Roverks,Roverks_(7)
   character*(50) :: filename
   results(:) = 0.0
   total = 0
   local_Roverks = Roverks

   !Roverks_ = (/ 15.,30.6,60.,126.,252.,507.,0.5/0.0003,0.5/0.000058 /)
   Roverks_ = (/ 15.,30.6,60.,126.,252.,507.,0.5/0.000058 /)


   100 FORMAT (1X, 1000ES22.14)
   write(filename,*) case_num
   filename = 'caseMoody_'//trim(adjustl(filename))//'.txt'
!   open(unit = 10, File = filename, status="replace", action = "write", iostat = ierror)
!   write(10,*) '  Roverks             Re_bulk             4CF                 Colebrook          '

   if(RoverksUser.gt.0.0) then
      Roverks = RoverksUser
      call run_case( Roverks,  ngood,results(1)); total = total + ngood
!       do i=1,ngood
!          write(10,100) Roverks,vRebulk(i),vCF(i),func_Colebrook(vRebulk(i))
!       end do
!       write(10,*)
      fitness_Moody_(1) = sqrt(results(1)/ngood)
      case_fitness_Moody = fitness_Moody_(1)
   else
      do jj=1,7
         Roverks = Roverks_(jj)
         call update_variable_vRoverks()
         call run_case( Roverks,  ngood,results(jj)); total = total + ngood
!          do i=1,ngood
!                                                                                                                write(10,100)
Roverks,vRebulk(i),vCF(i),func_Colebrook(vRebulk(i)),(func_Colebrook(vRebulk(i)) - vCF(i))**2
!          end do
!          write(10,*)
         fitness_Moody_(jj) = (results(jj)/ngood)
      enddo

      case_fitness_Moody = sum(fitness_Moody_)/7.
   end if

   !case_fitness_Moody =  sqrt(sum(results)/real(total))

!   write(10,*) '  kappa = ',kappa
!   write(10,*) '  gamma = ',gamma
!   write(10,*)
!   do jj=1,8
!      write(10,*) Roverks_(jj),results(jj)
!   enddo
!   write(10,*)
!   write(10,*) case_fitness_Moody, ' = case fitness'
!   close(10)
```

```
end function case_fitness_Moody


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

subroutine run_case(Roverks_val,ngood,error)
    use solver
    integer :: ngood
    real :: Roverks_val,error

   Roverks = Roverks_val
   call update_variable_vRoverks()


    call p_vksp(0,ngood,error)
!    write(*,*) 'ngood = ',ngood
!    write(*,*) 'fitness Moody chart = ',error
end subroutine run_case

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!



real function case_fitness(case_num)
    use solver
    integer :: ngood,ierror,i,jj,case_num
    real,dimension(14) :: results,local_Roverks
    real :: Roverks_(8)
    character*(50) :: filename
    results(:) = 0.0
    Roverks_ = (/ 15.,30.6,60.,126.,252.,507.,0.5/0.0003,0.5/0.000058 /)
    ksp = 80000.

    100 FORMAT (1X, 1000ES22.14)
!    write(filename,*) case_num
!    filename = 'caseUprofile_'//trim(adjustl(filename))//'.txt'
!    open(unit = 10, File = filename, status="replace", action = "write", iostat = ierror)
!    write(10,*) '  R/ks               Fitness'

    if(RoverksUser.gt.0.0) then
       Roverks = RoverksUser
       call update_variable_vRoverks()
       call run_case_Velocity( Roverks,1,results(1))
!       write(10,100) Roverks,results(1)
!       write(10,*)
       total = 1
       fitness_uprofile_ = results(1)
    else
       do jj=1,8
          Roverks = Roverks_(jj)
          call update_variable_vRoverks()
          call run_case_Velocity( Roverks,1,results(jj))
!          write(10,100) Roverks,results(jj)
!          write(10,*)
          fitness_uprofile_(jj) = results(jj)
```

```
        enddo
        total = 8
    end if

    case_fitness = sum(results)/real(total)

    !write(*,*) 'fitness Velocity TOTAL = ',case_fitness

!    write(10,*) '        A0 = ',A0
!    write(10,*) '        A1 = ',A1
!    write(10,*) '        B0 = ',B0
!    write(10,*) '        B1 = ',B1
!    write(10,*) '        B2 = ',B2
!    write(10,*) '     kwall = ',kwall
!!    write(10,*) 'lambdaWall = ',lambdaWall
!    !write(10,*) 'lambdaWallSlope = ',lambdaWallSlope
!!    write(10,*) 'R/ks used =',local_Roverks(1:nb_of_sets)
!    write(10,*)
!    write(10,*) case_fitness, ' = case fitness (Nikuradse u+ exp data)'
!    close(10)


end function case_fitness

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

subroutine run_case_Velocity(Roverks_val,case_number,error)
    use solver
    integer, intent(in) :: case_number ! can only be 1 through 6
    real :: Roverks_val,error

    call update_variable_vRoverks()

    call p_getu(case_number,kshat,error)

end subroutine run_case_Velocity

!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
subroutine set_solver_vals()
    use solver
    real :: RC_ltxt(50)
    real :: a,b,c,d,e,f

      Roverks = 15.0
          kr = 0.5/Roverks
         Cnu = 0.006
        Clam = Cnu**2
      sigmak = 2.0
         ksp = 80000.0

       kappa = 0.403
       gamma = 0.0324
        beta = 1.000002
           n = 3201
```

```
      nplot = 51

      nuhat = 2.*kr/ksp
       Rtau = 1./nuhat
       kshat = 2.*kr


       a = 0.00233
       b = 0.659
       c = 0.0307
       d = 0.00551
       e = 0.209
       f = 0.0421



end subroutine set_solver_vals
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
module solver
   IMPLICIT NONE
   integer :: n,nb_of_sets,nb_pts_per_set,nplot,nstat
   integer :: FullyRough,FullySmooth

   real :: Rtau,kr,ksp,beta,nuhat,kshat,Rshat,Roverks,qkguess
   real :: kappa,RoverksUser,fitness_Moody_(7),fitness_uprofile_(8)
   real :: Cnu, sigmak,Clam
   real :: kwall,lambdaWallSlope,lambdaWall,lambdaCenter
   real :: Re,Df,um,km
   real :: gamma
   real                                                                                  ::
A0,A1,B0,B1,B2,A10,A11,A12,A13,B00,B01,B02,B03,B10,B11,B12,B20,B21,B22,A1_exp,A1_cst

   CHARACTER*(80):: rec,init,file_r
   CHARACTER(LEN=100)::fn

   real, allocatable, dimension(:)::r_hat,yplus
   real, allocatable, dimension(:)::k,kprime
   real, allocatable, dimension(:)::qk,qkprime
   real, allocatable, dimension(:)::u,uprime
   real, allocatable, dimension(:)::nu
   real, allocatable, dimension(:)::h,lambda

   real, allocatable, dimension(:)::vCF,vRebulk


contains

subroutine solver_allocate()
   !Allocate Memory
   ALLOCATE(r_hat(n)); ALLOCATE(yplus(n))
   ALLOCATE(k(n));     ALLOCATE(kprime(n));
   ALLOCATE(qk(n));     ALLOCATE(qkprime(n));
   ALLOCATE(u(n));     ALLOCATE(uprime(n));
   ALLOCATE(nu(n));
```

```
   ALLOCATE(h(n)); ALLOCATE(lambda(n));
   ALLOCATE(vRebulk(nplot)); ALLOCATE(vCf(nplot))
end subroutine solver_allocate

subroutine solver_deallocate()
   !Deallocate Memory
   DEALLOCATE(r_hat); DEALLOCATE(yplus);
   DEALLOCATE(k);     DEALLOCATE(kprime);
   DEALLOCATE(qk);     DEALLOCATE(qkprime);
   DEALLOCATE(u);     DEALLOCATE(uprime);
   DEALLOCATE(nu);
   DEALLOCATE(h); DEALLOCATE(lambda);
   DEALLOCATE(vRebulk);DEALLOCATE(vCf);
end subroutine solver_deallocate


subroutine create_grid()
   integer :: j
   real :: eta,cbeta,dzeta

   if (beta>0) then
      dzeta = 1.0/real(n-1)
      !Create Grid
      do j=1,n,1
         eta = real(n-j)/real(n-1)
         cbeta = ((beta+1.0)/(beta-1.0))**(1.0-eta)
         if(beta .eq. 0.0) then
            r_hat(j) = 1.0 - real(j-1)*dzeta
         else
            r_hat(j) = 1.0 - (beta+1.0 - (beta-1.0)*cbeta)/(1.0 + cbeta)
         end if
      end do
      r_hat(1) = 0.0
   else ! uniform grid, same spacing
      do j=1,n
         r_hat(j) = real(j-1) / real(n-1)
      enddo
   endif

end subroutine create_grid

subroutine update_variable_vRoverks()
   real :: C_0,C_1,C_2,C_3,C_4,C_5
   real :: sigma_k, k_wall
   real                                                                       ::
A_lam0,A_lam10,A_lam11,A_lam12,A_lam13,B_lam00,B_lam01,B_lam02,B_lam03,B_lam10,B_lam11,
B_lam20,B_lam21


   kshat = 1./Roverks
   kr = kshat/2.
   nuhat = 2.*kr/ksp
   Rtau = 1./nuhat
   Rshat = 1./(gamma*kshat)
   sigma_k = sigmak
   k_wall = kwall
```

