11-23-93
E8231

NASA Technical Memorandum 106399
AIAA–93–2964

Some Practical Turbulence Modeling Options for Reynolds-Averaged Full Navier-Stokes Calculations of Three-Dimensional Flows

Trong T. Bui
Lewis Research Center
Cleveland, Ohio

Prepared for the
24th AIAA Fluid Dynamics Conference
sponsored by the American Institute of Aeronautics and Astronautics
Orlando, Florida, July 6–9, 1993

NASA

Some Practical Turbulence Modeling Options for Reynolds-Averaged Full Navier-Stokes Calculations of Three-Dimensional Flows

Trong T. Bui*
NASA Lewis Research Center
Cleveland, Ohio

ABSTRACT

New turbulence modeling options recently implemented for the 3-D version of Proteus, a Reynolds-averaged compressible Navier-Stokes code, are described. The implemented turbulence models include: the Baldwin-Lomax algebraic model, the Baldwin-Barth one-equation model, the Chien k-ε model, and the Launder-Sharma k-ε model. Features of this turbulence modeling package include: well documented and easy to use turbulence modeling options, uniform integration of turbulence models from different classes, automatic initialization of turbulence variables for calculations using one- or two-equation turbulence models, multiple solid boundaries treatment, and fully vectorized L-U solver for one- and two-equation models. Validation test cases include the incompressible and compressible flat plate turbulent boundary layers, turbulent developing S-duct flow, and glancing shock wave/turbulent boundary layer interaction. Good agreement is obtained between the computational results and experimental data. Sensitivity of the compressible turbulent solutions with the method of y^+ computation, the turbulent length scale correction, and some compressibility corrections are examined in detail. The test cases show that the highly optimized one- and two-equation turbulence models can be used in routine 3-D Navier-Stokes computations with no significant increase in CPU time as compared with the Baldwin-Lomax algebraic model.

INTRODUCTION

Rapid advancements in computer technology and numerical algorithms have made it possible to routinely perform 3-D Reynolds-averaged Navier-Stokes (RANS) calculations for a number of practical turbulent flow problems. At NASA Lewis, a computer code called Proteus has been developed to solve the Reynolds-averaged, unsteady compressible Navier-Stokes equations. This computer code has been developed with an emphasis on code readability, modularity, and documentations. Proteus is available in the 2-D/axisymmetric[1, 2] and 3-D[3] versions. Both versions have previously been released with the Baldwin-Lomax[4] algebraic model (BL) and the Chien[5] k-ε model (CH). The recent implementation effort has added the Baldwin-Barth[6] one-equation model (BB), the Launder-Sharma[7] k-ε model (LS), and an automatic starting routine for turbulent calculations to the 3-D version of Proteus. In addition, the compressibility corrections of Zhang et al.[8], Sarkar[9], Zeman-free shear[9], Zeman-wall bounded[9], and Wilcox[9], as well as a turbulent length scale correction, similar to the ones proposed by Vuong and Coakley[10] and described by Horstman[11], are available for any of the k-ε models above.

Proteus is a general purpose RANS code that makes no prior assumptions on the type or geometry of the fluid dynamics problem to be solved. Ideally, it is desirable to have a single universal turbulence model in Proteus that can handle every turbulent flow problem in that code's domain of applicability. However, none of the present turbulence models is universal, and the next best thing is to have a number of different well proven turbulence models available in Proteus as turbulence modeling options. Then, the Proteus code users can just try out the various turbulence models available and select those that work best for their particular applications.

In this implementation effort, there are four major objectives:

- The turbulence models should complement each other in terms of applicability, robustness, and accuracy so that together, they can cover the widest possible range of turbulent analysis that can be done.
- Although the formulations of the models can vary widely between different classes of models, the implementation of these models should nevertheless be uniform, highly integrated, and efficient.
- The turbulence modeling options need to be modular, easy to modify, easy to use, and well documented
- The implemented models should be validated as much as possible using standard benchmark test cases.

---

*Aerospace Engineer, Inlet, Duct, & Nozzle Flow Physics Branch, Internal Fluid Mechanics Division.
Member AIAA.

In this paper, the approaches used in the current work to meet the above objectives will be discussed. The new Proteus turbulence modeling capability and its features will be described. Comparisons will be made between the experimental data and the computational results using different turbulence models for the test cases of incompressible and compressible flat plate turbulent boundary layers, turbulent developing S-duct flow, and glancing shock wave/turbulent boundary layer interaction. Sensitivity of the compressible turbulent solutions with y^+ computation, turbulent length scale correction, and compressibility corrections will be examined. Finally, details in using the various turbulence options in 3-D Proteus computations will be given.

TURBULENCE MODELS

The four turbulence models used in this implementation effort can be organized into 3 major classes: algebraic, one-equation, and two-equation models. While all of these turbulence models are well known, the BL algebraic model is perhaps the most popular turbulence model for RANS computations. It is well proven, and considerable experience in using it has developed over the years. The BB one-equation model has recently been developed, and it has successfully been applied in a variety of 3-D turbulent calculations[12-14]. The CH and the LS are well proven low-Reynolds number k-ε models, and they have also been used in a number of implicit as well as explicit Navier-Stokes codes[15-21]. These low-Reynolds number k-ε models are applicable all the way to the solid wall with no additional wall treatments required.

The LS k-ε model is the only turbulence model in this group that does not require y^+ in its formulation. This is desirable, since in a generalized 3-D full RANS analysis with multiple solid boundaries, it is difficult to compute y^+.

Generally, using more sophisticated turbulence models requires more work and turbulence modeling expertise from the code user. Using one- and two-equation turbulence models, for example, will require initial conditions and boundary conditions for the turbulence variables in addition to those for the mean flow variables. Also, the stability requirements of the numerical methods used to solve the turbulence equations will need to be considered together with the mean flow equations. However, careful implementation of these models can significantly reduce the amount of work and frustration for the code user, and many features in the current turbulence modeling package have been implemented for this reason.

Upwind differencing for the turbulence equations is used for both numerical stability and ease of use. Automatic initialization of turbulence variables is available to simplify the task of starting calculations with one- and two-equation models. A minimal amount of information is required from the user to start a turbulent calculation with the Proteus code. The only inputs required for a turbulent calculation are the turbulence model type and the boundary conditions for the turbulence variables (if a one- or two-equation turbulence model was used). User inputs for other turbulence parameters or constants are optional.

Algebraic Model

The BL algebraic model is the first turbulence model implemented into the Proteus code. It is available in both the 2-D/axisymmetric and 3-D Proteus versions. Besides its primary use as a turbulence model, it is also used as part of the automatic initialization procedure for computations using one- and two-equation turbulence models.

Although using the BL algebraic model to compute complex 3-D turbulent flows can be questionable, especially for separated flows and for flows with multiple shear layers and/or multiple solid boundaries, this model is nevertheless simple, fast, and versatile. Most importantly, it gives reasonably good results in all of the test cases considered in this report. In this model, the formula for the turbulent viscosity is:

$$\mu_t = \{ \begin{matrix} (\mu_t)_{inner} & (y_n \le y_{crossover}) \\ (\mu_t)_{outer} & (y_n > y_{crossover}) \end{matrix} \qquad (1)$$

Where y_n is the normal distance from the wall and $y_{crossover}$ is the smallest value of y_n at which the inner and the outer values of turbulent viscosity are equal.

The inner turbulent viscosity is defined as

$$(\mu_t)_{inner} = \rho l^2 |\omega| Re_r \qquad (2)$$

with

$$l = \kappa y_n [1 - e^{-y_n^+/A^+}] \qquad (3)$$

$|\omega|$ is the magnitude of the vorticity and

$$y_n^+ = \left(\frac{y_n}{\nu_1}\right) \sqrt{\frac{\tau_w Re_r}{\rho_1}} \qquad (4)$$

where

ρ_1 can either be ρ_w or ρ_L

ν_1 can either be ν_w or ν_L

The subscripts w and L in the above terms denote wall and local conditions, respectively. The reason for the factor Re_r in eq. (2) and eq. (4) and in the equations that follow in this report is that all of the variables have been nondimensionalized in Proteus. τ_w is computed as the product of the fluid molecular viscosity at the wall and either the normal gradient of the tangential velocity at the wall or $|\omega|$ at the wall. $|\omega|$ is slightly cheaper to compute than the normal gradient of the tangential velocity, so it is used in most of the computations. For 2-D boundary layer flows, y_n^+ computed using $|\omega|$ is the same as using the normal gradient of the tangential velocity. For other flows considered in this report, the error in y_n^+ computed from $|\omega|$ is small and did not significantly affect the final solutions.

Options are available in Proteus to use the Launder-Priddin length scale modification for eq. (3) and to use the Spalding and Kleinstein inner turbulent viscosity formula instead of eq. (2), but none of these was used in the present investigation.

The outer turbulent viscosity is defined as

$$(\mu_t)_{outer} = KC_{cp}\rho F_{wake}F_{Kleb}Re_r \tag{5}$$

with

$$F_{wake} = y_{max}F_{max} \tag{6}$$

F_{max} in eq. (6) is the maximum value of

$$F(y_n) = y|\omega|\left(1 - e^{-y_n^+/A^+}\right) \tag{7}$$

and y_{max} is the value of y_n corresponding to F_{max}. Note that eq. (6) and eq. (7) only apply to wall bounded flows.

As recommended by Degani and Schiff[22], F_{max} is taken to be the first peak of $F(y_n)$ when searched from the solid wall. To prevent the selection of spurious peaks of $F(y_n)$ near the wall, a peak is considered to have been found when the value of $F(y_n)$ drops below FPMIN*F_{max}, where FPMIN is a user adjustable factor. The default value for FPMIN in Proteus is 0.9.

The function F_{Kleb} in eq. (5) is the Klebanoff intermittency factor. The modified F_{Kleb} formula used in Proteus is given by:

$$F_{Kleb} = C_{k,min} + (1-C_{k,min})\left[1 + B\left(\frac{C_{Kleb}y_n}{y_{max}}\right)^6\right]^{-1} \tag{8}$$

$C_{k,min}$ is a user adjustable constant normally. When this constant is set to zero, eq. (8) becomes the usual Klebanoff expression. However, when the BL model is used to generate the initial turbulent viscosity field to start the k-ε calculations, $C_{k,min}$ is set to be around 0.1. This will prevent the turbulent viscosity values from becoming too small in the uniform freestream regions and help minimize starting problems with the k-ε computations. The constants used are:

$A^+ = 26.0$; $C_{cp} = 1.6$; $C_{Kleb} = 0.3$;
$C_{wk} = 1.0$; $K = 0.0168$; $\kappa = 0.4$

In Proteus, modifications to the BL model were made so that it can also be used for turbulent free shear flows. These modifications are given in detail in the Proteus manual sets[1-3], and they will not be described here, since all of the test cases considered in this investigation involve only wall bounded flows.

One-Equation Model

The BB one-equation model is a recently developed model for the turbulence quantity k^2/ε that avoids the need for an algebraic turbulent length scale. Also, this model has less numerical difficulties than the k-ε models. Experiences with the BB model in Proteus calculations showed that it is very similar to the BL model in terms of speed, robustness, and grid point clustering requirement.

The actual implementation of this model uses the following nondimensionalized equations:

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = \mathbf{S} \tag{9}$$

where

$$\mathbf{w} = \rho a \tag{10}$$

$$\mathbf{F} = \rho u a - \frac{1}{Re_r}\mu_a\frac{\partial a}{\partial x} \tag{11}$$

$$\mathbf{G} = \rho v a - \frac{1}{Re_r}\mu_a\frac{\partial a}{\partial y} \tag{12}$$

$$\mathbf{H} = \rho w a - \frac{1}{Re_r}\mu_a\frac{\partial a}{\partial z} \tag{13}$$

$$\mathbf{S} = (c_{\varepsilon 2}f_2 - c_{\varepsilon 1})\frac{G_1}{B_1}\sqrt{\rho a P_k} \tag{14}$$

$$\mu_a = \frac{1}{B_1}\left[\left(\left(\mu + \frac{\mu_t}{\sigma_r}\right)G_1 + \frac{\mu_t}{\sigma_\varepsilon}\right)_{i,j,k} - \frac{\mu_t}{\sigma_\varepsilon}\right] \tag{15}$$

$$a = \frac{k^2}{\varepsilon} \tag{16}$$

$$\frac{1}{\sigma_\varepsilon} = (c_{\varepsilon 2} - c_{\varepsilon 1})\frac{\sqrt{c_\mu}}{\kappa^2} \tag{17}$$

$$\sigma_r = 0.25\sigma_\varepsilon \tag{18}$$

$$\mu_t = c_\mu \rho a D_1 D_2 \tag{19}$$

$$D_1 = 1 - e^{-y_n^+/A^+} \tag{20}$$

$$D_2 = 1 - e^{-y_n^+/A_2^+} \tag{21}$$

$$P_k = \mu_t P_1 \tag{22}$$

$$P_1 = 2\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2 \right] - \frac{2}{3}(P_2)^2 + \left[\frac{\partial u}{\partial y} + \frac{\partial u}{\partial z} + \frac{\partial v}{\partial x} + \frac{\partial v}{\partial z} + \frac{\partial w}{\partial x} + \frac{\partial w}{\partial y}\right]^2 \tag{23}$$

$$P_2 = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \tag{24}$$

$$f_2 = \frac{c_{\varepsilon 1}}{c_{\varepsilon 2}} + \left(1 - \frac{c_{\varepsilon 1}}{c_{\varepsilon 2}}\right)\left(\frac{1}{\kappa y^+} + D_1 D_2\right) \times \left(\sqrt{D_1 D_2} + \frac{y_n^+}{\sqrt{D_1 D_2}}\left(\frac{1}{A^+} e^{-y_n^+/A^+}\right) D_2 + \frac{D_1}{A^+{}_2} e^{-y_n^+/A_2^+}\right) \tag{25}$$

$$G_1 = 1 + \frac{F_r^4 + \varepsilon_0/B_2}{\omega^4 + B_2 F_r^4 + \varepsilon_0} \tag{26}$$

$$F_r = \frac{1}{Re_r}\frac{\nabla a \bullet \nabla a}{a} - \frac{\kappa^2}{c_\mu}|\omega| \tag{27}$$

$\kappa = 0.41$; $c_{\varepsilon 1} = 1.2$; $c_{\varepsilon 2} = 2.0$;
$c_\mu = 0.09$; $A^+ = 26$; $A_2^+ = 10$;
$B_1 = 0.4$; $B_2 = 0.01$; $\varepsilon_0 = 10^{-10}$.

In eq. (15), the term with the subscript (i,j,k) is held constant at a point (i,j,k) and not differenced like the rest of μ_a. That term is actually outside of the partial derivatives, but it is included inside μ_a for programming convenience. The turbulence variable $a = k^2/\varepsilon$ is nondimensionalized by ν_r, a reference kinematic viscosity value, and the other mean flow variables are nondimensionalized using standard reference values as described in the Proteus manuals.

Two-Equation Models

The CH and LS models are two-equation low-Reynolds number k-ε models. They are well proven, and a lot of experiences have been gained on the use of these two model over the years. Both the CH and LS models use the transport equations for the turbulent kinetic energy k and the isotropic dissipation rate ε. Other k-ε models use the transport equation for the total dissipation rate ε_t[23-25], but the wall boundary condition for the isotropic dissipation rate ε is simpler ($\varepsilon_w = 0$), and numerical experiments during this investigation have found that the CH and the LS models are more forgiving numerically than the k-ε models using ε_t, especially in the near wall region.

The LS k-ε model does not require y_n^+, and this is an important advantage for complex 3-D flows with multiple solid boundaries. On the other hand, the near wall formulation of the CH model is simpler and more robust numerically. The nondimensionalized k-ε equation in vector notation is the same as eq. (9), where

$$\mathbf{w} = \begin{bmatrix} \rho k \\ \rho \varepsilon \end{bmatrix} \tag{28}$$

$$\mathbf{F} = \begin{bmatrix} \rho u k - \frac{1}{Re_r}\mu_k \frac{\partial k}{\partial x} \\ \rho u \varepsilon - \frac{1}{Re_r}\mu_\varepsilon \frac{\partial \varepsilon}{\partial x} \end{bmatrix} \tag{29}$$

$$\mathbf{G} = \begin{bmatrix} \rho v k - \frac{1}{Re_r}\mu_k \frac{\partial k}{\partial y} \\ \rho v \varepsilon - \frac{1}{Re_r}\mu_\varepsilon \frac{\partial \varepsilon}{\partial y} \end{bmatrix} \tag{30}$$

$$\mathbf{H} = \begin{bmatrix} \rho w k - \frac{1}{Re_r}\mu_k \frac{\partial k}{\partial z} \\ \rho w \varepsilon - \frac{1}{Re_r}\mu_\varepsilon \frac{\partial \varepsilon}{\partial z} \end{bmatrix} \tag{31}$$

$$\mathbf{S} = \begin{bmatrix} P_k - Re_r \rho \varepsilon + D + C_k \\ c_1 P_k \frac{\varepsilon}{k} - Re_r c_2 f_2 \rho \frac{\varepsilon^2}{k} + E + C_\varepsilon \end{bmatrix} \tag{32}$$

$$\mu_k = \mu + \frac{\mu_t}{\sigma_k} \tag{33}$$

$$\mu_\varepsilon = \mu + \frac{\mu_t}{\sigma_\varepsilon} \tag{34}$$

$$\mu_t = c_\mu f_\mu \rho \frac{k^2}{\varepsilon} \tag{35}$$

$$P_k = \frac{1}{Re_r}\mu_t P_1 - \frac{2}{3}\rho k P_2 \tag{36}$$

where P_1 and P_2 are given by eq. (23) and eq. (24), respectively.

For the CH model, the following damping functions and constants apply:

$$f_\mu = 1 - e^{-0.0115y^+} \tag{37}$$

$$f_2 = 1 - \frac{2}{9}e^{-Re_t^2/36} \tag{38}$$

where

$$Re_t = \frac{\rho k^2}{\mu\varepsilon} \tag{39}$$

$c_1 = 1.35$; $c_2 = 1.80$; $c_\mu = 0.09$;
$\sigma_k = 1.00$; $\sigma_\varepsilon = 1.30$

For the LS model, the following damping functions and constants apply:

$$f_\mu = \exp\left[\frac{-3.4}{(1 + Re_t/50)^2}\right] \tag{40}$$

$$f_2 = 1 - 0.3e^{-Re_t^2} \tag{41}$$

Where Re_t is defined in eq. (39), and

$c_1 = 1.44$; $c_2 = 1.92$; $c_\mu = 0.09$
$\sigma_k = 1.00$; $\sigma_\varepsilon = 1.30$

D and E in eq. (32) are the extra near wall modifications used by the CH and the LS models, and they are given by:

$$D = \begin{cases} -\dfrac{2\mu k}{Re_r y_n^2} & \text{(CH)} \\ -\dfrac{2\mu}{Re_r}\left(\dfrac{\partial}{\partial x_i}\sqrt{k}\right)\left(\dfrac{\partial}{\partial x_i}\sqrt{k}\right) & \text{(LS)} \end{cases} \tag{42}$$

$$E = \begin{cases} -\dfrac{2\mu e^{-y^+/2}}{Re_r y_n^2}\varepsilon & \text{(CH)} \\ \dfrac{2\mu\mu_t}{Re_r^3\rho}\left(\dfrac{\partial^2 u_i}{\partial x_j \partial x_l}\right)\left(\dfrac{\partial^2 u_i}{\partial x_j \partial x_l}\right) & \text{(LS)} \end{cases} \tag{43}$$

The term D is present because the isotropic dissipation rate ε is used. According to Jones and Launder[26], term E is present to improve the computed peak level of turbulence kinetic energy at $y^+ \approx 20$. The turbulence variables k and ε have been nondimensionalized by u_r^2 and $\rho_r u_r^4/\mu_r$, respectively.

C_k and C_ε in eq. (32) are the compressibility corrections, and the definitions for those terms will depend on the particular compressibility correction used.

Five compressibility correction options for the k-ε models were examined: (1) a general correction due to Zhang et al.[8] (ZH), (2) the Sarkar correction (SA), (3) the Zeman correction for free shear flow (ZF), (4) the Zeman correction for wall bounded flow (ZW), and (5) the Wilcox correction (WI). Options number 2-5 were implemented as described by Wilcox[9].

The Zhang et al. generalized compressibility correction was obtained by including the Sarkar correction and a number of other models for the exact compressible terms in the k-ε model equations. For this implementation, these terms are:

$$\begin{aligned} C_k = & -P_k\alpha_1 M_t^2 - Re_r\rho\varepsilon_t(\alpha - \alpha_2)M_t^2 \\ & -\frac{1}{Re_r}\left(\frac{\mu_t}{\rho^2\sigma_\rho}\right)\frac{\partial\rho\,\partial P}{\partial x_i \partial x_i} \\ & +\frac{1}{Re_r^2}\left(\frac{\mu_t}{\rho^2\sigma_\rho}\right)\frac{\partial\rho\,\partial\sigma_{ij}}{\partial x_i \partial x_j} \end{aligned} \tag{44}$$

$$C_\varepsilon = \rho\varepsilon_t P_2\left[-\frac{1}{3} - n(\gamma - 1) + \frac{2}{3}c_1\right] \tag{45}$$

where

$$M_t^2 = \frac{2k}{\gamma RT} \tag{46}$$

$$\sigma_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu P_2\delta_{ij} \tag{47}$$

$$\varepsilon_t = \varepsilon + \frac{D}{Re_r\rho} \tag{48}$$

$\alpha = 0.5$; $\alpha_1 = 0.4$; $\alpha_2 = 0.2$;
$n = 0.7$; $\sigma_\rho = 0.5$

The remaining four compressible corrections only apply to the k equation so that C_ε is zero for these corrections. For these compressibility corrections, C_k is defined as follows

$$C_k = -Re_r\rho\varepsilon_t\xi^* F(M_t) \tag{49}$$

Sarkar's Model (SA)

$$\xi^* = 1 \tag{50}$$

$$F(M_t) = M_t^2 \tag{51}$$

Zeman's Model (ZF & ZW)

$$\xi^* = 0.75 \tag{52}$$

$$F(M_t) = \left[1 - \exp\left(-\frac{(M_t - M_{to})^2}{\Lambda^2}\right)\right] \times H(M_t - M_{to}) \tag{53}$$

where H(x) is the Heaviside step function, and

$M_{to} = 0.10$; $\Lambda = 0.60$ for free shear (ZF)
$M_{to} = 0.25$; $\Lambda = 0.66$ for wall bounded (ZW)

Wilcox's Model (WI)

$$\xi^* = 1.5 \tag{54}$$

$$F(M_t) = [M_t^2 - M_{to}^2]\, H(M_t - M_{to}) \tag{55}$$

Where M_{to} is 0.25

According to Coakley and Huang[27], the dissipation rate ε in standard k-ε models can collapse abnormally near reattachment points for separated flow and high speed flows. This leads to an erroneous strong growth in turbulent length scale, resulting in unrealistically high skin friction and heat transfer rates. To correct this defect, a turbulent length scale correction similar to the ones proposed by Vuong and Coakley[10] and described by Horstman[11] is also available for both of the k-ε models. With this correction, the turbulent length scale in the wall region is limited so that it can never be greater than a constant times the distance from the wall (C_{TL} x y_n). The default value for the constant C_{TL} is 2.5, which corresponds to the von Karman constant of $\kappa = 0.4$. This length scale correction is only applied to a region near the wall. In this work, a maximum value for y_n^+ is used to define the upper limit of this near wall region. A good discussion in the limit for the near wall region was given by Rodi[28]. The default maximum y_n^+ value used for the wall region is $y_{max}^+ = 20$. The correction is applied as follows:

$$\varepsilon = \max\left(\varepsilon, \frac{k^{3/2}}{Re_r (C_{TL} \cdot y_n)}\right) \tag{56}$$

for $y_n^+ \le y_{max}^+$

With this lower limit imposed on ε, the turbulent length scale will not be larger than the quantity C_{TL} x y_n in the specified near wall region. Horstman[11] limited the turbulent length scale by imposing an upper limit on the turbulent kinetic energy. However, in this study, the turbulent length scale is limited by limiting the collapse of the dissipation rate ε instead. This was done because of the above observation of Coakley and Huang[27] regarding the collapse of ε and because there is more uncertainties in the modeling of the ε variable. In general, the constants C_{TL} and y_{max}^+ will be different for different flows, and the determination of these constants will require some knowledge about the turbulent length scale in the near wall region of the flow under consideration.

Y$^+$ Computation for 3-D geometries

The determination of the distance from a solid wall is difficult for complex 3-D wall bounded flows with arbitrary geometries. In Proteus, the calculation of y_n and y_n^+ is done in a stand-alone, fully vectorized subroutine. The user can either manually indicate the solid portions of the computational boundaries (point-by-point, if necessary) or let Proteus automatically identifies them from the no-slip wall boundary condition. After the solid portions of the computational boundaries have been identified, the straight line distances from all of the solid portions of the computational boundaries to an interior point are computed. Then the distances are compared, and a nearest solid wall is selected. Finally, y_n and y_n^+ are computed with respect to that nearest solid wall. If there are no solid wall present, then y_n and y_n^+ are set to a very high numerical value. This has the effect of driving all of the Van Driest style turbulent damping terms with the form $[1 - \exp(-y_n^+/A^+)]$ to 1, which is expected for turbulent flows away from solid walls. Since the subroutine is separate from the rest of the program, the calculation of y_n and y_n^+ can be readily modified to accommodate any custom geometry.

Applying the above algorithm to a hypothetical 2-D grid in fig. 1, one can see that for an interior point A, the y_n value at A is simply the straight line distance between A and B, and the y_n^+ value at A will be computed using the wall properties at B. For point C, y_n is the straight line distance between C and D.

Numerical Algorithm for One- and Two-Equation Models

The partial differential equation represented by eq. (9) is a scalar equation for the BB model and a vector equation for the k-ε models. The solution procedures for the BB model and the k-ε models are analogous. Therefore, only the k-ε solution procedure is discussed below.

Applying the generalized grid transformation to eq. (9) and putting it into the conservative form results in

$$\frac{\partial \hat{w}}{\partial t} + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} + \frac{\partial \hat{H}}{\partial \zeta} = \hat{S} \tag{57}$$

$$\hat{F} = \hat{F}_C - \hat{F}_D - \hat{F}_{M1} - \hat{F}_{M2} \tag{58}$$

$$\hat{G} = \hat{G}_C - \hat{G}_D - \hat{G}_{M1} - \hat{G}_{M2} \tag{59}$$

$$\hat{H} = \hat{H}_C - \hat{H}_D - \hat{H}_{M1} - \hat{H}_{M2} \tag{60}$$

$$\hat{\mathbf{F}}_C = \frac{1}{J}\begin{bmatrix} \xi_x \rho u k + \xi_y \rho v k + \xi_z \rho w k \\ \xi_x \rho u \varepsilon + \xi_y \rho v \varepsilon + \xi_z \rho w \varepsilon \end{bmatrix} \tag{61}$$

$$\hat{\mathbf{G}}_C = \frac{1}{J}\begin{bmatrix} \eta_x \rho u k + \eta_y \rho v k + \eta_z \rho w k \\ \eta_x \rho u \varepsilon + \eta_y \rho v \varepsilon + \eta_z \rho w \varepsilon \end{bmatrix} \tag{62}$$

$$\hat{\mathbf{H}}_C = \frac{1}{J}\begin{bmatrix} \zeta_x \rho u k + \zeta_y \rho v k + \zeta_z \rho w k \\ \zeta_x \rho u \varepsilon + \zeta_y \rho v \varepsilon + \zeta_z \rho w \varepsilon \end{bmatrix} \tag{63}$$

$$\hat{\mathbf{F}}_D = \frac{1}{J Re_r}\begin{bmatrix} \mu_k (\xi_x^2 + \xi_y^2 + \xi_z^2)\, k_\xi \\ \mu_\varepsilon (\xi_x^2 + \xi_y^2 + \xi_z^2)\, \varepsilon_\xi \end{bmatrix} \tag{64}$$

$$\hat{\mathbf{G}}_D = \frac{1}{J Re_r}\begin{bmatrix} \mu_k (\eta_x^2 + \eta_y^2 + \eta_z^2)\, k_\eta \\ \mu_\varepsilon (\eta_x^2 + \eta_y^2 + \eta_z^2)\, \varepsilon_\eta \end{bmatrix} \tag{65}$$

$$\hat{\mathbf{H}}_D = \frac{1}{J Re_r}\begin{bmatrix} \mu_k (\zeta_x^2 + \zeta_y^2 + \zeta_z^2)\, k_\zeta \\ \mu_\varepsilon (\zeta_x^2 + \zeta_y^2 + \zeta_z^2)\, \varepsilon_\zeta \end{bmatrix} \tag{66}$$

$$\hat{\mathbf{F}}_{M1} = \frac{1}{J Re_r}\begin{bmatrix} \mu_k (\xi_x \eta_x + \xi_y \eta_y + \xi_z \eta_z)\, k_\eta \\ \mu_\varepsilon (\xi_x \eta_x + \xi_y \eta_y + \xi_z \eta_z)\, \varepsilon_\eta \end{bmatrix} \tag{67}$$

$$\hat{\mathbf{F}}_{M2} = \frac{1}{J Re_r}\begin{bmatrix} \mu_k (\xi_x \zeta_x + \xi_y \zeta_y + \xi_z \zeta_z)\, k_\zeta \\ \mu_\varepsilon (\xi_x \zeta_x + \xi_y \zeta_y + \xi_z \zeta_z)\, \varepsilon_\zeta \end{bmatrix} \tag{68}$$

$$\hat{\mathbf{G}}_{M1} = \frac{1}{J Re_r}\begin{bmatrix} \mu_k (\xi_x \eta_x + \xi_y \eta_y + \xi_z \eta_z)\, k_\xi \\ \mu_\varepsilon (\xi_x \eta_x + \xi_y \eta_y + \xi_z \eta_z)\, \varepsilon_\xi \end{bmatrix} \tag{69}$$

$$\hat{\mathbf{G}}_{M2} = \frac{1}{J Re_r}\begin{bmatrix} \mu_k (\eta_x \zeta_x + \eta_y \zeta_y + \eta_z \zeta_z)\, k_\zeta \\ \mu_\varepsilon (\eta_x \zeta_x + \eta_y \zeta_y + \eta_z \zeta_z)\, \varepsilon_\zeta \end{bmatrix} \tag{70}$$

$$\hat{\mathbf{H}}_{M1} = \frac{1}{J Re_r}\begin{bmatrix} \mu_k (\xi_x \zeta_x + \xi_y \zeta_y + \xi_z \zeta_z)\, k_\xi \\ \mu_\varepsilon (\xi_x \zeta_x + \xi_y \zeta_y + \xi_z \zeta_z)\, \varepsilon_\xi \end{bmatrix} \tag{71}$$

$$\hat{\mathbf{H}}_{M2} = \frac{1}{J Re_r}\begin{bmatrix} \mu_k (\eta_x \zeta_x + \eta_y \zeta_y + \eta_z \zeta_z)\, k_\eta \\ \mu_\varepsilon (\eta_x \zeta_x + \eta_y \zeta_y + \eta_z \zeta_z)\, \varepsilon_\eta \end{bmatrix} \tag{72}$$

The diffusion terms $\hat{\mathbf{F}}_{M1}$, $\hat{\mathbf{F}}_{M2}$, $\hat{\mathbf{G}}_{M1}$, $\hat{\mathbf{G}}_{M2}$, $\hat{\mathbf{H}}_{M1}$, and $\hat{\mathbf{H}}_{M2}$ have mixed second-order derivatives, and they are lagged in time on the right hand side to keep the block matrix tridiagonal.

An L-U approximate factorization numerical scheme described by Hoffmann[29] was used. The approximately factored equations for a two sweep L-U scheme are

Upward sweep

$$\{I + \frac{\theta_1 \Delta\tau}{1+\theta_2}[\partial_\xi^- A^+ + \partial_\eta^- C^+ + \partial_\zeta^- E^+ - (\partial_\xi B)^- - (\partial_\eta D)^- - (\partial_\zeta P)^-]\}\Delta \hat{w}^* = RHS \tag{73}$$

Downward sweep

$$\{I + \frac{\theta_1 \Delta\tau}{1+\theta_2}[\partial_\xi^+ A^- + \partial_\eta^+ C^- + \partial_\zeta^+ E^- - (\partial_\xi B)^+ - (\partial_\eta D)^+ - (\partial_\zeta P)^+ - Q]\}\Delta \hat{w}^n = \Delta \hat{w}^* \tag{74}$$

Where A, B, C, D, E, P, and Q are the Jacobian matrices of $\mathbf{F}_C$, $\mathbf{F}_D$, $\mathbf{G}_C$, $\mathbf{G}_D$, $\mathbf{H}_C$, $\mathbf{H}_D$, and $\mathbf{S}$, computed as

$$A = \begin{bmatrix} \xi_x u + \xi_y v + \xi_z w & 0 \\ 0 & \xi_x u + \xi_y v + \xi_z w \end{bmatrix} \tag{75}$$

$$C = \begin{bmatrix} \eta_x u + \eta_y v + \eta_z w & 0 \\ 0 & \eta_x u + \eta_y v + \eta_z w \end{bmatrix} \tag{76}$$

$$E = \begin{bmatrix} \zeta_x u + \zeta_y v + \zeta_z w & 0 \\ 0 & \zeta_x u + \zeta_y v + \zeta_z w \end{bmatrix} \tag{77}$$

$$B = \begin{bmatrix} \alpha_k (\frac{J}{\rho})_\xi & 0 \\ 0 & \alpha_\varepsilon (\frac{J}{\rho})_\xi \end{bmatrix} \tag{78}$$

$$D = \begin{bmatrix} \beta_k (\frac{J}{\rho})_\eta & 0 \\ 0 & \beta_\varepsilon (\frac{J}{\rho})_\eta \end{bmatrix} \tag{79}$$

$$P = \begin{bmatrix} \chi_k (\frac{J}{\rho})_\zeta & 0 \\ 0 & \chi_\varepsilon (\frac{J}{\rho})_\zeta \end{bmatrix} \tag{80}$$

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \tag{81}$$

where

$$\alpha_k = \frac{1}{JRe_r}\mu_k(\xi_x^2+\xi_y^2+\xi_z^2) \qquad (82)$$

$$\alpha_\varepsilon = \frac{1}{JRe_r}\mu_\varepsilon(\xi_x^2+\xi_y^2+\xi_z^2) \qquad (83)$$

$$\beta_k = \frac{1}{JRe_r}\mu_k(\eta_x^2+\eta_y^2+\eta_z^2) \qquad (84)$$

$$\beta_\varepsilon = \frac{1}{JRe_r}\mu_\varepsilon(\eta_x^2+\eta_y^2+\eta_z^2) \qquad (85)$$

$$\chi_k = \frac{1}{JRe_r}\mu_k(\zeta_x^2+\zeta_y^2+\zeta_z^2) \qquad (86)$$

$$\chi_\varepsilon = \frac{1}{JRe_r}\mu_\varepsilon(\zeta_x^2+\zeta_y^2+\zeta_z^2) \qquad (87)$$

$$Q_{11} = \frac{2P_k}{\rho k} + JD + JC_{k1} \qquad (88)$$

$$Q_{12} = -\frac{P_k}{\rho\varepsilon} - Re_r + JC_{k2} \qquad (89)$$

$$Q_{21} = \frac{\varepsilon}{k^2}\left[\frac{c_1P_k}{\rho} - Re_rc_2\varepsilon\right] + JC_{\varepsilon1} \qquad (90)$$

$$Q_{22} = \frac{c_1P_k}{\rho k} - 2Re_rc_2\frac{\varepsilon}{k} + JE + JC_{\varepsilon2} \qquad (91)$$

$$JD = \begin{cases} -\dfrac{2\mu}{Re_r\rho y_n^2} & \text{(CH)} \\ -\dfrac{2\mu}{\rho kRe_r}\left(\dfrac{\partial}{\partial x_i}\sqrt{k}\right)\left(\dfrac{\partial}{\partial x_i}\sqrt{k}\right) & \text{(LS)} \end{cases} \qquad (92)$$

$$JE = \begin{cases} -\dfrac{2\mu}{Re_r\rho y_n^2}e^{-y^*/2} & \text{(CH)} \\ -\dfrac{2\mu c_\mu f_\mu k^2}{\rho\varepsilon^2Re_r^3}\left(\dfrac{\partial^2u_i}{\partial x_j\partial x_l}\right)\left(\dfrac{\partial^2u_i}{\partial x_j\partial x_l}\right) & \text{(LS)} \end{cases} \qquad (93)$$

For the ZH compressibility correction,

$$JC_{k1} = \frac{2P_k}{\rho k}(-\alpha_1M_t^2) + JD(\alpha-\alpha_2)M_t^2 \qquad (94)$$

$$JC_{k2} = -\frac{P_k}{\rho\varepsilon}(-\alpha_1M_t^2) - Re_r(\alpha-\alpha_2)M_t^2 \qquad (95)$$

$$JC_{\varepsilon1} = \frac{JD}{Re_r}P_2\left[-\frac{1}{3} - n(\gamma-1) + \frac{2}{3}c_1\right] \qquad (96)$$

$$JC_{\varepsilon2} = P_2\left[-\frac{1}{3} - n(\gamma-1) + \frac{2}{3}c_1\right] \qquad (97)$$

And for the other compressibility corrections

$$JC_{k1} = JD\xi^*F(M_t) \qquad (98)$$

$$JC_{k2} = -Re_r\xi^*F(M_t) \qquad (99)$$

$$JC_{\varepsilon1} = 0 \qquad (100)$$

$$JC_{\varepsilon2} = 0 \qquad (101)$$

In eq. (73) and eq. (74), the Jacobian matrices A, C, and E are differenced using the first-order upwind differencing, and the Jacobian matrices B, D, and P are differenced using the second-order central differencing. In the RHS, the first-order upwind differencing is used to approximate the convective terms and the second-order central differencing is used for the diffusive terms. The central differencing operators for the Jacobian matrices B, D, and P have been split into a forward differencing part and a backward differencing part so that, for example:

$$\partial_\xi B = (\partial_\xi B)^- + (\partial_\xi B)^+ \qquad (102)$$

where $(\partial_\xi B)^-$ is the backward differencing part of the central differencing operator $\partial_\xi B$, and $(\partial_\xi B)^+$ is the forward differencing part. This is done so that only backward differencing operators are present in the upward sweep, eq. (73), and only forward differencing operators are present in the downward sweep, eq. (74).

In the solution algorithm, the upward sweep is followed by the downward sweep. When either eq. (73) or eq. (74) is applied at each interior point of the computational domain, only one unknown per equation needs to be solved. Therefore, eq. (73) and eq. (74) are solved point by point.