```
      Cnu   =  1.1430803817E-04*sigma_k**4 - 2.1568032682E-03*sigma_k**3                &
            + 1.5454411906E-02*sigma_k**2 - 5.3147683089E-02*sigma_k + 8.3648609370E-02      !
FINAL NO CHANGE FROM V8.1
!
      A_lam0  = - 5.4806699600E-06*sigma_k**4 + 1.0829889892E-04*sigma_k**3                &
            - 5.8824903497E-04*sigma_k**2 + 6.4266598213E-05*sigma_k + 7.0559295507E-03      !
FINAL NO CHANGE FROM V8.1
!
      A_lam10 =  6.0548405176E-06*sigma_k**4 - 1.7464041639E-04*sigma_k**3                &
            + 1.7083135016E-03*sigma_k**2 - 7.8015697237E-03*sigma_k + 2.3620283331E-02      !
FINAL NO CHANGE FROM V8.1
!
      A_lam11 =  1.6430210539E-04*sigma_k**4 - 3.2876048062E-03*sigma_k**3                &
            + 2.3668508652E-02*sigma_k**2 - 7.4817705811E-02*sigma_k + 1.3742572373E-01      !
FINAL NO CHANGE FROM V8.1
!
        C_4 = - 3.6249854255E-04*k_wall**2  - 2.3771783095E-04*k_wall  + 3.1811391789E-04      !
<<<<<<<<<<
        C_3 =   6.7456577268E-03*k_wall**2  + 3.5031152723E-03*k_wall  - 7.6039769524E-03      !
<<<<<<<<<<
        C_2 = - 4.3056163634E-02*k_wall**2  - 2.0050955122E-02*k_wall  + 5.7370834148E-02      !
<<<<<<<<<<
        C_1 =   1.1098738361E-01*k_wall**2  + 5.7323021287E-02*k_wall  - 1.5770938564E-01      !
<<<<<<<<<<
        C_0 = - 9.7888362645E-02*k_wall**2  - 7.6772775598E-02*k_wall  + 1.6397877418E-01      !
<<<<<<<<<<
      A_lam12 =    C_4*sigma_k**4 + C_3*sigma_k**3 + C_2*sigma_k**2 + C_1*sigma_k + C_0
! FINAL CHANGED FROM V8.1 <<<<<<<<<<
!
      A_lam13 =  2.1775071643E-04*sigma_k**4 - 4.6125648020E-03*sigma_k**3                &
            + 3.2004745006E-02*sigma_k**2 - 7.6059412219E-02*sigma_k + 2.6786227707E-01      !
FINAL NO CHANGE FROM V8.1
!
      B_lam00 =  1.5571555961E-03*sigma_k**4 - 3.7455015562E-02*sigma_k**3                &
            + 3.0085113473E-01*sigma_k**2 - 9.7875053089E-01*sigma_k + 1.1002011812E+00     ! V
2.5 NO CHANGE FROM V8.1
!
      B_lam01 = - 1.7611480539E-03*sigma_k**4 + 3.3472445299E-02*sigma_k**3                &
            - 2.1562199190E-01*sigma_k**2 + 4.9483981773E-01*sigma_k + 4.4081967221E-01      !
FINAL NO CHANGE FROM V8.1
!
        C_4 =  4.5958241000E-03*k_wall**2  + 2.0834333483E-03*k_wall  - 1.3277370846E-02
        C_3 = - 9.3116503886E-02*k_wall**2  - 2.8569790820E-02*k_wall  + 2.7587164438E-01
        C_2 =  6.0395589241E-01*k_wall**2  + 1.8761243453E-01*k_wall  - 2.0079111202E+00
        C_1 = - 1.4946016082E+00*k_wall**2  - 7.3780887177E-01*k_wall  + 6.0340918034E+00
        C_0 =  1.2142113765E+00*k_wall**2  + 1.2696368571E+00*k_wall  - 7.2873561012E+00
      B_lam02 =    C_4*sigma_k**4 + C_3*sigma_k**3 + C_2*sigma_k**2 + C_1*sigma_k + C_0
! FINAL NO CHANGE FROM V8.1
!
      B_lam03 =  6.3453858956E-04*sigma_k**4 - 1.7156286122E-02*sigma_k**3                &
            + 1.4904850864E-01*sigma_k**2 - 5.3187100685E-01*sigma_k + 7.9782349433E-01      !
FINAL NO CHANGE FROM V8.1
!
      B_lam10 = - 1.0134626360E-03*sigma_k**4 + 1.9147367622E-02*sigma_k**3                &
```

```
          - 1.2222170912E-01*sigma_k**2 + 2.5530412815E-01*sigma_k + 1.4355452292E+00        !
FINAL NO CHANGE FROM V8.1
!
     B_lam11 =                         - 6.6206887937E-03*sigma_k - 1.3830728894E+01      ! V 2   NO
CHANGE FROM V8.1
!B_lam11 = B11
       C_4 = - 2.0201876364E-03*k_wall**2  + 4.3279392861E-03*k_wall  - 1.3949382336E-03
       C_3 =   2.6570830671E-02*k_wall**2  - 6.4136620678E-02*k_wall  + 2.3621757343E-02
       C_2 = - 1.1989031734E-01*k_wall**2  + 3.3373664338E-01*k_wall  - 1.3864710895E-01
       C_1 =   2.1942375688E-01*k_wall**2  - 7.0814005373E-01*k_wall  + 3.0315634574E-01
       C_0 = - 1.4066477454E-01*k_wall**2  + 4.5837394149E-01*k_wall  - 8.0853545298E-01
     B_lam20  =    C_4*sigma_k**4  +  C_3*sigma_k**3  +  C_2*sigma_k**2  +  C_1*sigma_k  +  C_0
! FINAL NO CHANGE FROM V8.1
!
     B_lam21 =                         - 2.6815510578E-03*sigma_k + 4.0037296283E+01      ! V 2   NO
CHANGE FROM V8.1


    A0 = A_lam0
    A1   =   A_lam11  +  A_lam12*kshat  +  (A_lam10  -  A_lam11  -  A_lam12*kshat)*exp(-
(kshat**A_lam13))
    B0  =  B_lam01 + B_lam02*kshat + (B_lam00 - B_lam01 - B_lam02*kshat)*exp(-(kshat**B_lam03))
    B1  =  B_lam10 + B_lam11*kshat
    B2  =  B_lam20 + B_lam21*kshat


end subroutine



!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!   Compare   velocity   with   Nikuradse   exp   data
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
subroutine p_getu(case_number,ksoverR_in,error)
   integer :: iprint, niter,icompare,i,j
   integer, intent(in) :: case_number
   real :: error,junk,pRMS,RMS
   real, intent(in) :: ksoverR_in
   real :: uplus_reference(12),uhat_reference(12),r_hat_discretized(12),u_fromEmpLambda(12)
   real :: a,b,u1,u2,r1,r2


   do i = 1,n
      lambda(i) = lambdahat(r_hat(i))
   enddo

   call integrate_k_fromWall()

!    write(*,*)
!    write(*,*) '  Re (um) = ' , Re
!    write(*,*) ' 4Cf      = ' , Df
!    write(*,*) 'Colebrook = ', func_Colebrook(Re)
!    write(*,*)
!    write(*,*) '        B0 = ',B0
!    write(*,*) '        B1 = ',B1
!    write(*,*) '        B2 = ',B2
!    write(*,*) '        A0 = ',A0
```

```
!    write(*,*) '       A1 = ',A1
!    write(*,*) 'lambdaWall = ', lambdaWall
!    write(*,*) '     kwall = ',kwall
!    write(*,*)


!    open(unit=10,file='data.txt')
!       do i=n,1,-1
!          write(10,*) r_hat(i),1.-r_hat(i),u(i),k(i),lambda(i),nu(i), r_hat(i)/( nuhat+nu(i) )
!       enddo
!    close(10)


     ! Compare u+ ( u+ from empirical nut vs. u+ from RK4 with empirical lambda_hat)
     r_hat_discretized = (/ 0.0,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.95 /)


    j=1
    u_fromEmpLambda(1) = u(1)
    call get_uplus_uhat(0.0,uplus_reference(1),junk)


    do icompare=2,12
       do while ((j<=n).and. (r_hat(j)-r_hat_discretized(icompare)<=0.)  )
          j=j+1
       enddo
       j=j-1

       if        (r_hat_discretized(icompare)-r_hat(j)>         r_hat(j+1)-r_hat_discretized(icompare))         then
!r_hat_discretized(icompare) closer to r_hat(j+1)
          call get_uplus_uhat(r_hat(j+1),uplus_reference(icompare),junk)
          u_fromEmpLambda(icompare) = u(j+1)
          pRMS                                    =                                    abs((uplus_reference(icompare)-
u_fromEmpLambda(icompare))/uplus_reference(icompare))
       else
          call get_uplus_uhat(r_hat(j),uplus_reference(icompare),junk)
          u_fromEmpLambda(icompare) = u(j)
          pRMS                                    =                                    abs((uplus_reference(icompare)-
u_fromEmpLambda(icompare))/uplus_reference(icompare))
       endif

    enddo


    error = sum( abs(uplus_reference-u_fromEmpLambda)/uplus_reference)/12.