Assuming that the computational block has a dimension of (n1,n2,n3). The right hand side of eq. (73) is known at the current time level n. To advance the solution to the time level n+1, the upward sweep is first marched starting from the lower corner of the computational block, point (2,2,2) to the upper corner of the computational block, point (n1-1,n2-1,n3-1). During the upward sweep, eq. (73) is solved for the intermediate unknown $\Delta\hat{w}^*$ at a point (i1,i2,i3) using data at points (i1-1,i2,i3), (i1,i2-1,i3), and (i1,i2,i3-1). This is possible because the left hand side of eq. (73) contains only backward differencing operators. Then the downward sweep is marched in the opposite direction of the upward sweep, from point (n1-1,n2-1,n3-1) to point (2,2,2). During the downward sweep, eq. (74) is solved for the final unknown $\Delta\hat{w}^n$ at a point (i1,i2,i3) using data at points (i1+1,i2,i3), (i1,i2+1,i3), and (i1,i2,i3+1). This is possible because the left hand side of eq. (74) contains only forward differencing operators. Finally the solution is advanced to the new time level using

$$\hat{w}^{n+1} = \hat{w}^{n} + \Delta \hat{w}^{n} \tag{103}$$

The marching order in both sweeps can be manipulated to achieve better efficiency for the L-U solver. For example, a straight forward Fortran do-loop for the upward sweep is as follows:

```
         do 10 i1 = 2,n1-1
         do 10 i2 = 2,n2-1
         do 10 i3 = 2,n3-1
             *
             *
             *
    10   continue
```

The above do-loop will give the correct results, but it does not take full advantage of the vectorization capability of a vector computer. Since the loop has two nested do-loops, only the innermost loop is vectorized. On the other hand, if the marching is done in the direction normal to the diagonal planes i1+i2+i3 = constant, then a do-loop can be constructed with only one nested do-loop as follows:

```
         do 10 iplane = 1,nplane
         do 10 ipoint = 1,npoint
             *
             *
             *
    10   continue
```

Where nplane is the number of diagonal planes in the 3-D computational block, and npoint is the number of grid points contained a diagonal plane. In general, npoint will vary from plane to plane. It turns out that the points (i1-1,i2,i3), (i1,i2-1,i3), and (i1,i2,i3-1) are all located in the plane (iplane-1), and they are known (for an upward sweep). As the result, the only nested inner loop in the above do-loop can be vectorized over every point in a diagonal plane.

To implement the above marching scheme, an addressing scheme is needed to translate the do-loop variables (ipoint,iplane) to the grid indices (i1,i2,i3) so that the flow properties at (i1,i2,i3) can be conveniently retrieved in the inner do-loop during the computation of the block matrices. There are many ways that this can be accomplished, and, in this project, a simple scheme has been devised to compute the i1, i2, and i3 grid indices from the do-loop variables ipoint and iplane. This scheme does not require any machine-dependent routines, and will work for any Fortran 77 compiler. Basically, this scheme works as follows:

1. Outside of the inner loop, the i1 index of every point in the current diagonal plane iplane is stored in the array iloc(ipoint).
2. Outside of the inner loop, the diagonal line index of every point in a diagonal plane is computed and stored in the array line(ipoint).
3. Inside the inner loop, the i1, i2, and i3 grid indices can be computed from the iloc(ipoint) and line(ipoint) arrays as follows:

```
i1 = iloc(ipoint)
i2 = -i1 + line(ipoint) + 3
i3 = iplane - i1 - i2 + 5
```

Arrays iloc(npoint) and line(npoint) are necessary for a fully vectorized inner loop. This addressing scheme is illustrated in fig. 2. In this diagram, a computational block of (5x6x4) is used. The points on the computational boundaries are assumed to be known through the application of the explicit boundary conditions, and the unknown interior points are indicated by dots of different shapes. The dots with the same shapes belong to the same diagonal plane. It can be seen that for any interior point (i1,i2,i3) belonging to a diagonal plane iplane, the points (i1-1,i2,i3), (i1,i2-1,i3), and (i1,i2,i3-1) belong to the diagonal plane iplane-1, and the points (i1+1,i2,i3), (i1,i2+1,i3), and (i1,i2,i3+1) belong to the diagonal plane iplane+1. This makes it possible to vectorize the above L-U marching scheme.

With some effort, the code can be fully vectorized in the inner do-loop using the above marching scheme, resulting in a very fast L-U solver for the one- and two-equation models. This scheme can also be used in a Navier-Stokes solver.

Boundary values for k and ε are computed explicitly after the downward sweep. If spatially periodic boundary condition in the ξ direction is specified, then an additional set of grid points is added at i1 = n1+1, and flow properties at i1 = n1+1 are explicitly set to the values at i1 = 2. With periodic boundary condition in the i1 direction, the upward sweep is marched from point (2,2,2) to point (n1,n2-1,n3-1), and the downward sweep is march from point (n1,n2-1,n3-1) to point (2,2,2). An analogous procedure is used for periodic boundary conditions in the other two spatial directions.

Automatic Initialization Procedure for One- and Two-Equation Models

With the BL algebraic model, the user only needs to specify the initial conditions for the Navier-Stokes equation. However, with one- and two-equation turbulence models, the user also needs to specify the initial conditions for the turbulence variables to start the time marching process. An automatic initialization procedure for the turbulence variables would greatly simplify the task of the code user. This automatic initialization procedure must provide a unified, transparent integration of turbulence models, even if they are from different classes. Using this automatic starting procedure, the user should be able to use any of the turbulence models in a new or restart calculation without worrying about the initial conditions for the turbulence variables.

In Proteus, the type of the turbulence model used in a calculation is saved together with a minimum number of turbulence variables (zero for algebraic models, one for one-equation models, and two for two-equation models) in a restart file. For the BL model, no turbulence variable is saved. For the BB model, k^2/ε is saved, and for the k-ε models, k and ε are saved. The automatic initialization procedure will depend on the types of turbulence model selected for the current and previous calculations.

BL model

If the BL model is used in the current calculation, then μ_t is calculated using eq. (1). No further initialization is needed.

BB model

For a new calculation or a restart calculation that uses the result of a previous laminar calculation, the turbulent viscosity μ_t is first calculated using the BL model. Then the initial value for k^2/ε is calculated using eq. (19).

For a restart calculation that uses the result of a previous turbulent calculation, the turbulent viscosity μ_t is first calculated using the saved turbulence variable(s) and the appropriate expression for μ_t. Then the initial value for k^2/ε is calculated using eq. (19). If the previous calculation used the BB model, then only μ_t needs to be computed.

CH or LS k-ε model

For a new calculation or a restart calculation that uses the result of a previous laminar calculation, the turbulent viscosity μ_t is first calculated using the BL model. Then using the assumption of local equilibrium and the CH damping function f_μ, initial profiles for k and ε are generated with

$$\varepsilon = \frac{P_k}{Re_r \rho} \tag{104}$$

$$k = \sqrt{\frac{\mu_t \varepsilon}{c_\mu f_\mu \rho}} \tag{105}$$

For a restart calculation that uses the result of a previous turbulent calculation, the turbulent viscosity μ_t is first calculated using the saved turbulence variable(s) and the appropriate expression for μ_t. If the previous calculation used either the BL or BB models, then eq. (104) and eq. (105) are used to obtain the initial profiles for k and ε. However, if either the CH or the LS model was used in the previous calculation, then the k and ε profiles saved from the previous calculation are used as the initial profiles.

The previous automated procedure has been found to perform well for the test cases considered. For example, fig. 3 shows that the automatically generated profiles for k are actually very close to the converged k profiles for the cases of flat plate turbulent boundary layer, turbulent developing S-duct flow, and glancing shock wave/turbulent boundary layer interaction. For the flat plate case, the near wall peak in the initial k profile matches the experimental data even better than the converged k profile. The local equilibrium assumption, eq. (104), and eq. (105), can also be used to generate the inflow profiles for k and ε when such profiles are not otherwise available.

It is recommended that the turbulent calculations using one- or two-equation turbulence models be started from a converged turbulent solution. This will at least produce some reasonable initial profiles for the turbulence variables, minimizing the starting problems. In addition, the initial CFL number used for starting a k-ε calculation should be well below one. After reasonably smooth profiles for k and ε have been achieved, then the CFL number can be increased.

VALIDATION TEST CASES

Validation test cases include the incompressible and compressible flat plate turbulent boundary layers, turbulent developing S-duct flow, and glancing shock wave/turbulent boundary layer interaction. These test cases represent a good mixture of compressible and incompressible wall-bounded flows. The grid sizes for these cases range from 20,655 grid points for the incompressible flat plate case to 321,776 grid points for the glancing shock wave/turbulent boundary layer interaction case. All computations were done on a Cray YMP, and the CPU time required is reported for each test cases. In all of the test cases below, the default artificial viscosity options for the Navier-Stokes equations in Proteus were used. Since all of the flows considered are steady, convergence was assumed to be reached when either the residuals or interested flow properties stopped changing with more iterations at the highest CFL number used in a calculation.

All of the input and output parameters used in Proteus must be nondimensionalized by the appropriate reference conditions, and it is important to specify the correct reference conditions in a Proteus calculation. For this reason, all relevant reference conditions for a particular calculation are discussed below.

Incompressible Flat Plate Turbulent Boundary Layer

Problem Description

Incompressible turbulent flow over a flat plate at zero pressure gradient with a freestream Mach number of 0.3 was computed using the BL, BB, CH, and LS turbulence models. The computational domain for the flat plate ranges from $Re_x = 1.0 \times 10^4$ to $Re_x = 10 \times 10^6$. Comparisons are made with the Klebanoff[30] turbulent boundary layer profiles at $Re_\theta \approx 7700$ and the Wieghart[31] skin friction data.

Reference Conditions

For convenience, the reference Reynolds number Re_r was chosen to be 1×10^6. To simulate the incompressible flow, the freestream Mach number of 0.3 was used. The reference Mach number was set equal to the freestream Mach number. Assuming standard sea level values of air temperature, kinematic viscosity, and pressure for the corresponding reference values, the reference length was calculated to be 0.477631 ft.

Grid Geometry

This is a 2-D problem, but a 3-D 81x51x5 grid (fig. 4a) was used with 5 stations in the third dimension. The x (streamwise) coordinate for the flat plate under consideration runs from $x = 0.01L_r$ to $x=10.0L_r$ with 81 evenly distributed grid points. This streamwise distribution of grid points allows the x station corresponding to the Klebanoff profiles to be in the middle of the computational domain and away from the inflow and outflow boundaries, minimizing the effects of boundary conditions on the solution at that station. 51 grid points are used in the y (normal) direction. The lower y coordinates at the plate surface (lower boundary) are simply $y = 0$ for all x. For the upper freestream boundary, the maximum y coordinate increases with the streamwise distance along the plate to keep approximately the same number of grid points within the boundary layer as it grows. This is analogous to a boundary layer code that adds more grid points in the normal direction as the boundary layer grows. To obtain the maximum y coordinates, the turbulent boundary layer thickness at an x station was estimated from the power law formula[32]:

$$\delta = \frac{0.38x}{Re_x^{1/5}} \tag{106}$$

In the flat plate calculations, approximately 90% of the grid points are placed inside the turbulent boundary layer in the normal direction, and the rest of the grid points are placed in the freestream region. To resolve the near wall region, grid points are packed near the plate surface using an algebraic formula so that the y^+ value of the nearest point to the wall is approximately from 0.1 at the leading edge to 0.9 at the trailing edge. At the x station corresponding to the Klebanoff experiment, the nearest y^+ is about 0.5, and there are approximately 10 points within $y^+ < 10.0$.

Initial Conditions

To start the BL calculation, the initial velocities u and v were calculated using the Blasius solution for the laminar flat plate boundary layer. The non-dimensionalized pressure, P/P_r, was set to 1.0 everywhere. Calculations using the BB and CH models were done using the BL solution as the initial conditions. The initial values of k^2/ε, k, and ε for the BB and the CH model were obtained using the automatic starting procedure described above. The LS calculation was started from the CH solution.

Boundary Conditions

For all of the calculations for this test case, constant stagnation enthalpy was assumed. Therefore, no boundary condition was needed for temperature.

At the plate leading edge, u and v were specified using the Blasius solution. This was done because the Re_x value at that location is well within the laminar region. w was set to zero. The pressure was set to the freestream value. ξ gradients for k^2/ε, k, and ε were set to zero for the first approximately 800 iterations. After that, k^2/ε, k, and ε profiles were held fixed for the rest of the calculations. This was done to smooth out any kinks or sharp angles in the upstream profiles of k^2/ε, k, and ε that were produced when the BL model was used in the automatic initialization process.

At the plate trailing edge, u, v, k, and ε were linearly extrapolated in the ξ direction. w and the ξ gradient of k^2/ε were set to zero. The pressure was set to the freestream value.

At the flat plate surface, the velocity, k, ε, k^2/ε, and the normal pressure gradient were set to zero.

At the outer boundary, u and P were set to the freestream values. The velocity component w and the η gradients of v, k, and ε were set to zero. The nondimensional value of k^2/ε was set to 0.5.

Finally, the ζ gradients of all flow variables were set to zero to simulate 2-D flow.

Computational Details and Results

For the BL and the BB calculations, the starting CFL number was 5, and it was gradually increased to a final value of 20 during the calculations. For the CH calculations, the starting and ending CFL values were 1 and 20, and for the LS calculation, the minimum and maximum CFL numbers were 1 and 15. LS calculations with CFL number larger than 15 diverged.

For the BL calculation, the skin friction distribution stopped changing after 3839 iterations. The BB, CH, and LS calculations took 6631, 7531, and 8632 iterations, respectively, to reach the same level of convergence. The CPU times required for the BL, BB, CH, and LS calculations are 6.031×10^{-5}, 6.650×10^{-5}, 6.822×10^{-5}, and 8.667×10^{-5} sec/iteration/grid point, respectively. As can be seen, the differences in the CPU times required for computations with different turbulence models are relatively small, even with models from different classes. For example, the CH model only required 13% more CPU sec/iteration/grid point than the BL model for this test case. Note that the required CPU times above are for full Navier-Stokes calculations with no simplifying assumptions for the boundary layer flow.

The plots of the velocity profile, fig. 5a and fig. 5b, show that all of the turbulence models accurately compute the turbulent velocity profile as compared with the Klebanoff data at $Re_\theta \approx 7700$. The differences between the models are quite small. In fig. 5c, the Reynolds shearing stress predictions by all of the turbulence models are also in very good agreement with the Klebanoff profile with the BB and BL models giving better agreement for the Reynolds shearing stress in the outer region of the boundary layer, and the CH and LS models giving better prediction of the peak of the Reynolds shearing stress in the near wall region. In fig. 5d, the CH model is seen to predict the peak in k better than the LS model, but in the outer region, both models give practically the same k profile. The k^2/ε profiles obtained using the BB, CH, and LS models are compared in fig. 5e. There were differences between the BB model and the k-ε models, but the general trend and magnitude are approximately the same for all models.

In fig. 5f, the local skin friction coefficient predictions by the CH and LS are slightly better than the BL and BB models at higher values of Re_θ, but the differences are relatively small between the different turbulence models.

The LS is well known for not accurately predicting the peak in k in the near wall region. However, in this particular calculation, that defect does not seem to affect its accuracy in velocity, Reynolds shearing stress, k^2/ε, and skin friction predictions as compared with the CH model. The LS and the CH results are almost identical for those mean flow properties.

Compressible Flat Plate Turbulent Boundary Layer

Problem Description

Compressible turbulent flow over a near adiabatic flat plate at zero pressure gradient with a freestream Mach number of 3.0 was computed using the BL, BB, CH, and LS turbulence models. The computational domain for the flat plate ranges from $Re_x = 3.7\times10^4$ to $Re_x = 2.2\times10^7$. Comparisons are made with the Van Driest II correlation (VDII) as described by Hopkins and Inouye[33]. In addition, velocity and temperature profiles at the x station corresponding to $Re_\theta \approx 9751$ are also compared with the Kim and Settles[37] inflow profiles for their glancing shock wave/turbulent boundary layer interaction experiment.

Reference Conditions

The solution of this calculation is used as the inflow boundary condition for the glancing shock wave/turbulent boundary layer interaction calculation. Therefore, the flow conditions for this calculation have been carefully chosen to match those from the Kim and Settles experiment[34-37]. For convenience, the reference length L_r was set equal to the incoming boundary layer thickness at $Re_\theta \approx 9751$ in the Kim and Settles experiment, which is 0.009908 ft. The reference Mach number was set equal to the freestream Mach number of 3.0. Using the isentropic relation, the freestream static temperature was calculated to be 189 deg. R, and this value was used as the reference temperature. Assuming the reference pressure to be equal to the freestream static pressure, the reference density was calculated to be 0.04667 lbm/ft^3. The reference Reynolds number was calculated from the above reference values to be 1.8694×10^5.

Grid Geometry

This is a 2-D problem, but a 3-D 91x71x5 grid (fig. 4a) was used with 5 stations in the third dimension. The x (streamwise) coordinate for the flat plate under consideration runs from $x = 0.2L_r$ to $x=120.0L_r$ with 91 evenly distributed grid points. This streamwise distribution of grid points allows the x station corresponding to the Kim and Settles inflow profiles to be approximately in the middle of the computational domain and away from the boundaries, minimizing the effects of boundary conditions on the solution at that station. 71 grid points are used in the y (normal) direction. Following the procedure outlined above for the incompressible flat plate calculation, the height of the grid is increased in the streamwise direction to keep approximately the same number of grid points in the boundary layer. To resolve the near wall region, grid points are packed near the plate surface so that the y^+ value of the nearest point to the wall is about from 0.02 at the leading edge to 0.6 at the trailing edge. At the x station corresponding to the Kim and Settles inflow profiles, the nearest y^+ is about 0.4, and there are approximately 14 points within $y^+ < 10.0$.

Initial Conditions

To start the BL calculation, the initial velocities u and v were calculated using the Blasius solution for the laminar flat plate boundary layer. w was set to zero, and P/P_r was set to 1.0 everywhere. The boundary layer temperature profile was calculated using the following relation[38]:

$$T = T_w + (T_{aw} - T_w)\frac{U}{U_e} - \frac{rU^2}{2c_p} \qquad (107)$$

where

$$T_{aw} = T_e + \frac{rU_e^2}{2(C_p)_e} \qquad (108)$$

$$T_w = 1.06T_{aw} \qquad (109)$$

$$r = 0.896 \qquad (110)$$

Note that eq. (109) was taken from the Kim and Settles data.

Calculations using the BB and CH models were done using the BL solution as the initial conditions. The initial values of k^2/ε, k, and ε for the BB and the CH models were obtained using the automatic starting procedure described above. The LS calculation was started from the CH solution.

Boundary Conditions

For all of the calculations in this test case, the energy equation was solve simultaneously with the momentum and the continuity equations. The molecular viscosity and the thermal conductivity coefficients were calculated as function of the local temperature using Sutherland's formula for air[38].

At the flat plate leading edge, u and v velocity components were specified using the Blasius solution. w was set to zero. The pressure was set to the freestream value. Inflow static temperature profile was held fixed from the initial conditions. ξ gradients for k^2/ε, k, and ε were set to zero for the first approximately 300 iterations. After that, k^2/ε, k, and ε profiles were held fixed for the rest of the calculations. This was done for the same reason as discussed above in the incompressible flat plate test case.

At the plate trailing edge, ξ gradients of all flow variables were set to zero.

At the flat plate surface, the velocity, k, ε, k^2/ε, and the normal pressure gradient were set to zero. T was set to the wall temperature reported in the Kim and Settles experiment.