end subroutine p_getu



! ---------------------------------------------------------------------------------------
! uplus/uhat when nut = Eq. 106
! ---------------------------------------------------------------------------------------

subroutine get_uplus_uhat(r_hat,u_plus,u_hat)
    real :: yhat,r_hat
```

```
   real :: delta,deltap
   real :: u_plus,u_hat
   real :: D0,D1,D2,D3,D4,D5,C0,C1,C2,C3,C4,C5,C6,C7,Acoeff,Bcoeff,ecoeff,nutc

      Acoeff = 1.2
      Bcoeff = 0.0
      ecoeff = 1.4
      nutc = 0.053
      C7 = -1.3

      D0 = (Rshat+1.)**2*(log(Rshat+1.)-3./2.)+2.*Rshat+3./2.
      C0 = Acoeff*kshat**ecoeff+Bcoeff
      C1 = Rshat**3./((Rshat+1.)*D0)
      C2 = Rshat**2.*(Rshat**2/((Rshat+1.)**2)-kappa/nutc)/(2.*D0)
      C3 = Rshat**5./(3.*(Rshat+1.)**3*D0)
      D1 = -C0-C1-C2-C3-C7
      D2 = -C1-2.*C2-3.*C3-7.*C7
      D3 = -C0/2.-C1/3.-C2/4.-C3/5.-C7/9.
      C4 = -39.*D1+3.*D2+168.*D3
      C5 = 84.*D1-7.*D2-336.*D3
      C6 = -44.*D1+4.*D2+168.*D3


      yhat = 1.-r_hat

      delta = C0 + C1*r_hat + C2*r_hat**2 + C3*r_hat**3 + C4*r_hat**4 + C5*r_hat**5 + C6*r_hat**6
+ C7*r_hat**7
      deltap = -(C1 + 2.*C2*r_hat + 3.*C3*r_hat**2 + 4.*C4*r_hat**3 + 5.*C5*r_hat**4 +
6.*C6*r_hat**5 + 7.*C7*r_hat**6)

      u_hat = Rshat**2*log(Rshat*yhat+1.)/D0+delta
      u_plus = log(Rshat*yhat+1.)/kappa+D0*delta/(kappa*Rshat**2)

end subroutine



! ---------------------------------------------------------------------------------------
! Lambda_hat Empirical Function
! ---------------------------------------------------------------------------------------

function lambdahat(r_hat)
   real :: r_hat,lambdahat,yhat

   yhat = 1.-r_hat
   lambdahat        =        (A0*kshat+A1*yhat)*(1.-.5*yhat)*(B0+B1*r_hat**2+B2*r_hat**4+(1.-B0-B1-
B2)*r_hat**6)

end function


! ---------------------------------------------------------------------------------------
!   INTEGRATE from WALL
! ---------------------------------------------------------------------------------------

subroutine integrate_k_fromWall()
```

```
integer :: i,niter,nitermax
real :: t0,y0(5),dt,y(5),qwall(3),qcenter(3)

! integrates from the wall, shoots to the centerline to have Q+ = 0 at the centerline
!kwall = 6120.1*kshat**3-1503.6*kshat**2+144.43*kshat-0.0309

!write(*,*) 'kwall = ', kwall

niter = 1
nitermax = 200
nstat = 0
qkguess = 2.0

!write(*,*) 'Iterate on Q...'


do while ((qkguess>=-1.).and.(nstat==0) )
niter = 1

   ! GUESS 1 ON Q
   ! ---------------
   t0 = 0.
   k(n) = kwall
   qk(n) = qkguess   ! initial guess for q(wall)
   qwall(1) = qk(n)
   y0(1) = k(n)
   y0(2) = qk(n)
   !write(*,*) i,k(n),qk(n)

   do i = n-1,1,-1
      dt = r_hat(i+1)-r_hat(i)  ! dt is equivalent of dy, not dr_hat
      call RK4_fromWall(2,t0,y0,dt,y)
      t0 = t0+dt
      y0 = y
      k(i) = y(1)
      qk(i) = y(2)
   enddo
   qcenter(1) = qk(1)

   !write(*,*) niter, qk(1)


   ! GUESS 2 ON Q
   ! ---------------
   niter = niter + 1
   t0 = 0.
   k(n) = kwall
   if (qkguess<0) then
      qk(n) = qkguess*.95   ! initial guess for q(wall)
   else
      qk(n) = qkguess*1.05
   endif
   qwall(2) = qk(n)
   y0(1) = k(n)
   y0(2) = qk(n)
```

```
do i = n-1,1,-1
   dt = r_hat(i+1)-r_hat(i)
   call RK4_fromWall(2,t0,y0,dt,y)
   t0 = t0+dt
   y0 = y
   k(i) = y(1)
   qk(i) = y(2)
enddo
qcenter(2) = qk(1)
!write(*,*) niter, qk(1)


! ITERATE ON Q (secant method)
! ----------------------------
do while ((abs(qk(1))>1.0e-12).and.(niter<nitermax))
   niter = niter + 1
   t0 = 0.
   k(n) = kwall
   qk(n) = qwall(2)-qcenter(2)*( qwall(2)-qwall(1) ) / ( qcenter(2)-qcenter(1) )
   qwall(3) = qk(n)
   y0(1) = k(n)
   y0(2) = qk(n)

   do i = n-1,1,-1
      dt = r_hat(i+1)-r_hat(i)
      call RK4_fromWall(2,t0,y0,dt,y)
      t0 = t0+dt
      y0 = y
      k(i) = y(1)
      qk(i) = y(2)
   enddo
   qcenter(3) = qk(1)
   !write(*,*) niter, qk(1),qk(n)

   qwall(1) = qwall(2)
   qwall(2) = qwall(3)
   qcenter(1) = qcenter(2)
   qcenter(2) = qcenter(3)

enddo


niter = niter + 1
t0 = 0.
k(n) = kwall
qk(n) = qwall(2)-qcenter(2)*( qwall(2)-qwall(1) ) / ( qcenter(2)-qcenter(1) )
qwall(3) = qk(n)
y0 = 0.
y0(1) = k(n)
y0(2) = qk(n)
u(n) = 0.

do i = n-1,1,-1
   dt = r_hat(i+1)-r_hat(i)
   call RK4_fromWall(5,t0,y0,dt,y)
   t0 = t0+dt
```

```
            y0 = y
            k(i) = y(1)
            qk(i) = y(2)
            u(i) = y(3)
        enddo
        qcenter(3) = qk(1)
        km = y(5)
        um = y(4)
        !write(*,*) niter, Roverks,qk(1),qwall(3)
        nu = lambda*sqrt(k)
        h = Cnu*k /nuhat
        Df =  8./um**2.
        Re = 2.*um/nuhat

        if (isnan(Re)) then
            qkguess = qkguess-0.1
            nstat = 0
        else
            nstat =1
        endif

    enddo


    if ((niter>=nitermax).or.(isnan(Re))) then
        nstat = 0
    else
        nstat = 1
        !write(*,*) niter, Roverks,qk(1),qwall(3)
    endif


end subroutine




subroutine RK4_fromWall(nn,t0,y0,dt,y)
    ! solve a system of n first order differential equations
    ! dy(i)/dt = f(i,t,y)
    ! calls the function f

    integer, intent(in) :: nn ! number of equation to be solved
    real, dimension(nn,4) :: k_ ! table of coefficient (k1, k2, k3 k4); 4 coeffs per equation
    real, intent(inout) :: t0, y0(nn),dt
    real :: coeff(4),Rtau
    real, intent(out) :: y(nn)
    integer :: i,j

    if (t0+dt>1.) dt=1.-t0
    coeff = (/ 1./6., 1./3., 1./3., 1./6. /)
    do j=1,nn
        k_(j,1)=f_fromWall(j,t0,y0)*dt
        y(j)=y0(j)+k_(j,1)/2.
    end do
    do j=1,nn
```

```
      k_(j,2)=f_fromWall(j,t0+dt/2.,y)*dt
   end do
   do j=1,nn
      y(j)=y0(j)+k_(j,2)/2.
   end do
   do j=1,nn
      k_(j,3)=f_fromWall(j,t0+dt/2.,y)*dt
   end do
   do j=1,nn
      y(j)=y0(j)+k_(j,3)
   end do
   do j=1,nn
      k_(j,4)=f_fromWall(j,t0+dt,y)*dt
   end do
   do j=1,nn
      y(j)=y0(j)
      do i=1,4
        y(j)=y(j)+coeff(i)*k_(j,i)
      end do
   end do

end subroutine


function f_fromWall(i,zeta,y)
   integer, intent(in) :: i
   real, intent(in) :: zeta,y(*)   ! y(1) = k, y(2) = q
   real :: f_fromWall,r_hat
   ! input zeta = 0 at wall, 1 at centerline
   !if (zeta<0.) zeta=0.

   r_hat = 1. - zeta ! r_hat = 0 at centerline, 1 at wall
   if (i==1) then  ! dk/dr_hat
      if (r_hat <= 10**(-16./6.)) then   !(r_hat <= 10**(-16./6.)) then  ! at the centerline, exception /0
         f_fromWall = 0.
      else
         f_fromWall = -y(2)/r_hat*3./( nuhat+5.*(lambdahat(r_hat)*sqrt( abs(y(1)) ) )/sigmak) ! corresponds
to dk/dr_hat
         !write(*,*) nut(r_hat),y(2)
      endif
   elseif (i==2) then   ! dq/dr_hat
      f_fromWall                =                                (lambdahat(r_hat)*sqrt(abs(y(1))))*r_hat**3./(nuhat+
(lambdahat(r_hat)*sqrt(abs(y(1))))                                                    )**2.                                -
Cnu**2.*nuhat/(lambdahat(r_hat)*sqrt(abs(y(1))))**2.*y(1)**2.*r_hat
      !if  ((nut(r_hat)==0.).and.(r_hat==0) )  f_fromWall=0.
   elseif (i==3) then   ! du/dr_hat
      f_fromWall = -r_hat/( nuhat+(lambdahat(r_hat)*sqrt(y(1))) )
   elseif (i==4) then   ! dum/dr_hat
      f_fromWall = -y(3)*r_hat*2.
   elseif (i==5) then   ! dkm/dr_hat
      f_fromWall = -y(1)*r_hat*2.
   end if
   f_fromWall = -f_fromWall  ! gives the variables in terms of d_/dy
   !if (f_fromWall=='NaN') pause
end function
```

```
subroutine p_vksp(iprint,ngood,error)
   integer :: iprint,ngood,niter
   real :: error,power

   power = 2.0
   ksp = 10**power
   Re = 0.0
   ngood = 1
   error = 0.0
   Rshat = Roverks/(gamma)

   if(iprint.eq.1) write(*,*) '    ngood   R/ks                ksp                 Re                  4CF       &
                        &          4Cf Reference          error                ' !num_iter'
   do while (((Re<1.0e8).and.(ngood<=nplot)).or.(ngood<5))
      nuhat = 2.*kr/ksp
      Rtau = 1./nuhat

      call integrate_k_fromWall()
      vRebulk(ngood) = Re
      vCF(ngood) = Df

      if((Re<1.0e8).or.(ngood<4))  error  =  error  +  abs(reference_4Cf(Re,Roverks)  -
vCF(ngood))/reference_4Cf(Re,Roverks)
!                       if((Re<1.0e8).or.(ngood<4))  error  =  error  +  ((reference_4Cf(Re,Roverks)  -
vCF(ngood))/func_Colebrook(Re))**2 ! % RMS
      if(iprint.eq.1) write(*,*) ngood,Roverks,ksp,Re,Df,reference_4Cf(Re,Roverks),error !,niter
      ngood = ngood + 1
      power = power + 0.25
      ksp = 10.0**power
   end do
   ngood = ngood - 2
end subroutine p_vksp

real function func_Colebrook(Re)
   implicit none
   real :: Re,old,new
   if(Re<100.0) then
      func_Colebrook = 64.0/Re
   else
      new = (-1.8*log10((kr/3.7)**(1.11) + 6.9/Re))**(-2.0)
      old = 10.0
      do while(abs((new-old)/new) .gt. 1.0e-5)
         old = new
         new = (-2.0*log10(kr/3.7 + 2.51/(Re*sqrt(old))))**(-2.0)
      end do
      func_Colebrook = new
   end if
return; end function func_Colebrook


real function reference_4Cf(Re,Roverks)
   real :: Re,Roverks
   integer :: i,indix
```

```
   reference_4Cf = (2.0*log10(Roverks)+1.74)**(-2.0)

end function

end module solver



program readOUtput

real :: junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
integer:: ierr

open(unit=10,file='output.txt')
   open(unit=99,file='sk1kw1//optimization.txt')
   read(99,*)
   ierr=1
      do while (ierr/=-1)
         read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
      enddo
      write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
   close(99)

   open(unit=99,file='sk1kw05//optimization.txt')
   read(99,*)
   ierr=1
      do while (ierr/=-1)
         read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
      enddo
      write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
   close(99)

   open(unit=99,file='sk1kw01//optimization.txt')
   read(99,*)
   ierr=1
      do while (ierr/=-1)
         read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
      enddo
      write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
   close(99)

   open(unit=99,file='sk1kw005//optimization.txt')
   read(99,*)
   ierr=1
      do while (ierr/=-1)
         read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
      enddo
      write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
   close(99)

   open(unit=99,file='sk2kw1//optimization.txt')
   read(99,*)
   ierr=1
      do while (ierr/=-1)
         read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
      enddo
```

```
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk2kw05//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk2kw01//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk2kw005//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk3kw1//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk3kw05//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk3kw01//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)
```

```
open(unit=99,file='sk3kw005//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk4kw1//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk4kw05//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk4kw01//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk4kw005//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk5kw1//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk5kw05//optimization.txt')
```

```
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk5kw01//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk5kw005//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk6kw1//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk6kw05//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk6kw01//optimization.txt')
read(99,*)
ierr=1
   do while (ierr/=-1)
      read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
   enddo
   write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
close(99)

open(unit=99,file='sk6kw005//optimization.txt')
read(99,*)
ierr=1
```

```
      do while (ierr/=-1)
         read(99,*,iostat=ierr) junk,junk,junk,Roverks,sigmak,kwall,coef1,coef2,weighting,fitness, junk,junk
      enddo
      write(10,*) Roverks,sigmak,kwall,coef1,coef2,weighting,fitness
   close(99)

close(10)



end program


program createCase
real :: sk1_ini,sk2_ini,sk3_ini,sk4_ini,sk5_ini,sk6_ini
real :: sk1_ini0,sk2_ini0,sk3_ini0,sk4_ini0,sk5_ini0,sk6_ini0

sk1_ini0 = 0.04
sk2_ini0 = 0.04
sk3_ini0 = 0.04
sk4_ini0 = 0.04
sk5_ini0 = 0.04
sk6_ini0 = 0.04


sk1_ini = -0.06
sk2_ini = -0.06
sk3_ini = -0.06
sk4_ini = -0.06
sk5_ini = -0.06
sk6_ini = -0.06

call createFiles(8,'sk1kw005',1.,0.05,sk1_ini0,sk1_ini)
call createFiles(7,'sk1kw01',1.0,0.1,sk1_ini0,sk1_ini)
call createFiles(7,'sk1kw05',1.0,0.5,sk1_ini0,sk1_ini)
call createFiles(6,'sk1kw1',1.0,1.0,sk1_ini0,sk1_ini)
write(*,*) 'sk1'

call createFiles(8,'sk2kw005',2.0,0.05,sk2_ini0,sk2_ini)
call createFiles(7,'sk2kw01',2.0,0.1,sk2_ini0,sk2_ini)
call createFiles(7,'sk2kw05',2.0,0.5,sk2_ini0,sk2_ini)
call createFiles(6,'sk2kw1',2.0,1.0,sk2_ini0,sk2_ini)
write(*,*) 'sk2'

call createFiles(8,'sk3kw005',3.0,0.05,sk3_ini0,sk3_ini)
call createFiles(7,'sk3kw01',3.0,0.1,sk3_ini0,sk3_ini)
call createFiles(7,'sk3kw05',3.0,0.5,sk3_ini0,sk3_ini)
call createFiles(6,'sk3kw1',3.0,1.0,sk3_ini0,sk3_ini)
write(*,*) 'sk3'

call createFiles(8,'sk4kw005',4.0,0.05,sk4_ini0,sk4_ini)
call createFiles(7,'sk4kw01',4.0,0.1,sk4_ini0,sk4_ini)
call createFiles(7,'sk4kw05',4.0,0.5,sk4_ini0,sk4_ini)
call createFiles(6,'sk4kw1',4.0,1.0,sk4_ini0,sk4_ini)
write(*,*) 'sk4'
```