At the freestream boundary, u, P, and T were set to the corresponding freestream values. The velocity component w and the η gradients of v, k, and ε were set to zero. The nondimensional value of k^2/ε was set to 0.5.

Finally, the ζ gradients of all flow variables were set to zero to simulate 2-D flow.

Computational Details and Results

The minimum CFL numbers used are 2.0 for the BL model and 0.5 for all the other models. The maximum CFL numbers used are 10 for the BB model and 5 for all the other models. Note that the higher CFL value of 10 might also work for models other than the BB model, but this has not been tried due to time constraints.

For the BL calculation, the skin friction distribution remained constant after 2855 iterations. The BB, CH, and LS calculations took 2745, 3781, and 3174 iterations, respectively, to reach the same level of convergence. The CPU times required for the BL, BB, CH, and LS calculations are $9.042\text{x}10^{-5}$, $9.436\text{x}10^{-5}$, $9.836\text{x}10^{-5}$, and $11.77\text{x}10^{-5}$ sec/iteration/grid point, respectively. More CPU time is required for this calculation than the incompressible flat plate calculation, because the energy equation was solved. Even so, there are still very small differences in the CPU times required for different turbulence models. In this test case, The CH model only took 9% more CPU time than the BL model.

The skin friction plot, fig. 6a, compares the BL, BB, CH, and LS skin friction results with the VDII. Also shown is a bounding bar denoting a ±10% deviation from the VDII. It can be seen that all of the models give results that are within this bounding bar. The BL, BB, and the CH models produce the best agreement, whereas the LS model's prediction is about 10% lower than the VDII. It should be noted that with the BL, BB, and the CH models, the way that y^+ is computed affects the skin friction predictions, and this effect will be examined later. In these calculations, y^+ was computed using eq. (4) with $\nu_1 = \nu_L$ and $\rho_1 = \rho_w$.

In fig. 6b, the momentum thickness predictions of the BL, BB, CH, and LS models are compared with the VDII. Again, the LS model results deviates the most from the VDII. Note that if y^+ for the CH model was computed with $\nu_1 = \nu_w$ and $\rho_1 = \rho_w$, the CH skin friction and momentum thickness results will be very similar to the LS results. The LS results are not sensitive to y^+ computation, because that model does not use y^+ in its formulation.

The velocity and static temperature profiles are compared with the Kim and Settles data at $Re_\theta \approx 9751$ in fig. 6c and d. The experimental data were collected as the incoming boundary layer profiles for a glancing shock wave/turbulent boundary layer interaction experiment. All of the turbulence models used give results that are in excellent agreement with the experimental data. Also shown in the velocity plot is the Schlichting[39] empirical correlation of the velocity profile in a compressible boundary layer for a freestream Mach number of 2.4.

Computing y^+ for incompressible flow is straight forward. But for compressible flow, there are some ambiguities in the ν and ρ values which are needed to compute y^+ (see eq. (4)). Baldwin and Lomax[4] recommended $\nu_1=\nu_w$ and $\rho_1=\rho_w$ in their original report. But $\nu_1=\nu_L$ was used in more recent works in compressible turbulence modeling[6, 8]. Furthermore, Coakley and Huang[27] reported that the choice in ρ_1 has important implications for the predictions of skin friction and heat transfer, especially for cold walls. Therefore, to examine the sensitivity of the BL, BB, and the CH results with ν_1 and ρ_1 choices in the y^+ computation, flat plate computations were made using:

1. The BL model with ($\nu_1=\nu_L$, $\rho_1=\rho_w$), ($\nu_1=\nu_L$, $\rho_1=\rho_L$), ($\nu_1=\nu_w$, $\rho_1=\rho_w$), and ($\nu_1=\nu_w$,$\rho_1=\rho_L$)
2. The BB model with ($\nu_1=\nu_L$, $\rho_1=\rho_w$) and ($\nu_1=\nu_w$, $\rho_1=\rho_w$)
3. The CH model with ($\nu_1=\nu_L$,$\rho_1=\rho_w$) and ($\nu_1=\nu_w$,$\rho_1=\rho_w$).

For the BL model, fig. 7a and b show that the choice in ν_1 affects the results more significantly than ρ_1, and that the best results were obtained using ($\nu_1= \nu_L$, $\rho_1=\rho_w$). Using ($\nu_1=\nu_w$, $\rho_1=\rho_w$) as recommended by Baldwin and Lomax gives flat plate results that are more that 10% lower than the VDII.

Since the BL results show that ρ_1 does not significantly affect the results for these calculations and that the best results were obtained using $\rho_1=\rho_w$, $\rho_1=\rho_w$ was used for the BB and CH models, and only the choice in ν_1 was examined for these models.

It can be seen in fig. 7c and d that the BB is not as sensitive to the choice in ν_1 as the other turbulence models. Both choices in ν_1 produce excellent agreement with the VDII. However, using $\nu_1=\nu_w$ gives slightly lower skin friction and momentum thickness results than $\nu_1=\nu_L$.

On the other hand, as shown in fig. 7e and f, using $\nu_1=\nu_w$ in the CH model significantly lowers the skin friction and momentum thickness results. In fact, the CH model with $\nu_1=\nu_w$ and $\rho_1=\rho_w$ gives results that are almost identical to the LS model. This might be due to the facts that the CH and LS models are very close to each other in the actual formulations, and that both were derived and calibrated in the incompressible turbulent flow regime.

The unmodified k-ε models have been found to fail to predict the observed decrease in the spreading rate for the compressible mixing layer, and compressibility options are available to correct this deficiency. However, these compressibility corrections can adversely affect the flat plate results. As the result, Wilcox[9] recommended using the Zeman compressibility option with M_{to} and Λ set to 0.25 and 0.66, respectively to minimize its impact on the flat plate computation. Wilcox also suggested his own compressibility correction. To study the effects of these compressibility corrections on the Proteus flat plate results, the Mach 3.0 flat plate computations with the CH model were done with and without these corrections. Effects of these corrections on the results with the LS model are expected to be similar to the CH model, since both models are very similar in form. As can be seen in fig. 8a and b, the WI and ZW corrections have no effect on the flat plate results, while the ZH, SA, and ZF corrections significantly lower the skin friction and the momentum thickness Reynolds number predictions.

Turbulent Developing S-Duct Flow

Problem Description

Turbulent incompressible flow in an S-duct was computed using first the BL model and then the CH k-ε model. The CH calculation was started from a BL converged solution. The S-duct geometry and comparison data were obtained from an experiment conducted by Taylor et al.[40] The S-duct in that experiment consists of two 22.5 degree bends with constant area square cross section. Turbulent experimental data for the S-duct is available for the flow Reynolds number of 40,000 (based on the bulk velocity and the duct hydraulic diameter, D_H).

Reference Conditions

The calculations were done at a flow Mach number of 0.2 (based on bulk velocity) in order to minimize compressibility effects and, at the same time, achieve reasonable convergence rate with the Proteus code. For convenience, the reference Mach number was also set at 0.2, and the reference Reynolds number was chosen to be 40,000. Assuming standard sea level values of temperature, kinematic viscosity, and static pressure for the reference values of T_r,ν_r, and P_r respectively, the reference velocity and the reference length were calculated to be 223.32 ft/sec and 0.028658 ft, respectively:

Grid Geometry

The computational grid for the S-duct is shown in fig. 4b. The grid was created using the Gridgen grid generation software package for the Iris graphics workstation. The computational grid ranges from 7.5 D_H upstream of the start of the S-bend to 7.5 D_H downstream of the end of the S-bend with 81x31x61 grid points in the x, y, and z directions, respectively. Since the S-duct is symmetric with respect to the y = constant plane, only half of the duct is discretized. To resolve the viscous layer, grid points were tightly packed near the solid walls so that the y^+ value of the first grid point off of the solid walls is a little less than 0.5.

Initial Conditions

To start the BL calculation, the initial static pressure was set equal to the reference pressure, and the fluid velocity components u,v, and w were set to zero everywhere in the duct.

To start the CH calculation, the initial values of u, v, w, P, and μ_t for the S-duct flow were obtained from the converged BL solution. The initial profiles for k and ε were obtained using the automatic initialization procedure described above.

Boundary Conditions

For both the BL and the CH calculations, constant stagnation enthalpy was assume, eliminating the need for solving the energy equation and specifying its initial and boundary conditions.

At the duct inlet, the total pressure was specified to be $P_o/P_r = 1.028281121$. This was computed using the isentropic formula with the freestream static pressure and the bulk Mach number. Zero streamwise gradient was specified for u velocity component. The v and w velocity components were set to zero. Zero streamwise gradients were specified for k and ε for the first 41 iterations of CH calculation. After that, the inlet k and ε profiles were held fixed for the rest of the computation. Around 20-50 iterations at the lowest CFL number used is usually enough to smooth out those profiles without significantly changing the magnitudes of either k or ε.

At the duct exit, the duct exit pressure was specified to be $P_e/P_r = 0.98415512$. This was found by trial and error in order to match the experimental mass flow rate in the duct. Zero streamwise gradient was specified for all velocity components, k, and ε.

At the solid walls, fluid velocity, k, ε, and normal pressure gradient were set to zero. Standard symmetry boundary condition was used in the symmetry plane.

Computational Details and Results

The BL calculation was done with the following sequence of CFL numbers

CFL	No. of iterations
1.0	100
5.0	200
10.0	After 300 iterations

And for the CH calculation,

CFL	No. of iterations
1.0	762
3.0	1445
10.0	After 2207 iterations

For the BL calculation, the mass flow rate throughout the duct was found to be constant to within 0.1% after 6278 iterations. Started from the BL solution, the CH calculation took another 3667 iterations to achieve the same uniformity in the mass flow rate. The CPU times required for the BL calculation and the CH calculation are 5.168×10^{-5} and 6.506×10^{-5} sec/iteration/grid point, respectively. In this calculation, the CH model required approximately 26% more CPU time than the BL model.

The S-duct streamwise velocity contours and the coordinate system used in plotting the results are shown in fig. 9. The coordinate system r-z used here is the same as that defined in the original experiment. In fig. 10, the wall pressure predictions of the BL and CH calculations are compared with the experimental data. Note that the reference pressure used to calculate c_p in the plot was the centerline wall pressure at streamwise distance = -1.0 D_H (one hydraulic diameter upstream of the start of the S-bend), r = 0.5, and z = 1.0. The experimental data shows a larger scattering of the wall pressures at the streamwise station = -1.0 D_H than the Proteus result. Otherwise, the agreement is very good. Both turbulence models correctly computed the trend as well as the pressure drop along the S-duct walls. In fig. 11 and fig. 12, plots of experimental and computational streamwise velocity profiles on the symmetry plane (z = 0) and the midspan plane (r = 0.5) of the S-duct are shown for the 5 streamwise stations along the S-duct. The agreement with the experimental data is generally good for both turbulence models. The asymmetry in the symmetry plane velocity profiles due to the pressure induced secondary motion are correctly predicted by the Proteus code. No other turbulence model was used in the S-duct calculation, because the results obtained so far are fairly insensitive to the turbulence models used.

Glancing Shock Wave/Turbulent Boundary Layer Interaction

Problem Description

Glancing shock wave/turbulent boundary layer interaction produced by Mach 3.0 flow past an 10 degree sharp fin mounted on a flat plate was computed using the BL and CH models. Experimental data are available from the Kim and Settles experiment[34-37] at the Penn State Gas Dynamics Laboratory. The data available for the purpose of comparison include measured inflow turbulent boundary layer velocity and temperature profiles, flat plate surface pressure, skin friction, and surface flow angles. The surface skin friction was experimentally measured by laser interferometry.

Reference Conditions

This computation uses the same reference conditions as described in the compressible flat plate turbulent boundary layer test case above.

Grid Geometry

The geometry for the computational grid used is shown in fig. 4c. The grid was generated using the Gridgen grid generation software package for the Iris graphics workstation. A total of 52x68x91 grid points were used in the x, y, and z directions, respectively. a smaller number of grid points was used in the x direction than the other two directions, because numerical experimentation has found that the results are relatively insensitive to grid point distribution in the x direction, as long as Δx is not too large compared with the incoming boundary layer thickness. The computational domain extended from 6 L_r upstream of the fin leading edge to 45 L_r downstream of the fin leading edge. Recall that L_r was set equal to the incoming turbulent boundary layer thickness at $Re_\theta \approx 9751$ as reported in ref. 37. The width of the upstream and downstream boundaries are 4.2 L_r and 41.3 L_r, respectively. The upstream width was selected so that the computational boundaries are outside of the upstream influence region, and the downstream width was selected so that the inviscid shock wave would pass through the downstream computational boundary. The height of the computational domain is 7 L_r. Uniform grid spacing of was used in the x direction with $\Delta x = 1.0\ L_r$ Grid points were clustered in both the y and z directions in order to resolve the viscous layers on the flat plate and on the fin solid surfaces. The minimum y^+ value in the y direction (on the flat plate) is approximately 0.025 to 0.05, and the minimum y^+ value in the z direction (on the fin) is approximately 0.1 to 0.9. Maximum grid spacings in the y and z directions are 0.616 L_r and 0.531 L_r, respectively. Six streamwise grid points were placed upstream of the sharp fin.

Initial Conditions

To start the BL calculation, the converged BL flat plate turbulent boundary layer profiles at the inflow x station of the flow velocity, pressure, and temperature were used everywhere except near the fin surface. Near the fin surface, a 1/7th power law velocity profile and a linear temperature profile were used to smoothly blend the fin surface no-slip boundary condition to the rest of the initial flow field.

The converged BL solution was used to start the CH calculation. The initial profiles for k and ε were obtained using the automatic initialization procedure as described above.

Boundary Conditions

At the upstream boundary, the static pressure was set equal to the freestream static pressure. The static temperature T as well as the velocity components u, and v specified at the upstream boundary were interpolated from the converged solution of the BL flat plate calculation at the same x distance downstream of the Kim and Settles inflow profiles. The w velocity component was set to zero. Zero streamwise gradients were specified for k and ε for the first 20 iterations of the CH calculation. After that, the inlet k and ε profiles were held fixed for the rest of the CH computation.

At the downstream boundary and the far field boundaries (boundaries that are opposite from the fin and the flat plate), zero gradients was specified for all flow variables.

At the flat plate and fin solid surfaces, fluid velocity, k, ε, and normal pressure gradient were set to zero. The static temperature was set to 1.06 T_{aw} as reported in ref. 37.

Finally, symmetry boundary condition was set at the symmetry plane upstream of the fin leading edge.

Computational Details and Results

The BL calculation was done with the following sequence of CFL numbers

CFL	No. of iterations
0.5	425
1.0	425
2.0	850
3.0	850
5.0	After 2550 iterations

And for the CH calculation,

CFL	No. of iterations
0.2	700
1.0	700
3.0	700
4.0	700
5.0	700

Note that the use of CFL values higher than 5.0 was possible for both the BL and CH models, but at CFL numbers greater than 5.0 and with the default artificial viscosity parameters in Proteus, flow properties started to oscillate around shock wave in the inviscid region, even though the surface flow properties would still be smooth. Convergence was assumed to be reached when the wall pressure, skin friction, and surface flow angle distributions no longer change with more iterations at the highest CFL number. With the above schedule for CFL numbers, the BL calculation took approximately 4500 iterations to converge. Started from the BL model, the CH calculation required approximately another 3500 iterations to converge. The CPU times required for the BL calculation and the CH calculation were 7.888×10^{-5} and 9.757×10^{-5} sec/iteration/grid point, respectively. The CH k-ε model took about 24% more sec/iteration/grid point than the BL algebraic model for this test case.

Starting from the converged CH solution, the turbulent length scale correction took 1353 iterations to converge, and the Sarkar compressibility correction took 2848 iterations. Both of these corrections required only slightly more CPU sec/iteration/grid point than the standard CH computations. Note that when any of the turbulence options are changed in a calculation, such as a change in the turbulence model types, length scale corrections, or compressibility corrections, it is a good idea to reduce the CFL number to less than 1.0 to avoid starting problems. The CFL number can then be increased gradually afterwards.

When these computations were first attempted, both of the BL and the CH computations would always diverge in the vicinity of the symmetry plane in front of the fin leading edge, regardless of the time step size or the artificial viscosity options used. A careful examination of the Proteus computations revealed that the extremely tight clustering of grid points normal to the symmetry plane has caused Proteus to compute unreasonably high gradients of the mean flow velocity normal to the symmetry plane, causing both the total vorticity magnitude and the production of the turbulent kinetic energy to be too high. This caused the turbulent viscosity coefficient in that region to be unreasonably high, preventing the BL and CH calculations from convergence.

The grid clustering normal to the symmetry plane is not good, because there is no real physical reason for it. However, it has to be there, because of the structure of the grid used. Therefore, to alleviate this problem and allow the computations to proceed, the total vorticity magnitude and the production rate for the turbulent kinetic energy upstream of the fin were computed only in the x-y plane. This modification is reasonable, since upstream of the fin, the flow is essentially a supersonic 2-D turbulent boundary layer flow. For the rest of the flow field, those turbulent properties were computed using the full 3-D expressions.

The BL computed surface streamlines and the general layout of the test case is shown in fig. 13. The upstream influence and the primary separation lines can clearly be seen. From comparisons of the surface flow angles with the experimental data below, these computed streamlines are quite accurate.

The BL results are shown in fig. 14. It can be seen that

that the skin friction result, fig. 14b, is sensitive to the method of y^+ computation, and the best agreement was obtained with the computation of y^+ using $\nu_1 = \nu_w$. The surface pressure and flow angles distributions are seen to be relatively insensitive to the method of y^+ computation. Note that the BL c_f prediction shows a small initial rise in c_f at $\beta \approx 30°$. Through numerical experimentation, it was found that if the number of grid points in the z direction is not sufficient, then the BL calculation will not pick up this flow feature. On the other hand, the CH k-ε calculation will always predict this initial rise in c_f independent of the number of grid points in the z direction. Overall, the agreement between the BL results and the experimental data is very good.

The CH results and the effects of the turbulent length scale correction are shown in fig. 15. Note that the CH calculations were done with the method of y^+ computation that gave the best results for the BL calculation above. The agreements in the surface pressure and flow angle distributions are quite good and, for the most part, are relatively insensitive to the turbulent length scale correction applied. The standard CH k-ε model does quite well in predicting the general shape and trend of the c_f curve. In particular, the initial rise in c_f is always predicted by the CH model regardless of the number of grid points used in the z direction. This is not the case with the BL model above. However, the CH model significantly overpredicts the skin friction in the interaction region at b = 15°. This can be caused by either the collapse in the turbulent dissipation rate as discussed in ref. 27, or insufficient grid-clustering in the inner layer of the turbulent boundary layer as reported in ref. 41.

Avva et al.[41] have found that the overshoot in the skin friction prediction by the CH model in the reattachment region can be corrected by refining the grid in the near wall region. Different number of grid points as well as different grid clusterings in the y direction have been tried, but they did not correct the overshoot in skin friction prediction in any significant way. It should be noted that the total number of grid points in this calculation was limited by practical considerations, and it is possible that the skin friction prediction of the CH model will improve with further grid refinement.

On the other hand, the turbulent length scale correction and the compressibility corrections will directly affect the skin friction results without requiring more grid points or more CPU time. Both of these corrections have the net effect of lowering the skin friction prediction. As can be seen in fig. 15, the turbulent length scale correction significantly lowers the skin friction as well as increasing the surface flow angles in the interaction region. A number of different values for the turbulent length scale constant C_{TL} and y^+_{max} were tried, but the CH skin friction predictions with the length scale correction are still too low (see fig. 15b) compared with the experimental data. Picking the correct turbulent length scale constant and the near wall region requires knowledge about the turbulent length scale of the flow and necessitates a trial and error approach if this data is not available. It is possible that good results can be achieved with the right combination of constants.