```
call createFiles(8,'sk5kw005',5.0,0.05,sk5_ini0,sk5_ini)
call createFiles(7,'sk5kw01',5.0,0.1,sk5_ini0,sk5_ini)
call createFiles(7,'sk5kw05',5.0,0.5,sk5_ini0,sk5_ini)
call createFiles(6,'sk5kw1',5.0,1.0,sk5_ini0,sk5_ini)
write(*,*) 'sk5'

call createFiles(8,'sk6kw005',6.0,0.05,sk6_ini0,sk6_ini)
call createFiles(7,'sk6kw01',6.0,0.1,sk6_ini0,sk6_ini)
call createFiles(7,'sk6kw05',6.0,0.5,sk6_ini0,sk6_ini)
call createFiles(6,'sk6kw1',6.0,1.0,sk6_ini0,sk6_ini)
write(*,*) 'sk6'

end program



subroutine createFiles(len,myDir,sk,kw,initialGuess1,initialGuess2)
integer :: len
character(len) :: myDir
real :: initialGuess1,initialGuess2,sk,kw


   call system('mkdir ' // trim(myDir) )
   call system('cp commonFiles/bfgs.f90 ./' // trim(adjustl(myDir)) )
   call system('cp commonFiles/main_bfgs.f90 ./' // trim(myDir) )
   call system('cp commonFiles/solver.f90 ./' // trim(myDir) )
   call system('cp commonFiles/main_solve.f90 ./' // trim(myDir) )
   call system('cp commonFiles/Makefile ./' // trim(myDir) )

   open(unit=99,file='job')
      write(99,*) '#PBS -q sawtooth'
      write(99,*) '#PBS -S /bin/bash'
      write(99,*) '#PBS -l nodes=1:ppn=8'
      write(99,*) '#PBS -l walltime=72:00:00'
      write(99,*) '#PBS -N ' // trim(myDir)
      write(99,*) '#PBS -j oe'
      write(99,*)
      write(99,*) 'WORK_DIR=/home/A01208071/TRANSITION/122811/currentRun/' // trim(myDir)
      write(99,*)
      write(99,*) 'cd $WORK_DIR'
      write(99,*) 'make'
      write(99,*) './bfgs.out '
      write(99,*) 'rm *.o*'
   close(99)
   call system('cp job ./' // trim(myDir) )
   call system('rm job')

   open(unit=99,file='bfgs_start.txt')
     write(99,*) '        6       num vars'
     write(99,*) '   R/ks                    sigmak                  kwall                   B00                     B20
weighting '
     write(99,*) '-1.0000000000000E+00   ',sk,kw,initialGuess1,initialGuess2 ,'5.0000000000000E-01'
     write(99,*) '                0                  0                  0                  1                  1                  0'
     write(99,*) '1.0000000000000000E-008    default line search alpha'
     write(99,*) '1.0000000000000000E-005    delta step size used for gradient calculations'
     write(99,*) '        1    differencing scheme (1=central diff, 0=forward diff)'
```

```
    write(99,*) '1.0000000000000000E-012    stop delta'
    write(99,*) '        8      number of simultaneous cases in the line search'

  close(99)
  call system('cp bfgs_start.txt ./' // trim(myDir) )
  call system('rm bfgs_start.txt')

end subroutine
```

IV. Matlab Code

```
function moody()

clc;clear all;format compact;close all

set(0,'DefaultTextFontsize',8, ...
'DefaultTextFontname','Times New Roman', ...
'DefaultTextFontWeight','normal', ...
 'DefaultAxesFontsize',10, ...
 'DefaultAxesFontname','Times New Roman',...
 'DefaultLineLineWidth',0.4)

% ---------------------------------------------------------------------
% COLEBROOK EQUATION
% ---------------------------------------------------------------------

xmax = 9;
xstart = log10(3500);
Ncolebrook = 101;
backgroundColor = [1 1 1]*.2;
symbolWidth = 0.05;

dlog=(xmax-xstart)/real(Ncolebrook);
for i=1:Ncolebrook;
   ReSmooth(i)=10^(xstart+i*dlog);
   DfSmooth(i)=get_Df_Colebrook(0.0,ReSmooth(i));
end


figure;
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .85])


smooth = loglog(ReSmooth,DfSmooth,'color',backgroundColor);
set(get(get(smooth,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');

ylabel('Darcy Friction Factor');
xincr = [10^3 10^4 10^5 10^6 10^7 10^8 10^9];
yincr = [0.01 0.02 0.04 0.06 0.08 0.1];
```

```
xlim([610 2.*10^xmax]);
ylim([0.008 0.1]);
ax1=gca;
set(gca,'ytick',yincr);
set(gca,'xtick',xincr);
xlabel('Bulk Reynolds Number');

hold on

%kr_ = [0.05*2 1/15 1/30.6 1/60 1/126 1/252 1/507 0.0003*2 0.000058*2]/2.;
kr_ = [1/15 1/30.6 1/60 1/126 1/252 1/507 0.0003*2 0.000058*2]/2.;

for j=1:length(kr_);
   for i=1:Ncolebrook;
      Re(i) = 10^(xstart+real(i)*dlog);
      Df(i) = get_Df_Colebrook(kr_(j),Re(i));
   end
   Re(end)=2.*10^9;
   Df(end) = get_Df_Colebrook(kr_(j),Re(end));
   rough = loglog(Re,Df,'color',backgroundColor);
   set(get(get(rough,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');
end



% ------------------------------------------------------------------------
% EXPERIMENTAL DATA
% ------------------------------------------------------------------------

nikuradse15 = load('nikuradse15.txt');
nikuradse30p6 = load('nikuradse30p6.txt');
nikuradse60 = load('nikuradse60.txt');
nikuradse126 = load('nikuradse126.txt');
nikuradse252 = load('nikuradse252.txt');
nikuradse507 = load('nikuradse507.txt');
shockling8621 = load('shockling8621.txt');

loglog(nikuradse15(:,1),nikuradse15(:,2),'o','Markersize',3,'MarkerEdgeColor',backgroundColor,'LineWidth
',symbolWidth);
loglog(nikuradse30p6(:,1),nikuradse30p6(:,2),'^','Markersize',3,'MarkerEdgeColor',backgroundColor,'Line
Width',symbolWidth);
loglog(nikuradse60(:,1),nikuradse60(:,2),'s','Markersize',3,'MarkerEdgeColor',backgroundColor,'LineWidth
',symbolWidth);
loglog(nikuradse126(:,1),nikuradse126(:,2),'v','Markersize',3,'MarkerEdgeColor',backgroundColor,'LineWi
dth',symbolWidth);
loglog(nikuradse252(:,1),nikuradse252(:,2),'d','Markersize',3,'MarkerEdgeColor',backgroundColor,'LineWi
dth',symbolWidth);
loglog(nikuradse507(:,1),nikuradse507(:,2),'p','Markersize',3,'MarkerEdgeColor',backgroundColor,'LineWi
dth',symbolWidth);
loglog(shockling8621(:,1),shockling8621(:,2),'d','Markersize',3,'MarkerEdgeColor',backgroundColor,'Mark
erFaceColor',backgroundColor,'LineWidth',symbolWidth);



% ------------------------------------------------------------------------
```

```
% ADD OTHER DATA
% ------------------------------------------------------------------------
moody = load('moody.txt');

loglog(moody(:,2),moody(:,3),'o','Markersize',3,'MarkerEdgeColor','k','MarkerFaceColor','k');
%loglog(moody(:,2),moody(:,3),'o','Markersize',6,'MarkerEdgeColor','k');
%loglog(moody_dataOLD(:,2),moody_dataOLD(:,3),'o','Markersize',2,'MarkerEdgeColor','k','MarkerFace
Color','k');



% ------------------------------------------------------------------------
% ADD LEGEND
% ------------------------------------------------------------------------

legend_handle = legend('\itR/k_s\rm = 15','\itR/k_s\rm = 30.6','\itR/k_s\rm = 60', ...
   '\itR/k_s\rm = 126','\itR/k_s\rm = 252','\itR/k_s\rm = 507', ...
   '\itR/k_s\rm = 8621','Phillips model','Location','southwest');
legend('boxoff')
set(legend_handle, 'fontsize', 7)




% ------------------------------------------------------------------------
% SAVE PLOT
% ------------------------------------------------------------------------

set(gcf,'paperpositionmode','auto')
saveas(gcf,'moody.eps','epsc')



% ------------------------------------------------------------------------
% FUNCTIONS
% ------------------------------------------------------------------------
   function [Df] = get_Df_Colebrook(kr,Re);

   Df=(-1.8*log10((kr/3.7)^1.11 + 6.9/Re))^(-2);
   R=Df-(-2.0*log10((kr/3.7) + 2.51/(Re*sqrt(Df))))^(-2);

   while(abs(R)>1e-8)
      Df=(-2.0*log10((kr/3.7) + 2.51/(Re*sqrt(Df))))^(-2);
      R=Df-(-2.0*log10((kr/3.7) + 2.51/(Re*sqrt(Df))))^(-2);

   end
   end

end

clc;clear all;close all

data15_fromEmpLambda = load('data15_fromEmpLambda.txt');
data30p6_fromEmpLambda = load('data30p6_fromEmpLambda.txt');
data60_fromEmpLambda = load('data60_fromEmpLambda.txt');
data126_fromEmpLambda = load('data126_fromEmpLambda.txt');
```

```
data252_fromEmpLambda = load('data252_fromEmpLambda.txt');
data507_fromEmpLambda = load('data507_fromEmpLambda.txt');
data1667_fromEmpLambda = load('data1667_fromEmpLambda.txt');
data8621_fromEmpLambda = load('data8621_fromEmpLambda.txt');

backgroundColor=[1 1 1]*0;
yhat = data15_fromEmpLambda(1:2:end,2);

f1 = figure;
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .85])
kappa=0.403;
gamma=0.0324;

Roverks = [15 30.6 60 126 252 507 0.5/0.0003 0.5/0.000058];

nsymbol15 = 161;
nsymbol30 = 158;
nsymbol60 = 155;
nsymbol126 = 152;
nsymbol252 = 149;
nsymbol507 = 145;
nsymbol1667 = 141;
nsymbol8621 = 135;

semilogx(data15_fromEmpLambda(nsymbol15,2),data15_fromEmpLambda(nsymbol15,3),'-
ok','MarkerFaceColor','k');hold on
emp = semilogx(data15_fromEmpLambda(1:2:end,2),data15_fromEmpLambda(1:2:end,3),'-k');
set(get(get(emp,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');
lawWall =
semilogx(yhat,1/kappa*log(yhat*Roverks(1))+1/kappa*log(1/gamma),':','color',backgroundColor,'linewidt
h',1);
set(get(get(lawWall,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');

semilogx(data30p6_fromEmpLambda(nsymbol30,2),data30p6_fromEmpLambda(nsymbol30,3),'-ok');
emp = semilogx(data30p6_fromEmpLambda(1:2:end,2),data30p6_fromEmpLambda(1:2:end,3),'-k');
set(get(get(emp,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');
lawWall =
semilogx(yhat,1/kappa*log(yhat*Roverks(2))+1/kappa*log(1/gamma),':','color',backgroundColor,'linewidt
h',1);
set(get(get(lawWall,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');

semilogx(data60_fromEmpLambda(nsymbol60,2),data60_fromEmpLambda(nsymbol60,3),'-
dk','MarkerFaceColor','k');
emp = semilogx(data60_fromEmpLambda(1:2:end,2),data60_fromEmpLambda(1:2:end,3),'-k');
set(get(get(emp,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');
lawWall =
semilogx(yhat,1/kappa*log(yhat*Roverks(3))+1/kappa*log(1/gamma),':','color',backgroundColor,'linewidt
h',1);
set(get(get(lawWall,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');

semilogx(data126_fromEmpLambda(nsymbol126,2),data126_fromEmpLambda(nsymbol126,3),'-dk');
emp = semilogx(data126_fromEmpLambda(1:2:end,2),data126_fromEmpLambda(1:2:end,3),'-k');
set(get(get(emp,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');
```

```
lawWall =
semilogx(yhat,1/kappa*log(yhat*Roverks(4))+1/kappa*log(1/gamma),':','color',backgroundColor,'linewidt
h',1);
set(get(get(lawWall,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');

semilogx(data252_fromEmpLambda(nsymbol252,2),data252_fromEmpLambda(nsymbol252,3),'-
sk','MarkerFaceColor','k');
emp = semilogx(data252_fromEmpLambda(1:2:end,2),data252_fromEmpLambda(1:2:end,3),'-k');
set(get(get(emp,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');
lawWall =
semilogx(yhat,1/kappa*log(yhat*Roverks(5))+1/kappa*log(1/gamma),':','color',backgroundColor,'linewidt
h',1);
set(get(get(lawWall,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');

semilogx(data507_fromEmpLambda(nsymbol507,2),data507_fromEmpLambda(nsymbol507,3),'-sk');
emp = semilogx(data507_fromEmpLambda(1:2:end,2),data507_fromEmpLambda(1:2:end,3),'-k');
set(get(get(emp,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');
lawWall =
semilogx(yhat,1/kappa*log(yhat*Roverks(6))+1/kappa*log(1/gamma),':','color',backgroundColor,'linewidt
h',1);
set(get(get(lawWall,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');

semilogx(data1667_fromEmpLambda(nsymbol1667,2),data1667_fromEmpLambda(nsymbol1667,3),'-
>k','markersize',8,'MarkerFaceColor','k');
emp = semilogx(data1667_fromEmpLambda(1:2:end,2),data1667_fromEmpLambda(1:2:end,3),'-k');
set(get(get(emp,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');
lawWall =
semilogx(yhat,1/kappa*log(yhat*Roverks(7))+1/kappa*log(1/gamma),':','color',backgroundColor,'linewidt
h',1);
set(get(get(lawWall,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');

semilogx(data8621_fromEmpLambda(nsymbol8621,2),data8621_fromEmpLambda(nsymbol8621,3),'-
>k','markersize',8);
emp = semilogx(data8621_fromEmpLambda(1:2:end,2),data8621_fromEmpLambda(1:2:end,3),'-k');
set(get(get(emp,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');
lawWall =
semilogx(yhat,1/kappa*log(yhat*Roverks(8))+1/kappa*log(1/gamma),':','color',backgroundColor,'linewidt
h',1);
set(get(get(lawWall,'Annotation'),'LegendInformation'),'IconDisplayStyle','off');

%annotation(f1,'textarrow',[0.3 0.4] ,[0.5,0.4],'String','All k_r values','FontSize',10,'TextEdgeColor','none');

%xlim([10^xstart 10^xend])
xlabel('\ity / R')
ylabel('\itV_z / u_\tau')
text(10^-5.57,14.6,'_','rotation',90)
ylim([0 32])
xlim([10^-5.1 1])

legend_handle = legend('\itR/k_s\rm = 15','\itR/k_s\rm = 30.6','\itR/k_s\rm = 60','\itR/k_s\rm = 126'...
    ,'\itR/k_s\rm = 252','\itR/k_s\rm = 507','\itR/k_s\rm = 1667'...
    ,'\itR/k_s\rm = 8621','location','northwest')
legend('boxoff')
set(legend_handle, 'fontsize', 7)
```

```
% ------------------------------------------------------------------------
% SAVE semilogx
% ------------------------------------------------------------------------

set(gcf,'paperpositionmode','auto')
saveas(gcf,'alluplus.eps','epsc')


clc;close all;clear all

data15_fromEmpLambda = load('data15_fromEmpLambda.txt');
data30p6_fromEmpLambda = load('data30p6_fromEmpLambda.txt');
data60_fromEmpLambda = load('data60_fromEmpLambda.txt');
data126_fromEmpLambda = load('data126_fromEmpLambda.txt');
data252_fromEmpLambda = load('data252_fromEmpLambda.txt');
data507_fromEmpLambda = load('data507_fromEmpLambda.txt');
data1667_fromEmpLambda = load('data1667_fromEmpLambda.txt');
data8621_fromEmpLambda = load('data8621_fromEmpLambda.txt');

backgroundColor=[1 1 1]*0;

f1 = figure;
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .85])

Roverks = [15 30.6 60 126 252 507 0.5/0.0003 0.5/0.000058];

xstart=-2;
xend=3;

dx=xend-xstart;
npts=100;
kappa=0.403;
gamma=0.0324;
for i=1:npts
   yoverks(i)=10^(xstart+(i-1)/(npts-1)*dx);
   uplus_emp(i) = 1/kappa*log(yoverks(i))+1/kappa*log(1/gamma);
   uplus_ml_nrWll_fllrgh(i)=1/kappa*log(yoverks(i)/gamma+1);
end


semilogx(yoverks,uplus_emp ,':','color',backgroundColor,'linewidth',1);
hold on
semilogx(yoverks,uplus_ml_nrWll_fllrgh ,'-','color',backgroundColor);

semilogx(data15_fromEmpLambda(:,2)*Roverks(1),data15_fromEmpLambda(:,3),'-k');
semilogx(data30p6_fromEmpLambda(:,2)*Roverks(2),data30p6_fromEmpLambda(:,3),'-k');
semilogx(data60_fromEmpLambda(:,2)*Roverks(3),data60_fromEmpLambda(:,3),'-k');
semilogx(data126_fromEmpLambda(:,2)*Roverks(4),data126_fromEmpLambda(:,3),'-k');
semilogx(data252_fromEmpLambda(:,2)*Roverks(5),data252_fromEmpLambda(:,3),'-k');
semilogx(data507_fromEmpLambda(:,2)*Roverks(6),data507_fromEmpLambda(:,3),'-k');
%semilogx(data1667_fromEmpLambda(:,2)*Roverks(7),data1667_fromEmpLambda(:,3),'-k');
```

```
%semilogx(data8621_fromEmpLambda(:,2)*Roverks(8),data8621_fromEmpLambda(:,3),'-k');

x = [yoverks(npts/2) 1000];
y = [uplus_emp(npts/2) 20];
%annotation(f1,'textarrow',ds2nfu(x, y),ds2nfu(x, y),'String','All k_r
values','FontSize',10,'TextEdgeColor','none');
%annotation(f1,'textarrow',ds2nfu(x, y),'TextEdgeColor','none',...
%    'String',{'here'});


% xPlot=yoverks(npts/2);
% yPlot=uplus_emp(npts/2);
%
% axPos = get(gca,'Position');
% xMinMax = xlim;
% yMinMax = ylim;
% xAnnotation(1) = axPos(1) + ((xPlot - xMinMax(1))/((xMinMax(2)-xMinMax(1))-xMinMax(1))) *
axPos(3)
% yAnnotation(1) = axPos(2) + ((yPlot - yMinMax(1))/((yMinMax(2)-yMinMax(1))-yMinMax(1))) *
axPos(4)
%
% xAnnotation(2) = xAnnotation(1)+0.1;
% yAnnotation(2) = yAnnotation(1)+0;

%annotation(f1,'textarrow',[xAnnotation(1) yAnnotation(1)],[xAnnotation(2) yAnnotation(2)],'String','All
k_r values','FontSize',10,'TextEdgeColor','none');
annotation(f1,'textarrow',[0.3 0.36] ,[0.5,0.32],'String',char({'\itk_r\rm values','in the range','[0.0020-
0.034]'}),'FontSize',10,'TextEdgeColor','none');



xlim([10^xstart 10^xend])
ylim([0 25])
xlabel('\ity / k_s')
ylabel('\itV_z / u_\tau')
text(10^-2.45,11.3,'_','rotation',90)

legend(sprintf('Log Law'),'location','southeast')
legend('boxoff')