Wilcox[9] has found that the Sarkar and other compressibility correction will improve the k-ε skin friction prediction in the reattachment region of separations induced by shock wave/turbulent boundary layer interactions. To investigate the effects of the compressibility corrections in the present problem, the Sarkar correction was used. From fig. 16b, it can be seen that the standard Sarkar correction brings the CH skin friction prediction in the interaction region down to match the experimental data precisely. As can be seen in fig. 16a and c, the effects of the Sarkar correction on the surface pressure and flow angle distribution are small.

In fig. 17, the effects of the turbulent length scale correction and the Sarkar compressibility correction on the turbulence variables k, ε, and the turbulent length scale constant C_{TL} at a point in the interaction region are shown. It can be seen that both of these corrections lower the values of k, ε, as well as the turbulent length scale constant C_{TL}. In fig. 17c, for example, the Sarkar compressibility correction smoothly lowers the entire C_{TL} curve, whereas the length scale correction will clip the turbulent length scale constant C_{TL} values to below 3.0 for y^+ values less than 20, as specified. It is interesting to note that the k and ε profiles (fig. 17a-b) computed with the length scale correction are still smooth, even though the turbulent length scale constant itself is clipped quite abruptly by the length scale correction. Also, the turbulent length scale constant tends to a value around 2 - 2.5 at y+ > 100. This corresponds to a von Karman constant of approximately 0.4 - 0.5.

Finally, the quasiconical property of this glancing shock wave/turbulent boundary layer interaction flow as reported by Kim and Settles[42] is investigated. Since Proteus provided the solution for every point in the computational domain, it is easy to compute the skin friction at any location on the flat plate. In fig. 18, the best BL and CH results of surface pressure, skin friction, and surface flow angle distributions are compared for three different radii from the fin leading edge. These three radii have been chosen so that they will be away from both the initial non-conical inception zone and the downstream computational boundary. Note that these radii should really be measured from the conical virtual origin which is slightly upstream of the fin leading edge, as described in detail by Settles and Kimmel[43]. But in this study, the radii were measured from the fin leading edge to facilitate comparisons with the experimental data. Moreover, the error in locating the conical origin is small for larger values of the radius R. As can be seen, both of the BL and CH results show only small variations with different radii, and the surface pressure, local skin friction coefficient, and surface flow angle distributions are essentially independent of the radius R.

CONCLUSION

A general turbulence modeling capability has been implemented in Proteus, a 3-D compressible Reynolds-averaged Navier-Stokes computer code. This new turbulence modeling capability consists of four turbulence models: the Baldwin-Lomax algebraic model, the Baldwin-Barth one-equation model, the Chien k-ε model, and Launder-Sharma k-ε model.

Features of the current turbulence modeling implementation include:

- Well documented, modular, and easy to use turbulence modeling options.
- Uniform integration of turbulence models from different classes.
- Multiple solid boundaries treatment.
- Automatic initialization of turbulence variables for calculations using one- and two-equation turbulence model.
- Fully vectorized L-U solver for one- and two-equation models.

Validation test cases showed good agreement between the Proteus calculations using different turbulence models and the experimental data. The compressible turbulent solutions have been found to be sensitive to the method of y^+ computation, turbulent length scale correction, and compressibility corrections.

Finally, the test cases demonstrated that the highly optimized one- and two-equation turbulence models can be used in routine 3-D Navier-Stokes computations with no significant increase in CPU time as compared with the Baldwin-Lomax algebraic model. As an example, the Chien k-ε model only required 9%-26% more CPU sec/iteration/grid point than the Baldwin-Lomax algebraic model, depending on the test case.

REFERENCES

1. Towne, C. E., Schwab, J. R., Benson, T. J., and Suresh, A., "PROTEUS Two-Dimensional Navier-Stokes Computer Code - Version 1.0, Volumes 1-3," NASA TM's 102551-3.
2. Towne, C. E., Schwab, J. R., and Bui, T. T., "Proteus Two-Dimensional Navier-Stokes Computer Code - Version 2.0, Volumes 1-3," to be published as NASA TM's.
3. Towne, C. E., Schwab, J. R., and Bui, T. T., "Proteus Three-Dimensional Navier-Stokes Computer Code - Version 1.0, Volumes 1-3," to be published as NASA TM's.
4. Baldwin, B. S., and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257.
5. Chien, K. Y., "Prediction of Channel and Boundary-Layer Flows with a Low-Reynolds-Number Turbulence Model," *AIAA Journal*, Vol. 20, No. 1, Jan. 1982, pp. 33-38.
6. Baldwin, B. S. and Barth, T. J., "A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows," AIAA Paper 91-0610, Jan. 1992.
7. 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, 1974, pp. 131-138.
8. Zhang, H. S., So, R. M. C., Speziale C. G., and Lai, Y. G., "A Near-Wall Two-Equation Model for Compressible Turbulent Flows," NASA CR-189565, ICASE Report No. 91-82, Nov. 1991.
9. Wilcox, D. C., "Progress in Hypersonic Turbulence Modeling," AIAA Paper 91-1785, June, 1991.
10. Vuong, S. T. and Coakley, T. J., "Modeling of Turbulence for Hypersonic Flow with and Without Separation," AIAA Paper 87-0286, Jan. 1987.
11. Horstman, C. C., "Hypersonic Shock-Wave Turbulent-Boundary-Layer Interaction Flows—Experiment and Computation," AIAA Paper 91-1760, June, 1991.
12. Menter, F. R., "Performance of Popular Turbulence Models for Attached and Separated Adverse Pressure Gradient Flows," AIAA Paper 91-1784, June 1991.
13. Yu, N. J., Allmaras, S. R., and Moschetti, K. G., "Navier-Stokes Calculations for Attached and Separated Flows Using Different Turbulence Models," AIAA Paper 91-1791, June 1991.
14. Stanaway, S. K., McCroskey, W. J., and Kroo, I. M., "Navier-Stokes Analysis of Blunt Trailing Edge Airfoils," AIAA Paper 92-0024, Jan. 1992.
15. Peters, G. R., Agarwal, R. K., and Deese, J. E., "Numerical Simulation of Several Shock-Separated Boundary-Layer Interaction Flows Using Zero- and Two-Equation Turbulence Models," AIAA Paper 86-0248, Jan. 1986.
16. Sahu, J., and Danberg, J. E., "Navier-Stokes Computations of Transonic Flows with a Two-Equation Turbulence Model," *AIAA Journal*, Vol. 24, No. 11, Nov. 1986, pp. 1744-1751.
17. Nichols, R. H., "A Two-Equation Model for Compressible Flows," AIAA Paper 90-0494, Jan. 1990.

18. Gerolymos, G. A., "Implicit Multiple-Grid Solution of the Compressible Navier-Stokes Equations Using k-ε Turbulence Closure," *AIAA Journal*, Vol. 28, No. 10, Oct. 1990, pp. 1707-1717.

19. Nichols, R. H., "Calculation of the Flow in a Circular S-Duct Inlet," AIAA Paper 91-0174, Jan. 1991.

20. Kunz, R. F., and Lakshminarayana, B., "Explicit Navier-Stokes Computation of Cascade Flows Using the k-ε Turbulence Model," *AIAA Journal*, Vol. 30, No. 1, Jan. 1992, pp. 13-22.

21. Ladd, J. A., and Kral, L. D., "Development and Application of a Zonal k-e Turbulence Model for Complex 3-D Flowfields," AIAA Paper 92-3176, July 1992.

22. Degani, D., and Schiff, L. B., "Computation of Turbulent Supersonic Flows around Pointed Bodies Having Crossflow Separation," *Journal of Computational Physics*, Vol. 66, 1986, pp. 173-196.

23. Lam, C. K. G., and Bremhorst, K., "A Modified Form of the k-e Model for Predicting Wall Turbulence," *Journal of Fluids Engineering*, Vol. 103, Sept. 1981, pp. 456-460.

24. Myong, H. K., and Kasagi, N., "A New Approach to the Improvement of k-ε Turbulence Model for Wall-Bounded Shear Flows," *JSME International Journal*, Series II, Vol. 33, No. 1, 1990, pp. 63-72.

25. So, R. M. C., Zhang, H. S., and Speziale, C. G., "Near-Wall Modeling of the Dissipation Rate Equation," *AIAA Journal*, Vol. 29, No. 12, Dec. 1991, pp. 2069-2076.

26. 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.

27. Coakley, T. J. and Huang, P. G., "Turbulence Modeling for High Speed Flows," AIAA Paper 92-0436, Jan. 1992.

28. Rodi, W., "Experience with Two-Layer Models Combining the k-ε Model with a One-Equation Model Near the Wall," AIAA Paper 91-0216, Jan. 1991.

29. Hoffmann, K. A., *Computational Fluid Dynamics for Engineers*, Engineering Educational System, Austin, Texas, 1989.

30. Klebanoff, P. S., "Characteristics of Turbulence in a Boundary Layer with Zero Pressure Gradient," NACA Report 1247, May 1953.

31. Coles, D. E., and Hirst, E. A., Editors, *Computation of Turbulent Boundary Layers-1968* AFOSR-IFP-Stanford Conference, Vol. II, Stanford University, Aug. 1968.

32. Daily, J. W., and Harleman, D. R. F., *Fluid Dynamics*, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1966.

33. Hopkins, E. J., and Inouye, M., "An Evaluation of Theories for Predicting Turbulent Skin Friction and Heat Transfer on Flat Plates at Supersonic and Hypersonic Mach Numbers," *AIAA Journal*, Vol. 9, No. 6, June 1971, pp. 993-1003.

34. Kim, K. S., and Settles, G. S., "Skin Friction Measurements by Laser Interferometry in Swept Shock Wave/Turbulent Boundary-Layer Interactions," AIAA Paper 88-0497, Jan. 1988.

35. Kim, K. S., Lee, Y., Alvi, F. S., Settles, G. S., and Horstman, C. C., "Laser Skin Friction Measurements and CFD Comparison of Weak-to-Strong Swept Shock/Boundary-Layer Interactions," AIAA Paper 90-0378, Jan. 1990.

36. Kim, K. S., and Settles, G. S., "Skin Friction Measurements by Laser Interferometry in Swept Shock/Boundary-Layer Interactions," *AIAA Journal*, Vol. 28, No. 1, Jan. 1990, pp. 133-139.

37. Settles, G. S., and Dodson, L. J., "Hypersonic Shock/Boundary-Layer Interaction Database," NASA CR 177577, April 1991.

38. White, F. M., *Viscous Fluid Flow*, McGraw-Hill Publishing Company, 1974.

39. Schlichting, H., *Boundary-Layer Theory*, Seventh Edition, McGraw-Hill Book Company, 1979.

40. Taylor, A. M. K. P., Whitelaw, J. H., and Yianneskis, M., "Developing Flow in S-Shaped Ducts I - Square Cross-Section Duct," NASA CR 3550, May 1982.

41. Avva R., Smith C., and Singhal A., "Comparative Study of High and Low Reynolds Number Versions of k-ε Models," AIAA Paper 90-0246, Jan. 1990.

42. Kim, K. S. and Settles, G. S., "Skin Friction Measurements by Laser Interferometry in Swept Shock/Boundary-Layer Interactions," *AIAA Journal*, Vol. 28, No. 1, January 1990, pp. 133-139.

43. Settles, G. S. and Kimmel, R. L., "Similarity Conditions for Conical Shockwave Turbulent Boundary Layer Interactions," AIAA Paper 84-1557, June 1984.

FIGURES

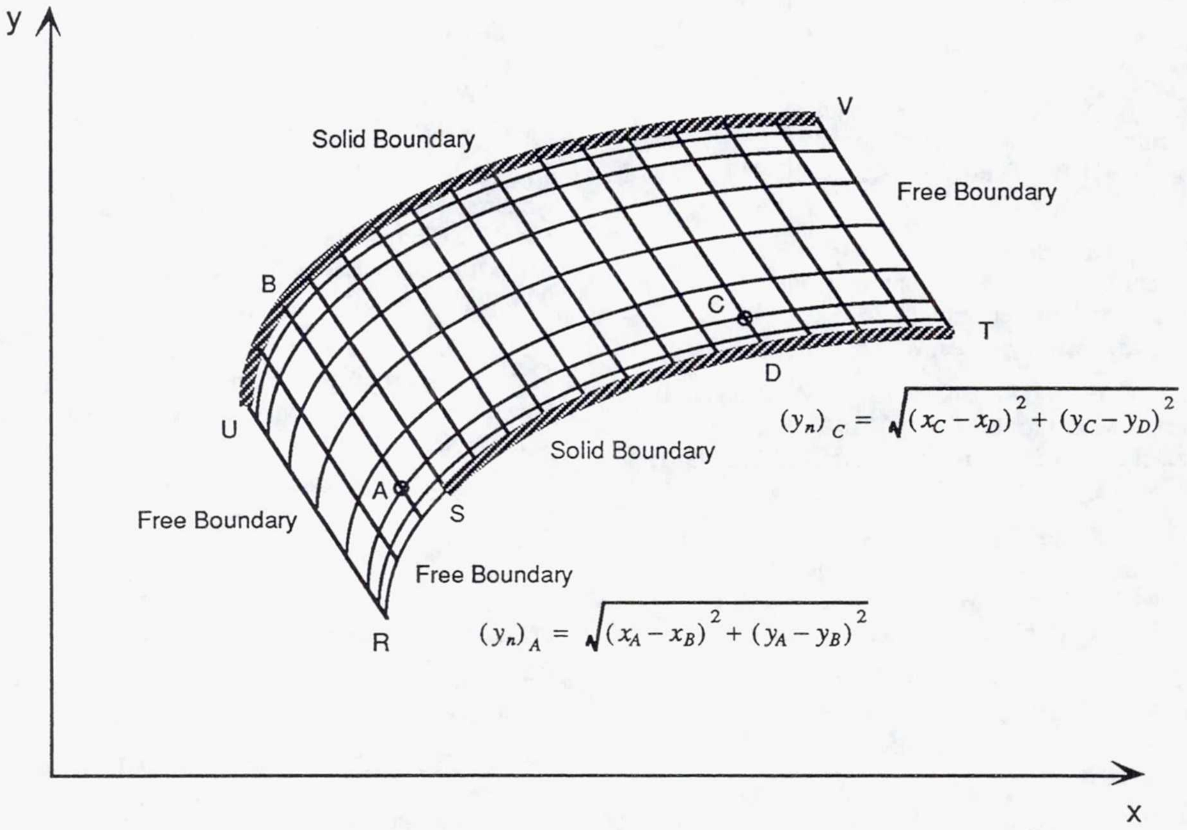
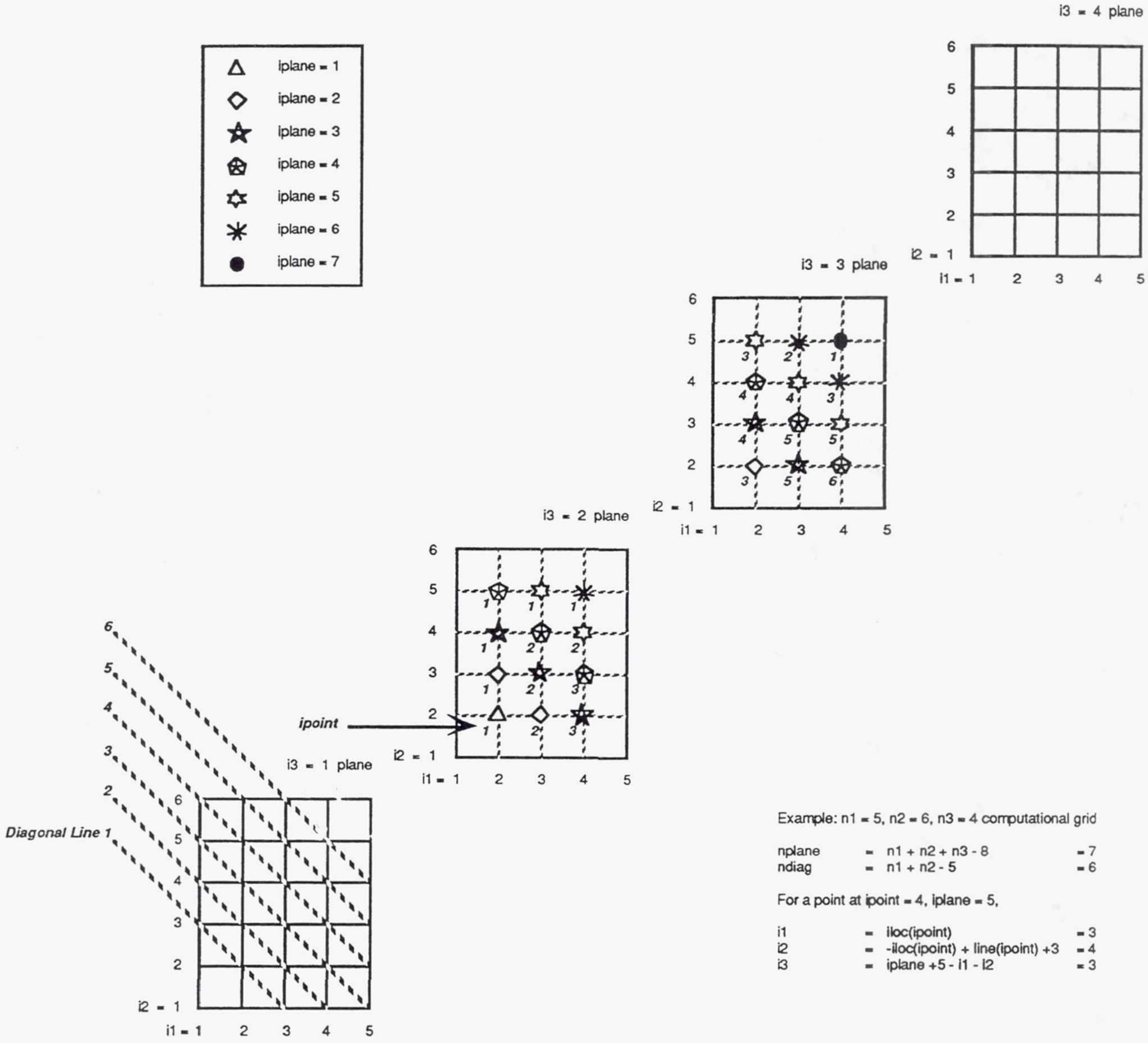