% ------------------------------------------------------------------------
% SAVE PLOT
% ------------------------------------------------------------------------

set(gcf,'paperpositionmode','auto')
saveas(gcf,'allkr.eps','epsc')


clc;close all;clear all

Nikuradse = load('Nikuradse.txt');
Phillips = importdata('vRoverks.txt');
```

```
Reichardt = load('Reichardt.txt');
yoverR = [0:0.01:1]';
kappa = 0.404;
gamma = 0.0341;

figure
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .85])

plot(Nikuradse(1:90,1),Nikuradse(1:90,2),'dk','MarkerSize',4);
hold on;
xlabel('\ity / R')
ylabel('\it\nu_t / (u_\tau R)')

plot(Reichardt(:,1),Reichardt(:,2),'>k','MarkerSize',4);



A0=0.345;
A1=0;
a=0;
ksoverR = 1/30.6;

yoverRP69 = [0:0.01:0.17]';
nearWall = kappa*(gamma*ksoverR+(1-gamma/2*ksoverR)*yoverRP69);
plot(yoverRP69,nearWall,'k-.');
plot(Phillips.data(1:201,2),Phillips.data(1:201,7),'k-');
plot(Phillips.data(202:402,2),Phillips.data(202:402,7),'k-');

legend_handle = legend('Nikuradse','Reichardt',...
   'Near-wall fully rough limit','Phillips','Location','Best')
legend('boxoff')
set(legend_handle, 'fontsize', 10)

% ---------------------------------------------------------------------
% SAVE PLOT
% ---------------------------------------------------------------------

set(gcf,'paperpositionmode','auto')
saveas(gcf,'compareNut.eps','epsc')



return


clc;close all;clear all

data = importdata('vksp.txt');

symbolWidth = 0.05;
n= 201;
%r^ y^ u+ k+ w+ lambda^ nu+

% ---------------------------------------------------------------------
```

```
% OMEGA
% ------------------------------------------------------------------------
figure
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .8])
xColumn = 2;
yColumn = 5;
nbegin = 1;
nend = n;
semilogx(data.data(nbegin:nend,xColumn),data.data(nbegin:nend,yColumn),'-
k','MarkerSize',2,'markerfacecolor','k','LineWidth',symbolWidth);
hold on;
semilogx(data.data(nbegin+n:nend+n,xColumn),data.data(nbegin+n:nend+n,yColumn),':k','MarkerSize',5,'
LineWidth',2);
semilogx(data.data(nbegin+2*n:nend+2*n,xColumn),data.data(nbegin+2*n:nend+2*n,yColumn),'k-
.','MarkerSize',7);
xlabel('\ity/R')
ylabel('\it\omegaR/u_\tau')

legend_handle = legend('\itk_s^+ = \rm80,000','\itk_s^+ =\rm 1000','\itk_s^+ =
\rm100','location','southeast');
% legend_handle = legend('k^+_{wall} = 1.0','k^+_{wall} = 0.5','k^+_{wall} = 0.1','k^+_{wall} =
0.05','location','northeast')
%     '\itR/k_s\rm = 126','\itR/k_s\rm = 126','\itR/k_s\rm = 507',...
%     char({'Near-wall fully','rough limit'}),char({'Fully rough','empirical fit'}),...
%     'Location','SouthEast')
legend('boxoff')
set(legend_handle, 'fontsize', 10)


set(gcf,'paperpositionmode','auto')
saveas(gcf,'omega_vksplog.eps','epsc')


% ------------------------------------------------------------------------
% k
% ------------------------------------------------------------------------
figure
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .8])
xColumn = 2;
yColumn = 4;
nbegin = 1;
nend = n;
plot(data.data(nbegin:nend,xColumn),data.data(nbegin:nend,yColumn),'-
k','MarkerSize',2,'markerfacecolor','k','LineWidth',symbolWidth);
hold on;
plot(data.data(nbegin+n:nend+n,xColumn),data.data(nbegin+n:nend+n,yColumn),':k','MarkerSize',5,'Line
Width',2);
plot(data.data(nbegin+2*n:nend+2*n,xColumn),data.data(nbegin+2*n:nend+2*n,yColumn),'k-
.','MarkerSize',7);
xlabel('\ity/R')
ylabel('\itk/u_\tau^2')
```

```
legend_handle = legend('\itk_s^+ = \rm80,000','\itk_s^+ =\rm 1000','\itk_s^+ = \rm100','location','southeast');
% legend_handle = legend('k^+_{wall} = 1.0','k^+_{wall} = 0.5','k^+_{wall} = 0.1','k^+_{wall} = 0.05','location','northeast')
%    '\itR/k_s\rm = 126','\itR/k_s\rm = 126','\itR/k_s\rm = 507',...
%    char({'Near-wall fully','rough limit'}),char({'Fully rough','empirical fit'}),...
%    'Location','SouthEast')
legend('boxoff')
set(legend_handle, 'fontsize', 10)


set(gcf,'paperpositionmode','auto')
saveas(gcf,'k_vksp.eps','epsc')


% ---------------------------------------------------------------------
% OMEGA
% ---------------------------------------------------------------------
figure
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .8])
xColumn = 2;
yColumn = 5;
nbegin = 1;
nend = n;
plot(data.data(nbegin:nend,xColumn),data.data(nbegin:nend,yColumn),'-k','MarkerSize',2,'markerfacecolor','k','LineWidth',symbolWidth);
hold on;
plot(data.data(nbegin+n:nend+n,xColumn),data.data(nbegin+n:nend+n,yColumn),'ok','MarkerSize',2,'LineWidth',symbolWidth,'markerfacecolor','k');
plot(data.data(nbegin+2*n:nend+2*n,xColumn),data.data(nbegin+2*n:nend+2*n,yColumn),'ko','MarkerSize',7);
xlabel('\ity/R')
ylabel('\it\omegaR/u_\tau')
ylim([0 1000])
legend_handle = legend('\itk_s^+ = \rm80,000','\itk_s^+ =\rm 1000','\itk_s^+ = \rm100','location','northeast');
% legend_handle = legend('k^+_{wall} = 1.0','k^+_{wall} = 0.5','k^+_{wall} = 0.1','k^+_{wall} = 0.05','location','northeast')
%    '\itR/k_s\rm = 126','\itR/k_s\rm = 126','\itR/k_s\rm = 507',...
%    char({'Near-wall fully','rough limit'}),char({'Fully rough','empirical fit'}),...
%    'Location','SouthEast')
legend('boxoff')
set(legend_handle, 'fontsize', 10)


set(gcf,'paperpositionmode','auto')
saveas(gcf,'omega_vksp.eps','epsc')


% ---------------------------------------------------------------------
% lambda
% ---------------------------------------------------------------------
figure
```

```
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .8])
xColumn = 2;
yColumn = 6;
nbegin = 1;
nend = n;
plot(data.data(nbegin:nend,xColumn),data.data(nbegin:nend,yColumn),'-
k','MarkerSize',2,'markerfacecolor','k','LineWidth',symbolWidth);
hold on;
plot(data.data(nbegin+n:nend+n,xColumn),data.data(nbegin+n:nend+n,yColumn),'ok','MarkerSize',2,'Line
Width',symbolWidth,'markerfacecolor','k');
plot(data.data(nbegin+2*n:nend+2*n,xColumn),data.data(nbegin+2*n:nend+2*n,yColumn),'ko','MarkerSiz
e',7);
xlabel('\ity/R')
ylabel('\it\lambda/R')

legend_handle = legend('\itk_s^+ = \rm80,000','\itk_s^+ =\rm 1000','\itk_s^+ =
\rm100','location','southeast');
% legend_handle = legend('k^+_{wall} = 1.0','k^+_{wall} = 0.5','k^+_{wall} = 0.1','k^+_{wall} =
0.05','location','northeast')
%    '\itR/k_s\rm = 126','\itR/k_s\rm = 126','\itR/k_s\rm = 507',...
%    char({'Near-wall fully','rough limit'}),char({'Fully rough','empirical fit'}),...
%    'Location','SouthEast')
legend('boxoff')
set(legend_handle, 'fontsize', 10)


set(gcf,'paperpositionmode','auto')
saveas(gcf,'lambda_vksp.eps','epsc')


% ------------------------------------------------------------------------
% nu
% ------------------------------------------------------------------------
figure
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .8])
xColumn = 2;
yColumn = 7;
nbegin = 1;
nend = n;
plot(data.data(nbegin:nend,xColumn),data.data(nbegin:nend,yColumn),'-
k','MarkerSize',2,'markerfacecolor','k','LineWidth',symbolWidth);
hold on;
plot(data.data(nbegin+n:nend+n,xColumn),data.data(nbegin+n:nend+n,yColumn),'k:','MarkerSize',4,'Line
Width',2);
plot(data.data(nbegin+2*n:nend+2*n,xColumn),data.data(nbegin+2*n:nend+2*n,yColumn),'k-
.','MarkerSize',7);
xlabel('\ity/R')
ylabel('\it\nu/(u_\tau R)')

legend_handle = legend('\itk_s^+ = \rm80,000','\itk_s^+ =\rm 1000','\itk_s^+ =
\rm100','location','southeast');
% legend_handle = legend('k^+_{wall} = 1.0','k^+_{wall} = 0.5','k^+_{wall} = 0.1','k^+_{wall} =
0.05','location','northeast')
```