Fig. 1 Computation of y_n for multiple solid boundaries



Example: n1 = 5, n2 = 6, n3 = 4 computational grid

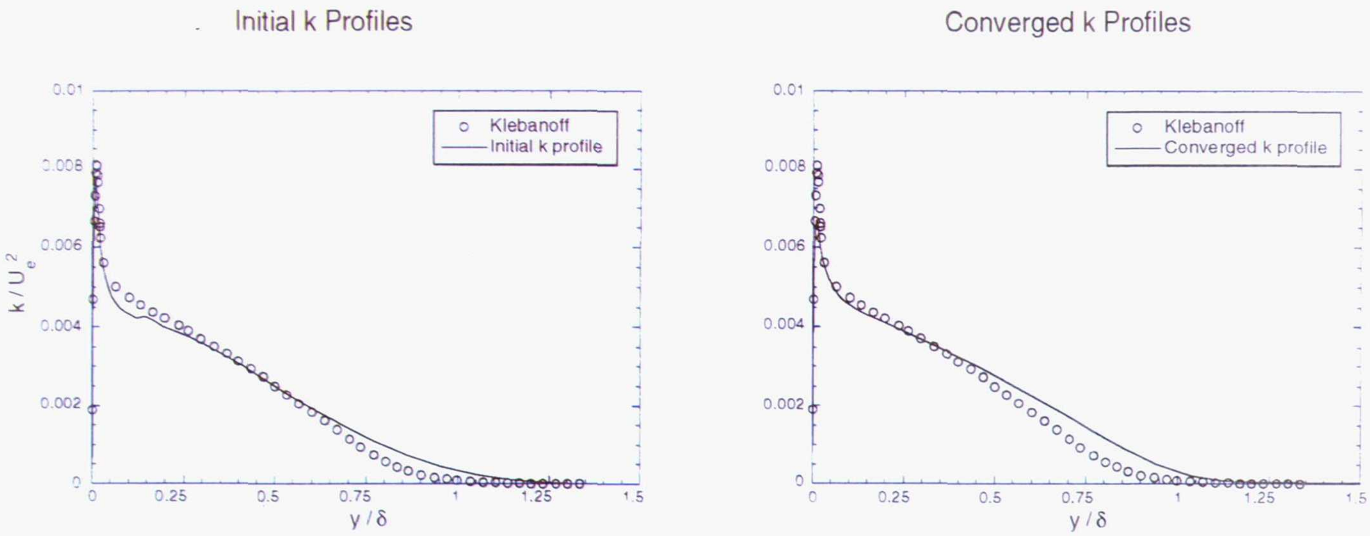
nplane = n1 + n2 + n3 - 8 = 7
ndiag = n1 + n2 - 5 = 6

For a point at ipoint = 4, iplane = 5,

i1 = iloc(ipoint) = 3
i2 = -iloc(ipoint) + line(ipoint) +3 = 4
i3 = iplane +5 - i1 - i2 = 3

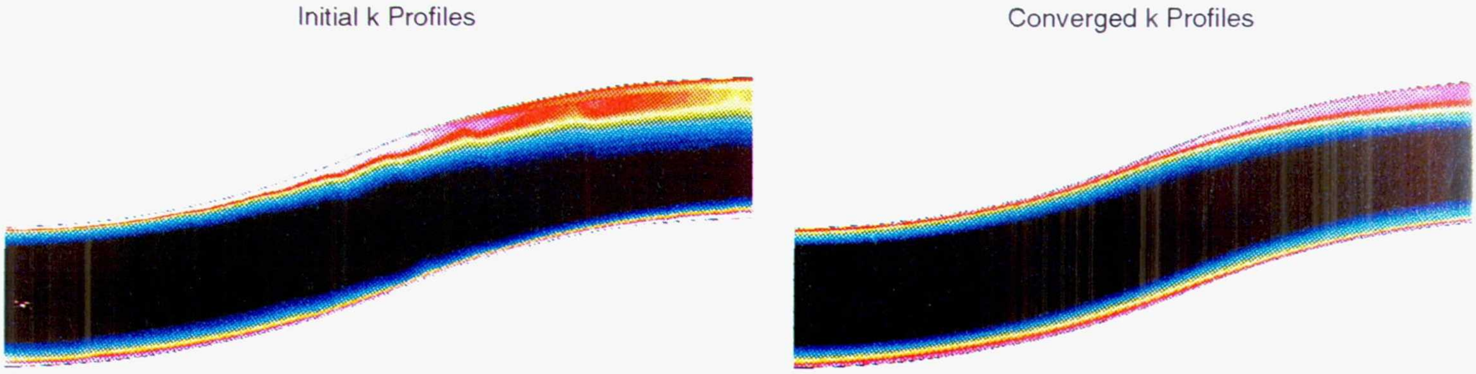
Fig. 2 Grid addressing scheme for the implementation of the L-U marching algorithm

Page intentionally left blank


Initial k Profiles

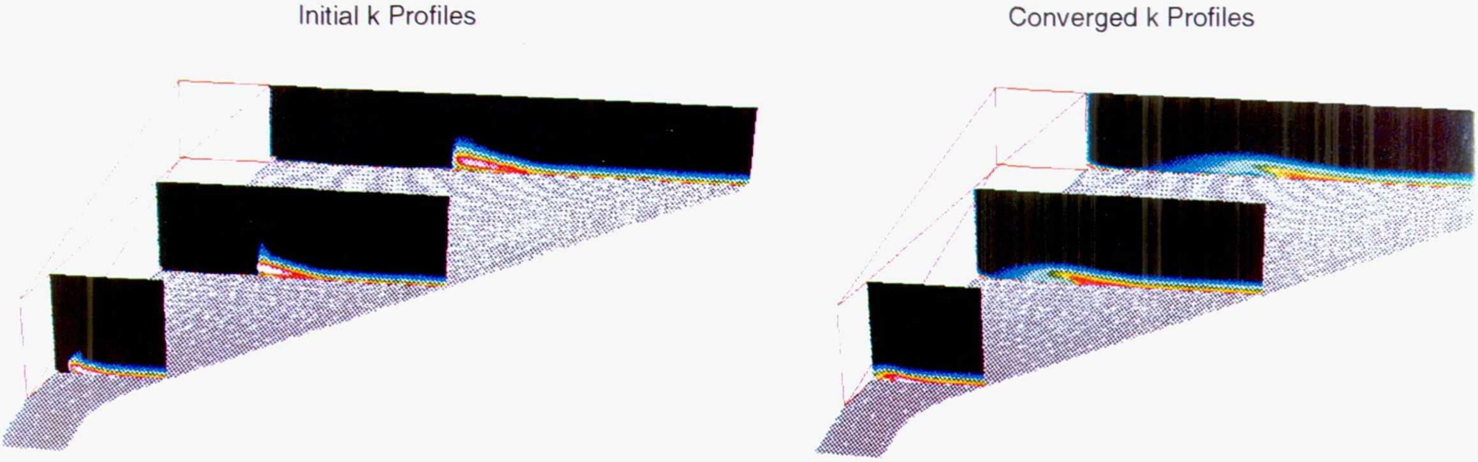
Converged k Profiles

Flat plate turbulent boundary layer


Initial k Profiles

Converged k Profiles

Turbulent developing S-duct

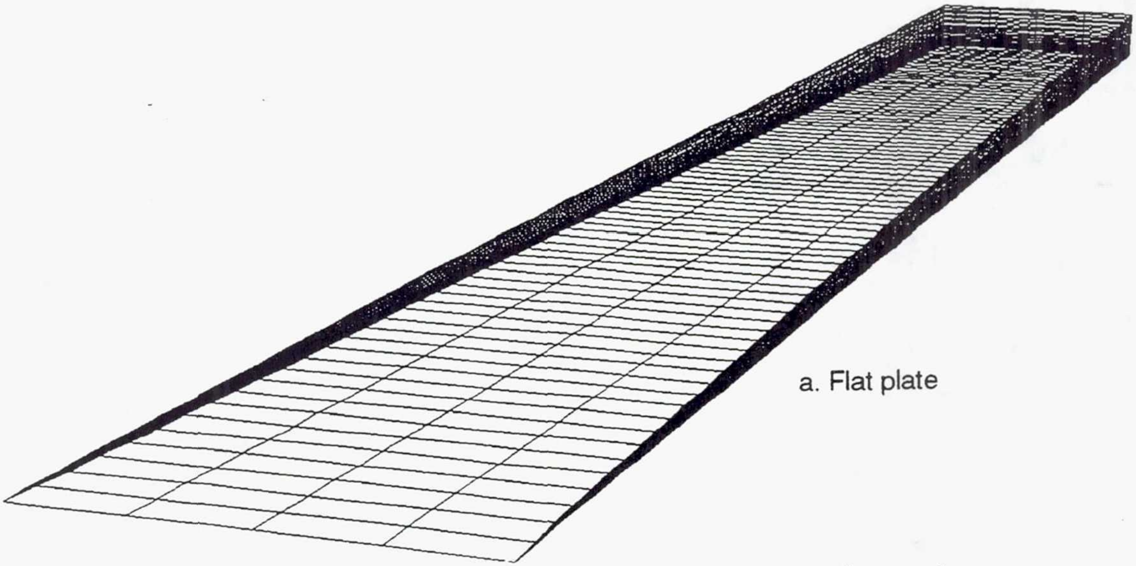

Initial k Profiles

Converged k Profiles

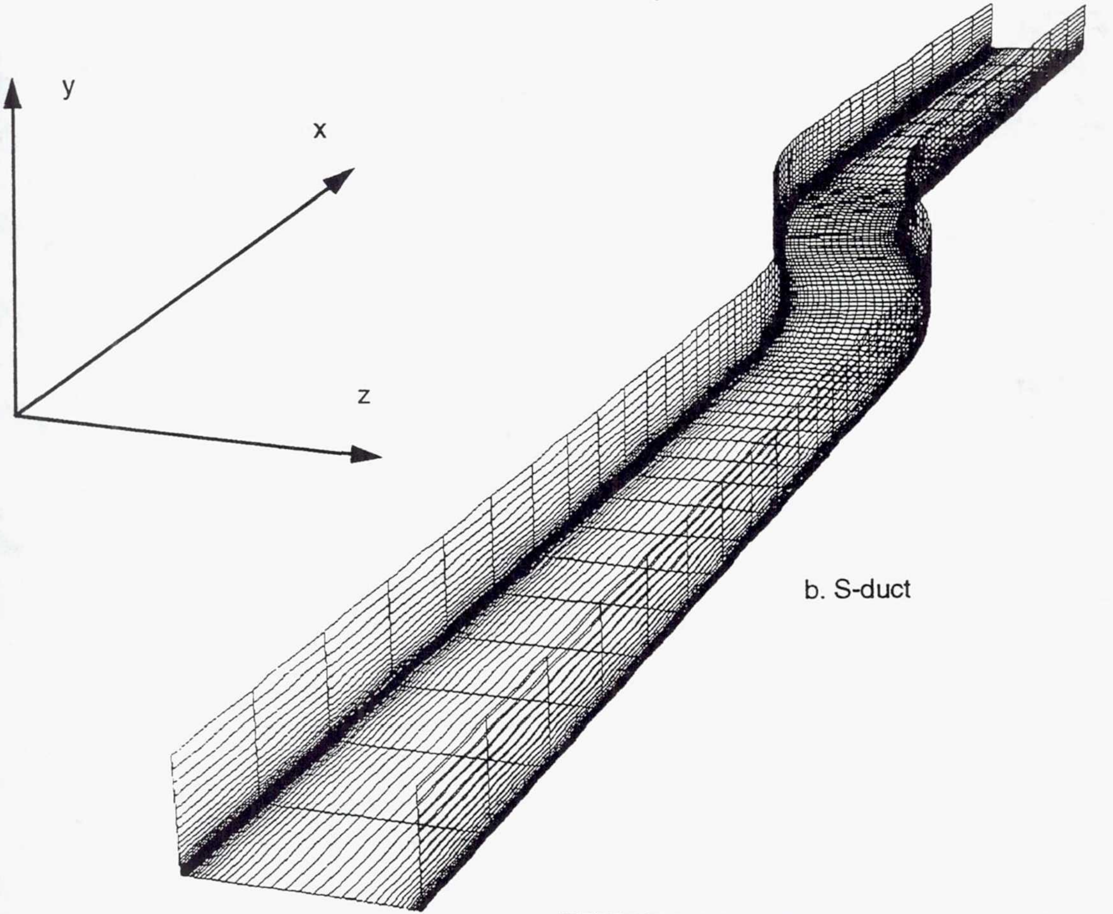
Glancing shock wave/turbulent boundary layer interaction