```
%     '\itR/k_s\rm = 126','\itR/k_s\rm = 126','\itR/k_s\rm = 507',...
%     char({'Near-wall fully','rough limit'}),char({'Fully rough','empirical fit'}),...
%     'Location','SouthEast')
legend('boxoff')
set(legend_handle, 'fontsize', 10)


set(gcf,'paperpositionmode','auto')
saveas(gcf,'nut_vksp.eps','epsc')

return


clc;close all;clear all

data = importdata('vkwall.txt');

symbolWidth = 0.05;
n= 201;
%r^ y^ u+ k+ w+ lambda^ nu+

% ------------------------------------------------------------------------
% kwall
% ------------------------------------------------------------------------
figure
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .8])
xColumn = 2;
yColumn = 4;
nbegin = 1;
nend = n;
semilogx(data.data(nbegin:nend,xColumn),data.data(nbegin:nend,yColumn),'-
k','MarkerSize',2,'markerfacecolor','k','LineWidth',symbolWidth);
hold on;
semilogx(data.data(nbegin+n:nend+n,xColumn),data.data(nbegin+n:nend+n,yColumn),':k','MarkerSize',5,'
LineWidth',2);
semilogx(data.data(nbegin+2*n:nend+2*n,xColumn),data.data(nbegin+2*n:nend+2*n,yColumn),'k-
.','MarkerSize',7);
xlabel('\ity/R')
ylabel('\itk/u_\tau^2')
xlim([10^-5 1])
legend_handle = legend('\itk^+_{\rmwall} \rm= 0.05','\itk^+_{\rmwall} \rm= 0.1','\itk^+_{\rmwall} \rm=
1.0','location','southeast');
% legend_handle = legend('k^+_{wall} = 1.0','k^+_{wall} = 0.5','k^+_{wall} = 0.1','k^+_{wall} =
0.05','location','northeast')
%     '\itR/k_s\rm = 126','\itR/k_s\rm = 126','\itR/k_s\rm = 507',...
%     char({'Near-wall fully','rough limit'}),char({'Fully rough','empirical fit'}),...
%     'Location','SouthEast')
legend('boxoff')
set(legend_handle, 'fontsize', 10)


set(gcf,'paperpositionmode','auto')
saveas(gcf,'k_vkwalllog.eps','epsc')
```

```
% ------------------------------------------------------------------------
% k
% ------------------------------------------------------------------------
figure
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .8])
xColumn = 2;
yColumn = 4;
nbegin = 1;
nend = n;
plot(data.data(nbegin:nend,xColumn),data.data(nbegin:nend,yColumn),'-
k','MarkerSize',2,'markerfacecolor','k','LineWidth',symbolWidth);
hold on;
plot(data.data(nbegin+n:nend+n,xColumn),data.data(nbegin+n:nend+n,yColumn),'ok','MarkerSize',2,'Line
Width',symbolWidth,'markerfacecolor','k');
plot(data.data(nbegin+2*n:nend+2*n,xColumn),data.data(nbegin+2*n:nend+2*n,yColumn),'ko','MarkerSiz
e',7);
xlabel('\ity/R')
ylabel('\itk/u_\tau^2')

legend_handle = legend('\itk^+_{\rmwall} \rm= 0.05','\itk^+_{\rmwall} \rm= 0.1','\itk^+_{\rmwall} \rm=
1.0','location','southeast');
% legend_handle = legend('k^+_{wall} = 1.0','k^+_{wall} = 0.5','k^+_{wall} = 0.1','k^+_{wall} =
0.05','location','northeast')
%     '\itR/k_s\rm = 126','\itR/k_s\rm = 126','\itR/k_s\rm = 507',...
%     char({'Near-wall fully','rough limit'}),char({'Fully rough','empirical fit'}),...
%     'Location','SouthEast')
legend('boxoff')
set(legend_handle, 'fontsize', 10)


set(gcf,'paperpositionmode','auto')
saveas(gcf,'k_vkwall.eps','epsc')


% ------------------------------------------------------------------------
% OMEGA
% ------------------------------------------------------------------------
figure
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .8])
xColumn = 2;
yColumn = 5;
nbegin = 1;
nend = n;

plot(data.data(nbegin:nend,xColumn),data.data(nbegin:nend,yColumn),'-
k','MarkerSize',2,'markerfacecolor','k','LineWidth',symbolWidth);
hold on;
plot(data.data(nbegin+n:nend+n,xColumn),data.data(nbegin+n:nend+n,yColumn),'ok','MarkerSize',2,'Line
Width',symbolWidth,'markerfacecolor','k');
```

```
plot(data.data(nbegin+2*n:nend+2*n,xColumn),data.data(nbegin+2*n:nend+2*n,yColumn),'ko','MarkerSiz
e',7);
xlabel('\ity/R')
ylabel('\it\omegaR/u_\tau')
ylim([0 1000])
legend_handle = legend('\itk^+_{\rmwall} \rm= 0.05','\itk^+_{\rmwall} \rm= 0.1','\itk^+_{\rmwall} \rm=
1.0','location','northeast');
% legend_handle = legend('k^+_{wall} = 1.0','k^+_{wall} = 0.5','k^+_{wall} = 0.1','k^+_{wall} =
0.05','location','northeast')
%     '\itR/k_s\rm = 126','\itR/k_s\rm = 126','\itR/k_s\rm = 507',...
%     char({'Near-wall fully','rough limit'}),char({'Fully rough','empirical fit'}),...
%     'Location','SouthEast')
legend('boxoff')
set(legend_handle, 'fontsize', 10)


set(gcf,'paperpositionmode','auto')
saveas(gcf,'omega_vkwall.eps','epsc')


% ------------------------------------------------------------------------
% lambda
% ------------------------------------------------------------------------
figure
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .8])
xColumn = 2;
yColumn = 6;
nbegin = 1;
nend = n;
plot(data.data(nbegin:nend,xColumn),data.data(nbegin:nend,yColumn),'-
k','MarkerSize',2,'markerfacecolor','k','LineWidth',symbolWidth);
hold on;
plot(data.data(nbegin+n:nend+n,xColumn),data.data(nbegin+n:nend+n,yColumn),'ok','MarkerSize',2,'Line
Width',symbolWidth,'markerfacecolor','k');
plot(data.data(nbegin+2*n:nend+2*n,xColumn),data.data(nbegin+2*n:nend+2*n,yColumn),'ko','MarkerSiz
e',7);
xlabel('\ity/R')
ylabel('\it\lambda/R')

legend_handle = legend('\itk^+_{\rmwall} \rm= 0.05','\itk^+_{\rmwall} \rm= 0.1','\itk^+_{\rmwall} \rm=
1.0','location','southeast');
% legend_handle = legend('k^+_{wall} = 1.0','k^+_{wall} = 0.5','k^+_{wall} = 0.1','k^+_{wall} =
0.05','location','northeast')
%     '\itR/k_s\rm = 126','\itR/k_s\rm = 126','\itR/k_s\rm = 507',...
%     char({'Near-wall fully','rough limit'}),char({'Fully rough','empirical fit'}),...
%     'Location','SouthEast')
legend('boxoff')
set(legend_handle, 'fontsize', 10)


set(gcf,'paperpositionmode','auto')
saveas(gcf,'lambda_vkwall.eps','epsc')
```

```
% ------------------------------------------------------------------------
% nu
% ------------------------------------------------------------------------
figure
set(gcf, 'Units','inches','Position', [2 2 6.5 4])
set(gca,'fontsize',10,'position',[0.1 0.12 .88 .8])
xColumn = 2;
yColumn = 7;
nbegin = 1;
nend = n;
plot(data.data(nbegin:nend,xColumn),data.data(nbegin:nend,yColumn),'-
k','MarkerSize',2,'markerfacecolor','k','LineWidth',symbolWidth);
hold on;
plot(data.data(nbegin+n:nend+n,xColumn),data.data(nbegin+n:nend+n,yColumn),'ok','MarkerSize',2,'Line
Width',symbolWidth,'markerfacecolor','k');
plot(data.data(nbegin+2*n:nend+2*n,xColumn),data.data(nbegin+2*n:nend+2*n,yColumn),'ko','MarkerSiz
e',7);
xlabel('\ity/R')
ylabel('\it\nu/(u_\tau R)')

legend_handle = legend('\itk^+_{\rmwall} \rm= 0.05','\itk^+_{\rmwall} \rm= 0.1','\itk^+_{\rmwall} \rm=
1.0','location','southeast');
% legend_handle = legend('k^+_{wall} = 1.0','k^+_{wall} = 0.5','k^+_{wall} = 0.1','k^+_{wall} =
0.05','location','northeast')
%     '\itR/k_s\rm = 126','\itR/k_s\rm = 126','\itR/k_s\rm = 507',...
%     char({'Near-wall fully','rough limit'}),char({'Fully rough','empirical fit'}),...
%     'Location','SouthEast')
legend('boxoff')
set(legend_handle, 'fontsize', 10)


set(gcf,'paperpositionmode','auto')
saveas(gcf,'nut_vkwall.eps','epsc')

return
```