Fig. 3 Automatic initialization of the k-ε models

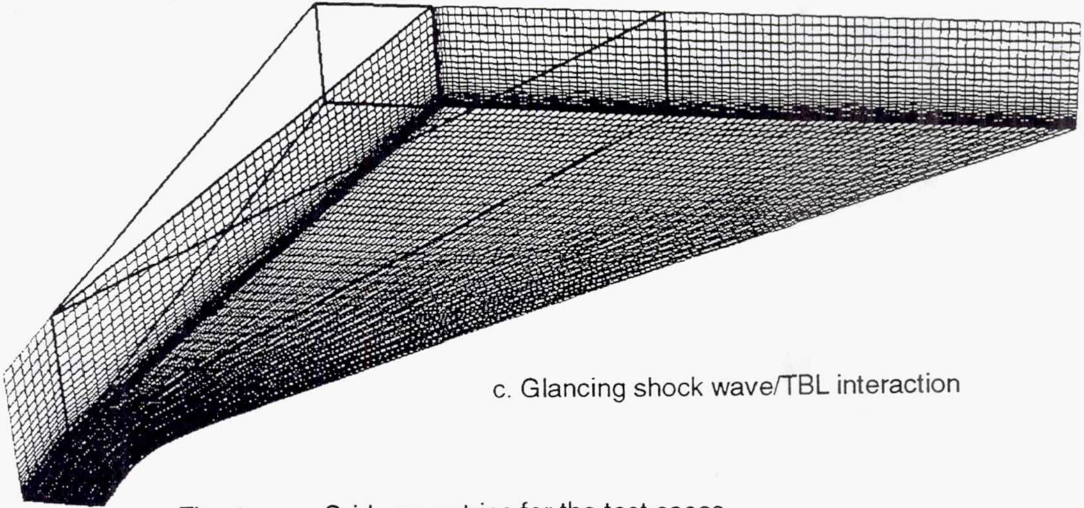
Page intentionally left blank



a. Flat plate

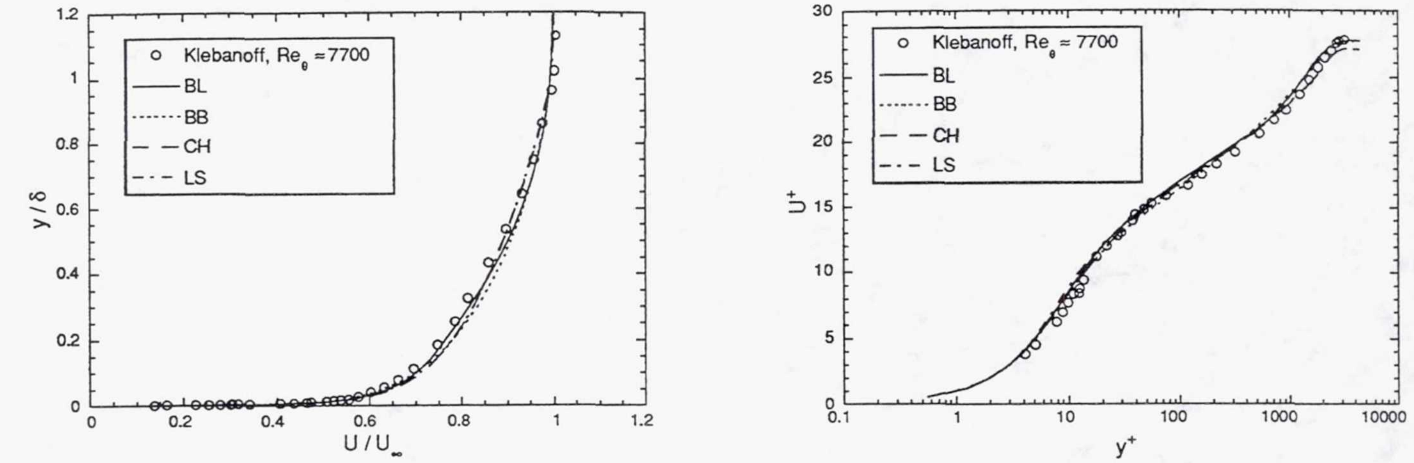


b. S-duct



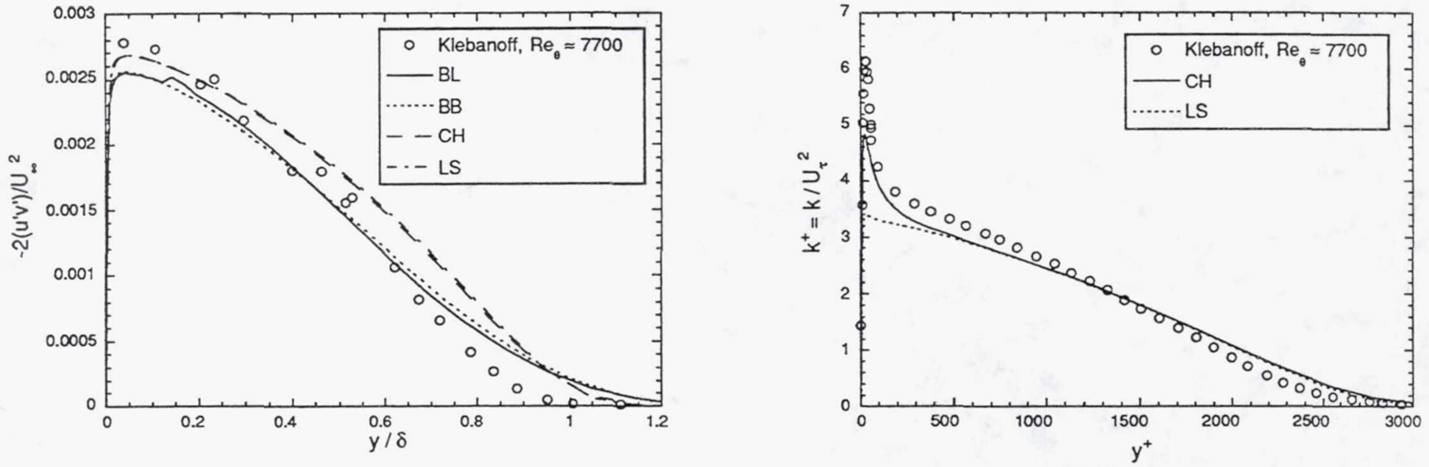
c. Glancing shock wave/TBL interaction

Fig. 4 Grid geometries for the test cases



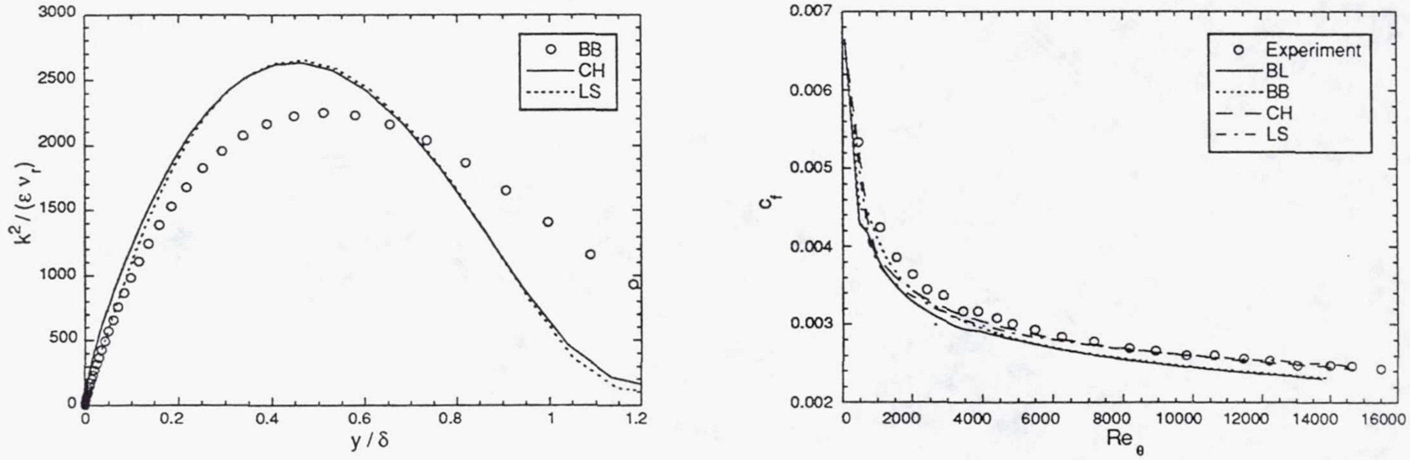
a. Velocity profile

b. Nondimensional velocity profile



c. Reynolds shearing stress profile

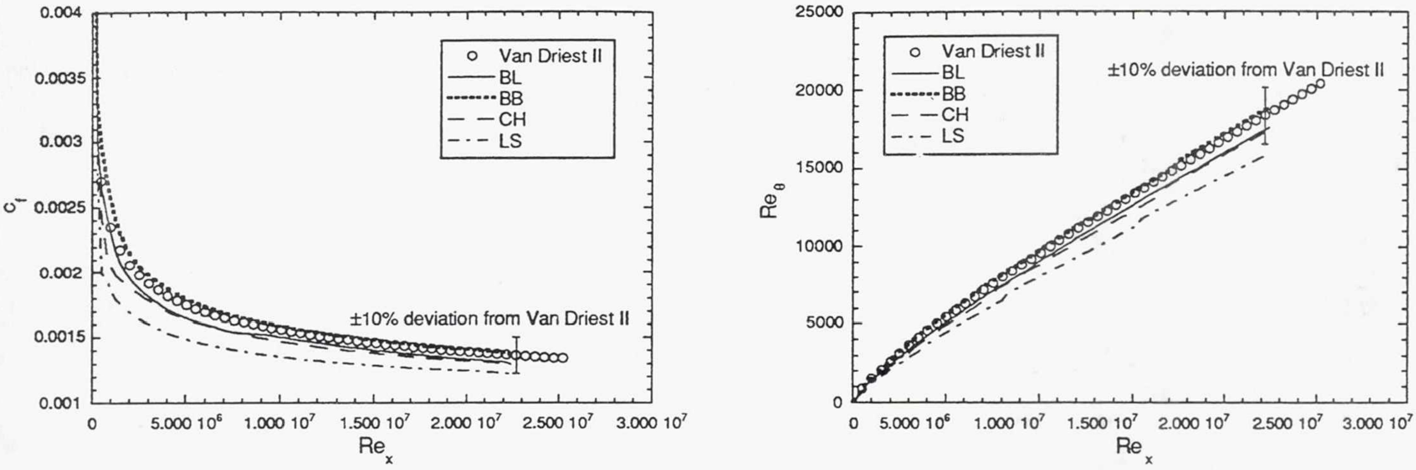
d. Turbulent kinetic energy profile



e. Baldwin-Barth variable k2/e profile

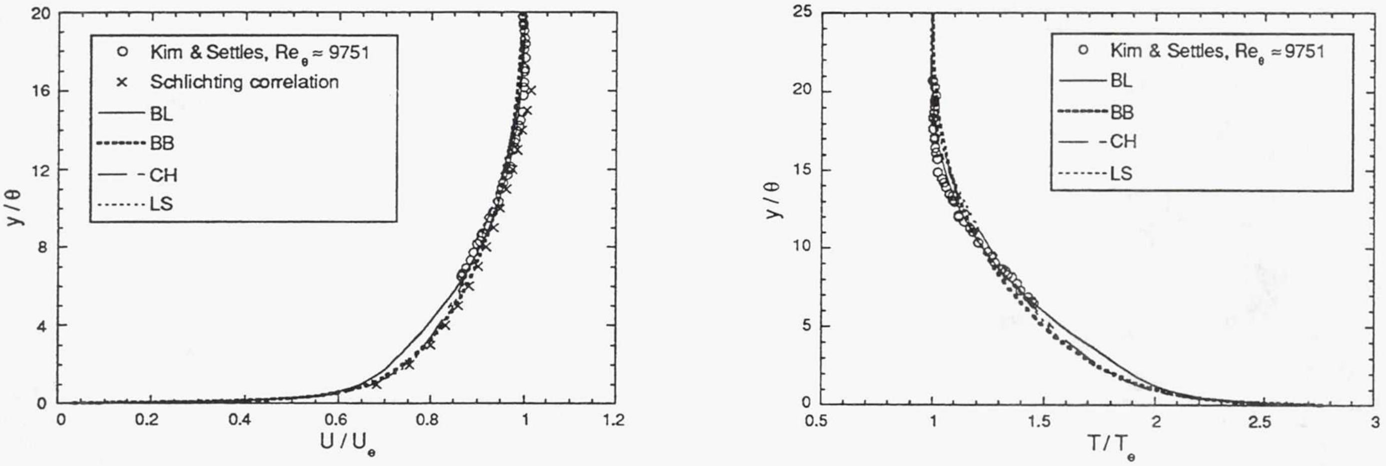
f. Local skin friction coefficient distribution

Fig. 5 Incompressible turbulent boundary layer results



a Local skin friction distribution

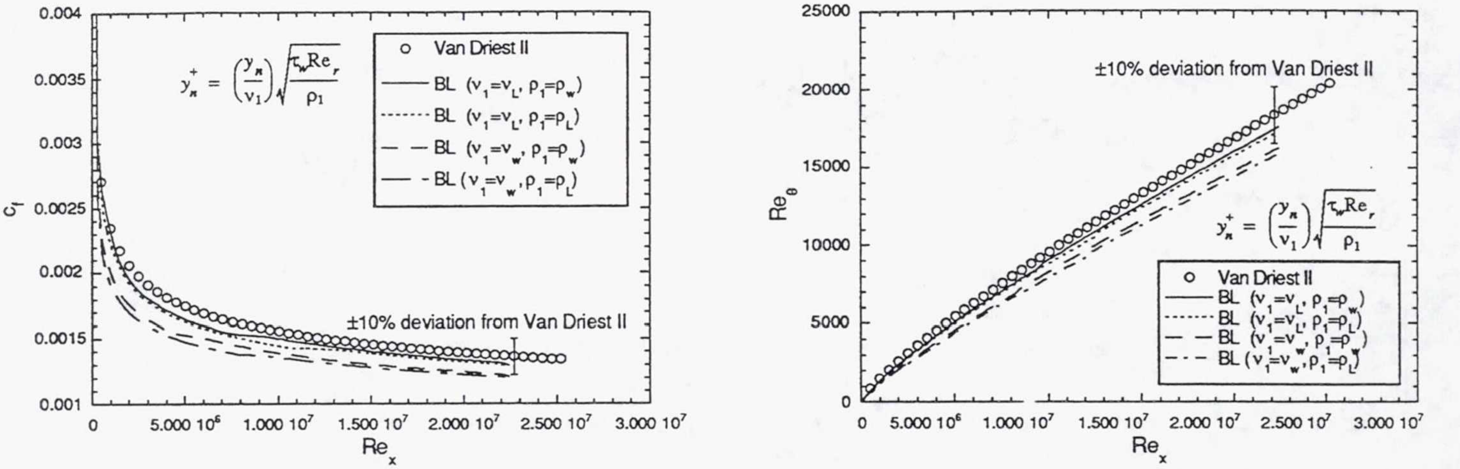
b. Momentum thickness distribution



c. Velocity profile

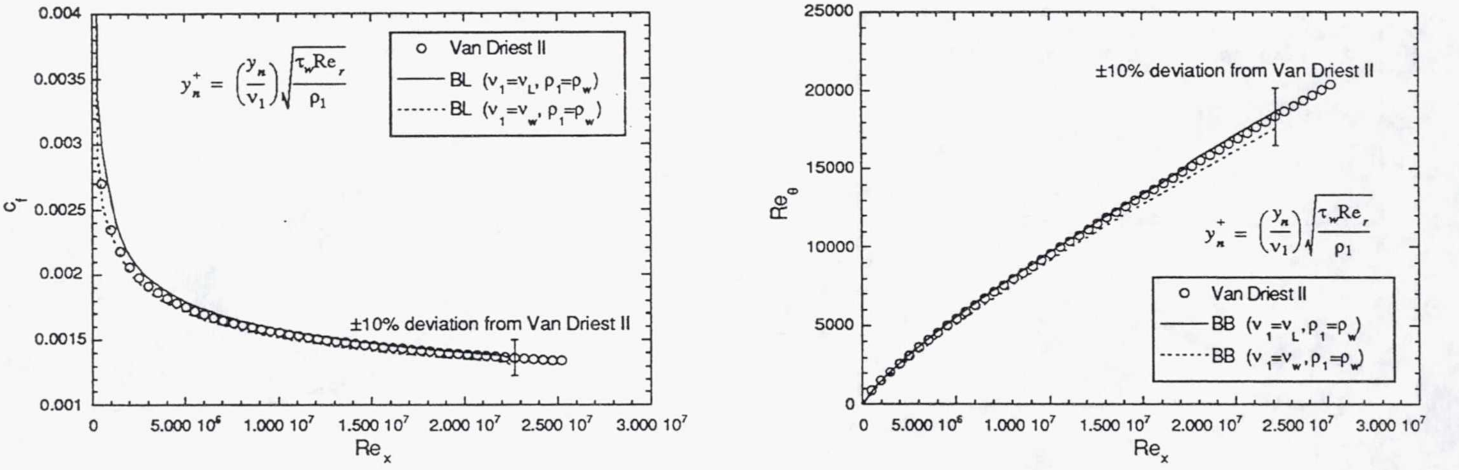
d. Temperature profile

Fig. 6 Compressible flat plate turbulent boundary layer results



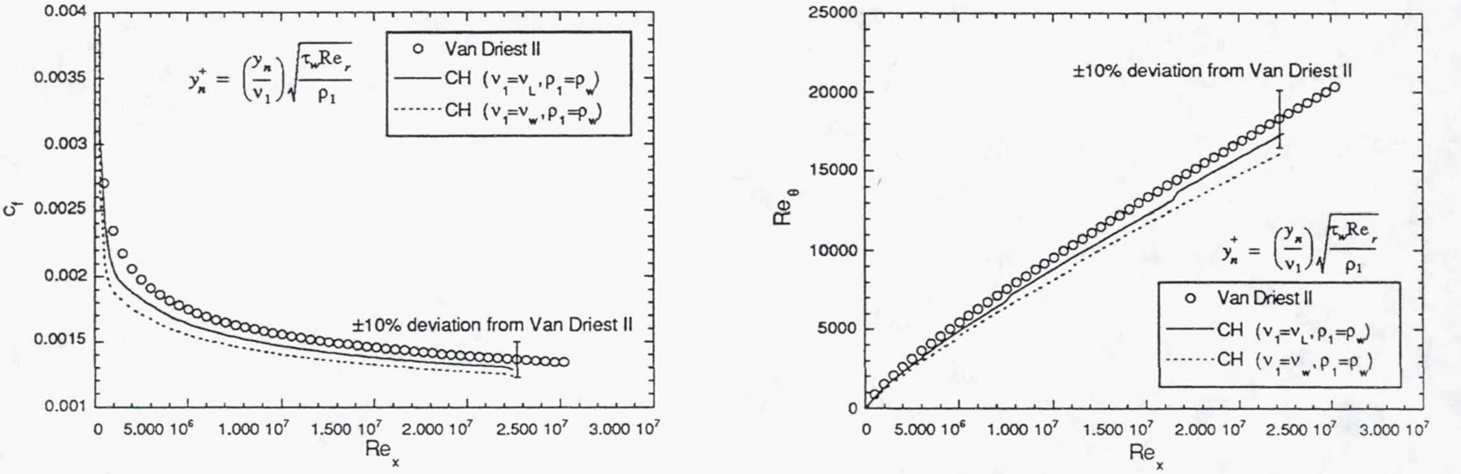
a. Skin friction, BL model

b. Momentum thickness, BL model



c. Skin friction, BB model

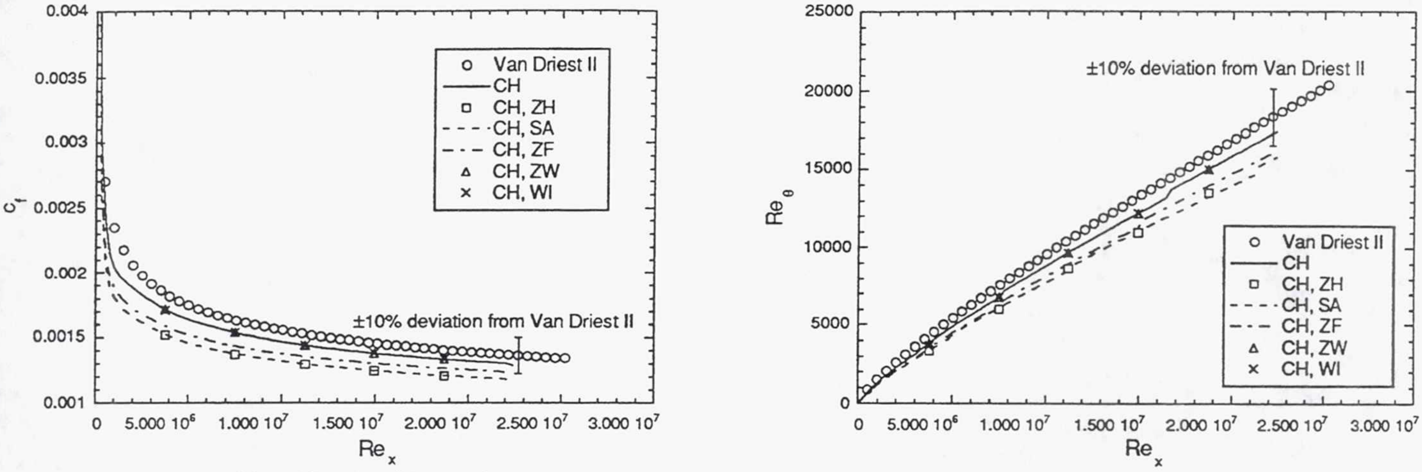
d. Momentum thickness, BB model



e. Skin friction, CH model

f. Momentum thickness, CH model

Fig. 7 Effects of method of y_n^+ computation on compressible flat plate turbulent boundary layer results



a. Skin friction, CH model

b. Momentum thickness, CH model

Fig. 8 Effects of k-ε compressibility corrections on compressible flat plate turbulent boundary layer results

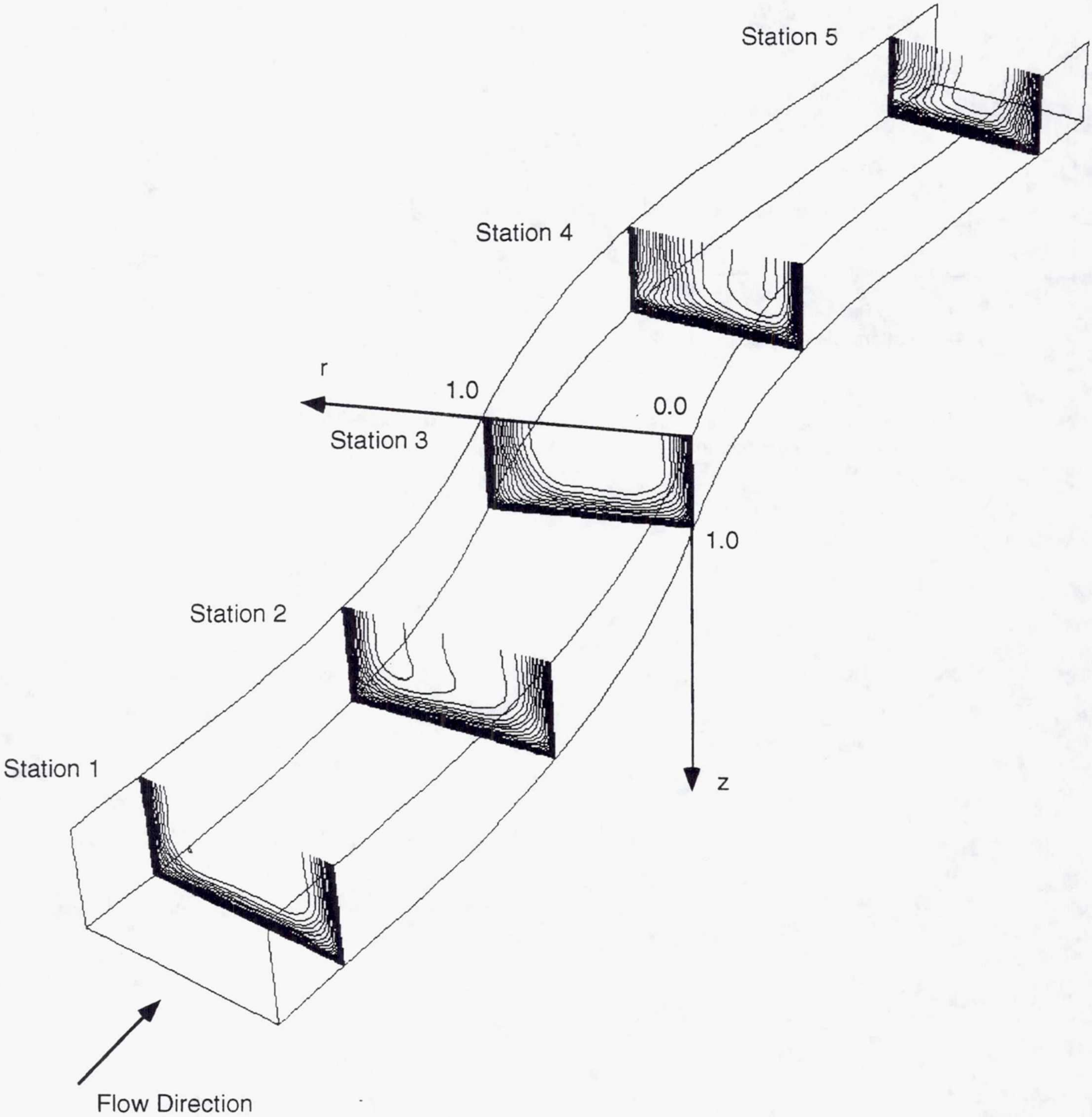


Fig. 9 Streamwise velocity contours and the general geometry for the turbulent developing S-duct test case

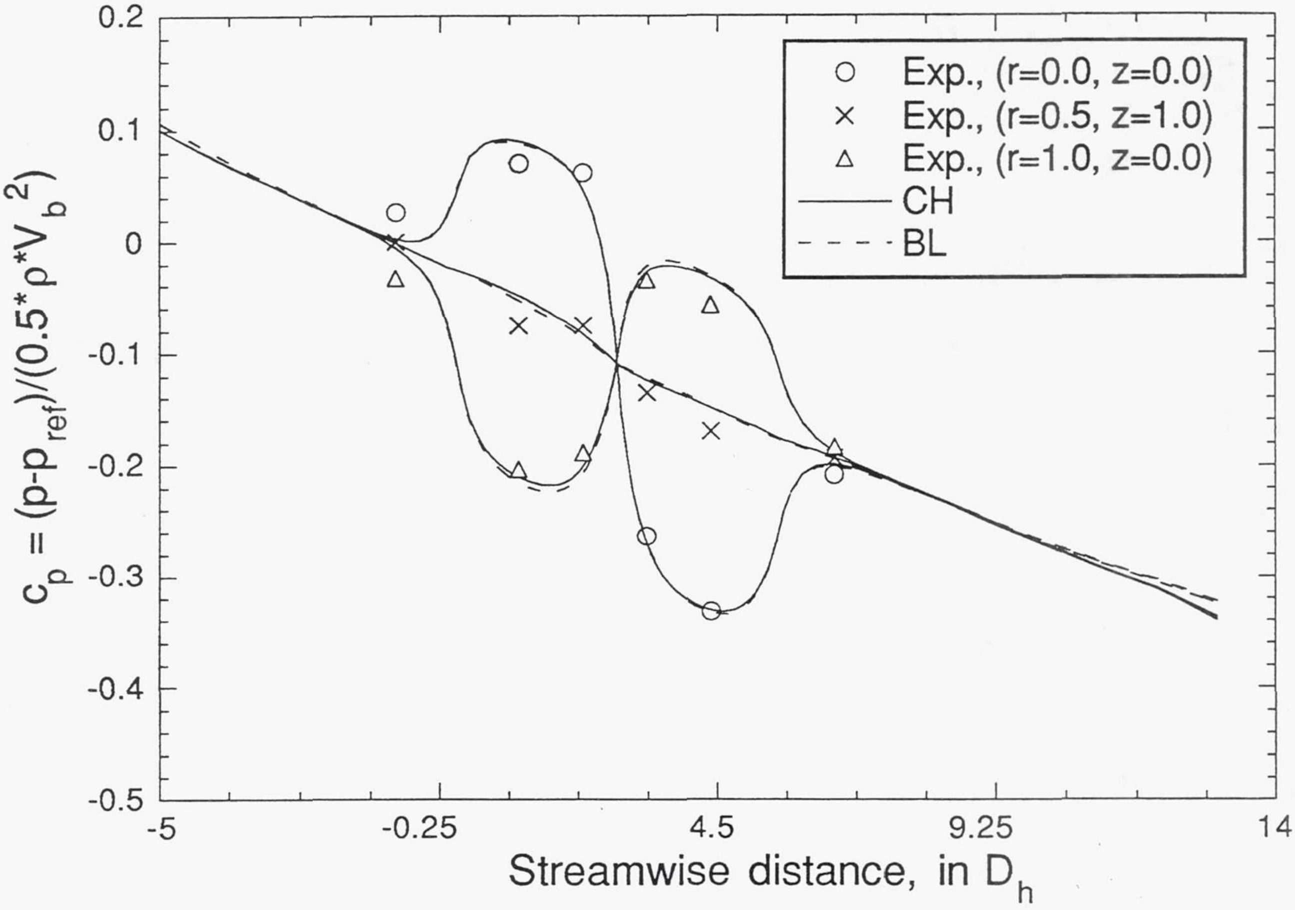


Fig. 10 Wall pressure distribution, turbulent developing S-duct flow

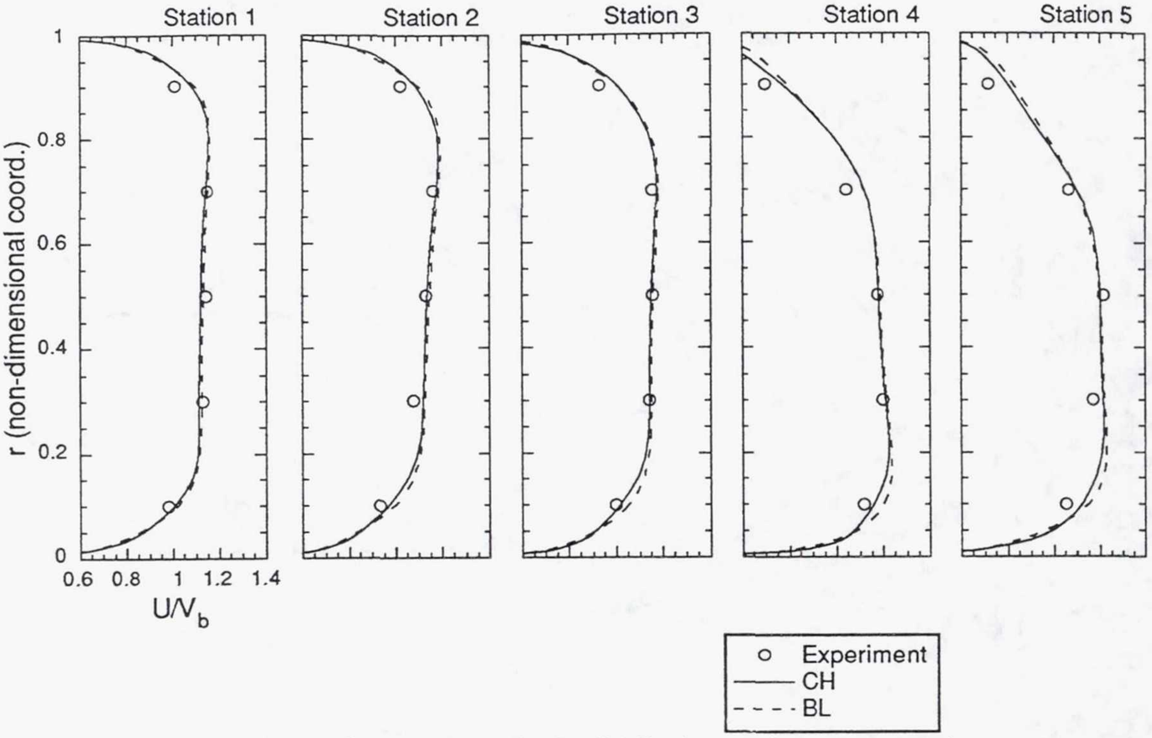


Fig. 11 Streamwise velocity distribution on the symmetry plane — turbulent developing S-duct flow

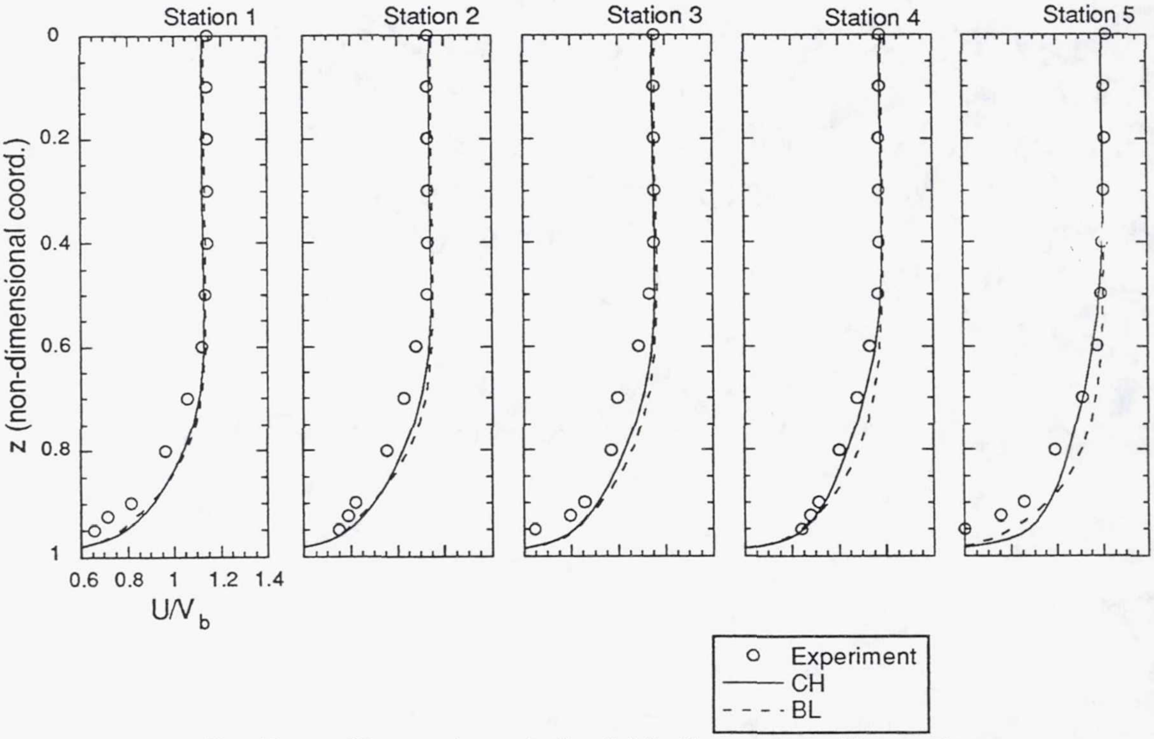


Fig. 12 Streamwise velocity distribution on the midspan plane — turbulent developing S-duct flow

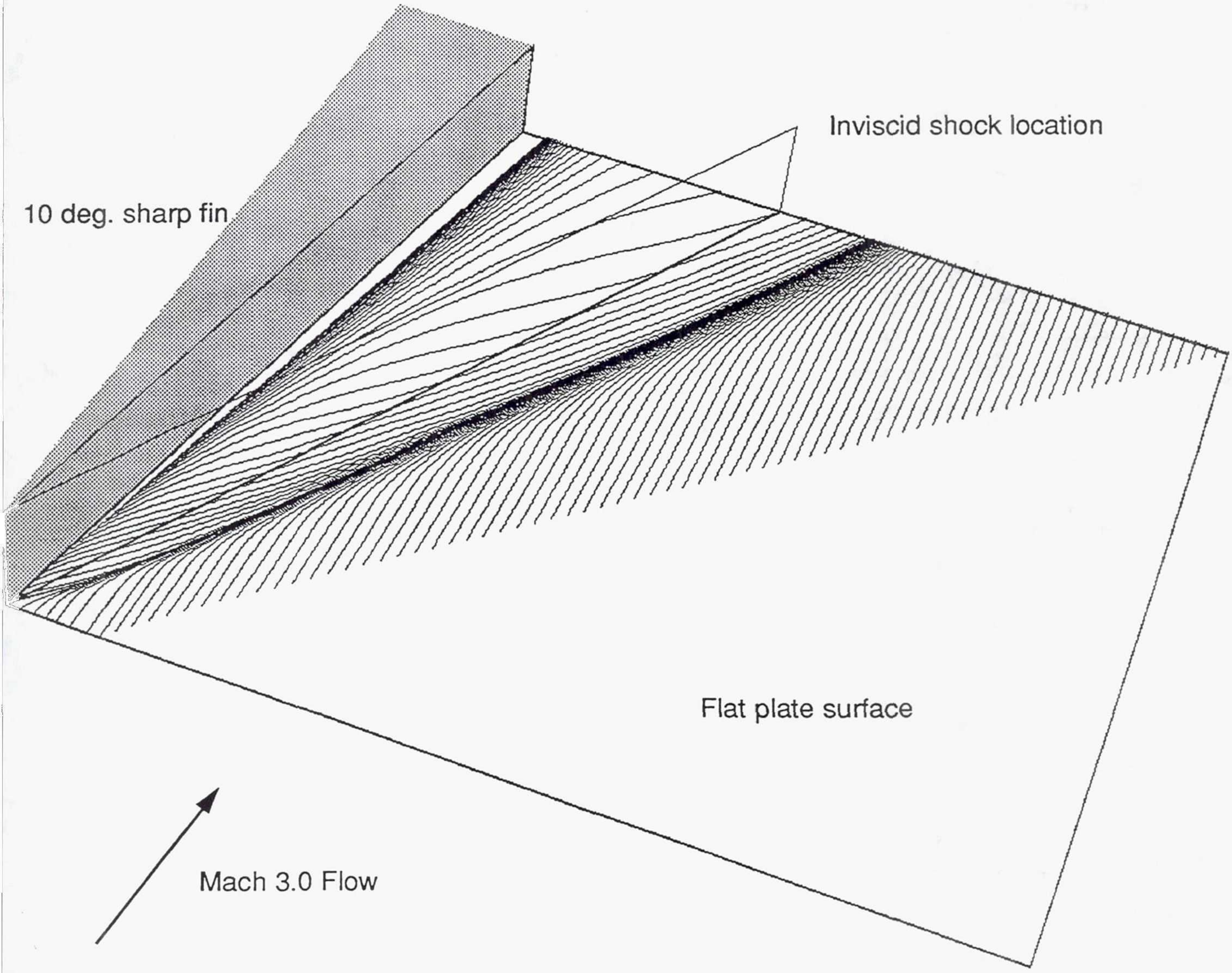
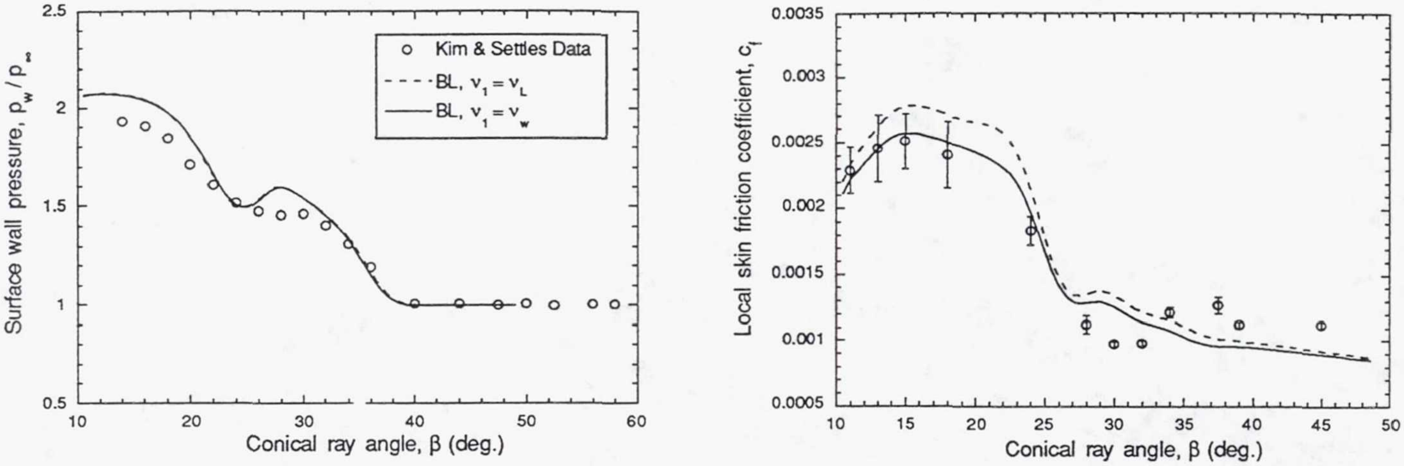
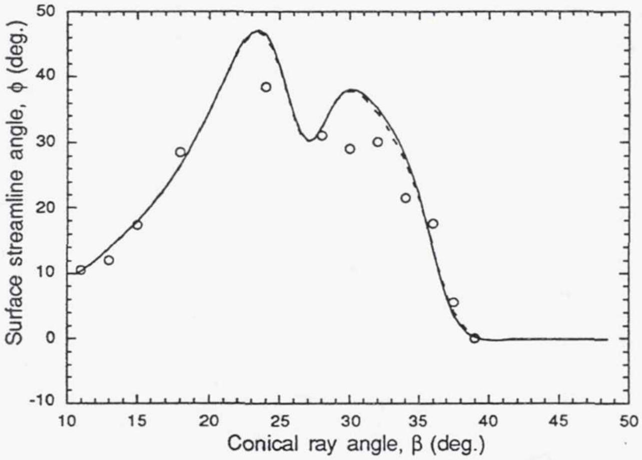


Fig. 13 Computed surface flow streamlines on the flat plate and the general geometry— Glancing shock wave / turbulent boundary layer interaction



a. Surface wall pressure, BL model

b. Local skin friction, BL model



c. Surface streamline angle, BL model

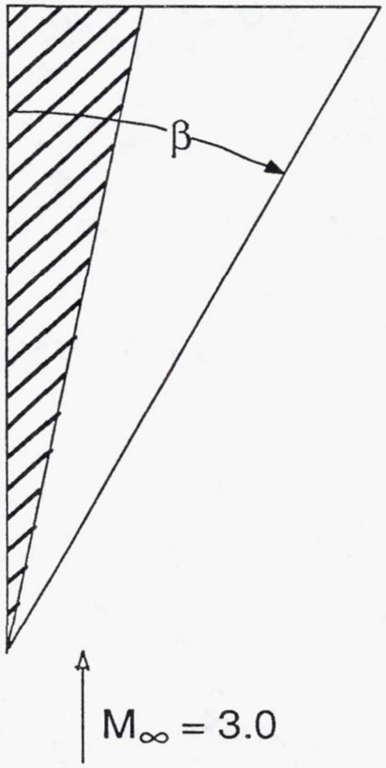
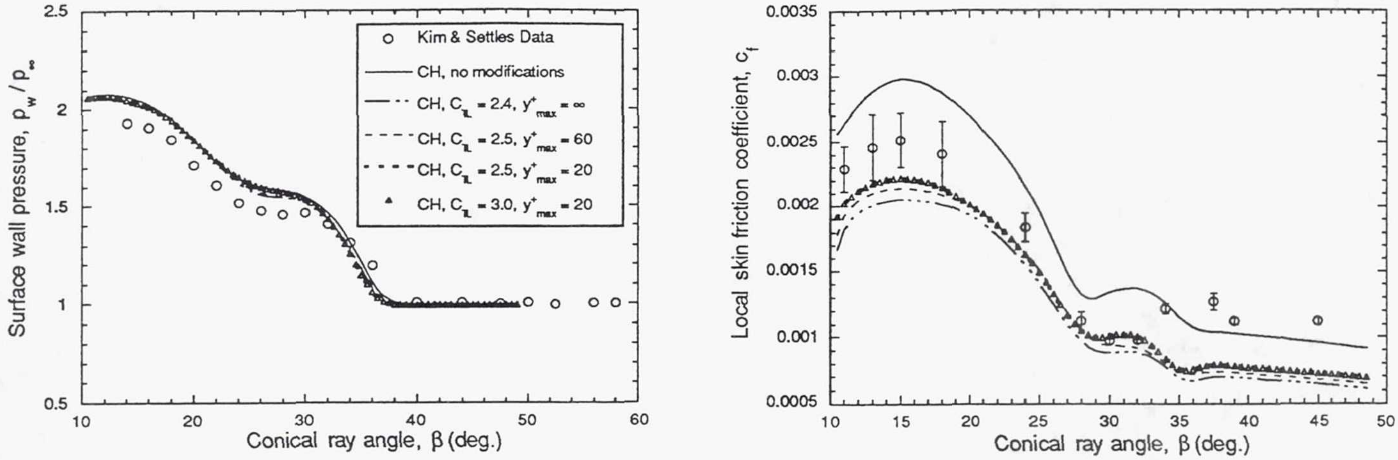
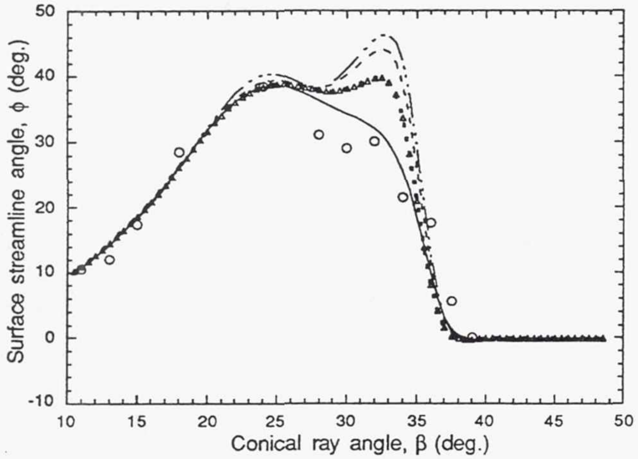


Fig. 14 Glancing shock wave/turbulent boundary layer interaction results — The BL turbulence model with effects of y^+ computations



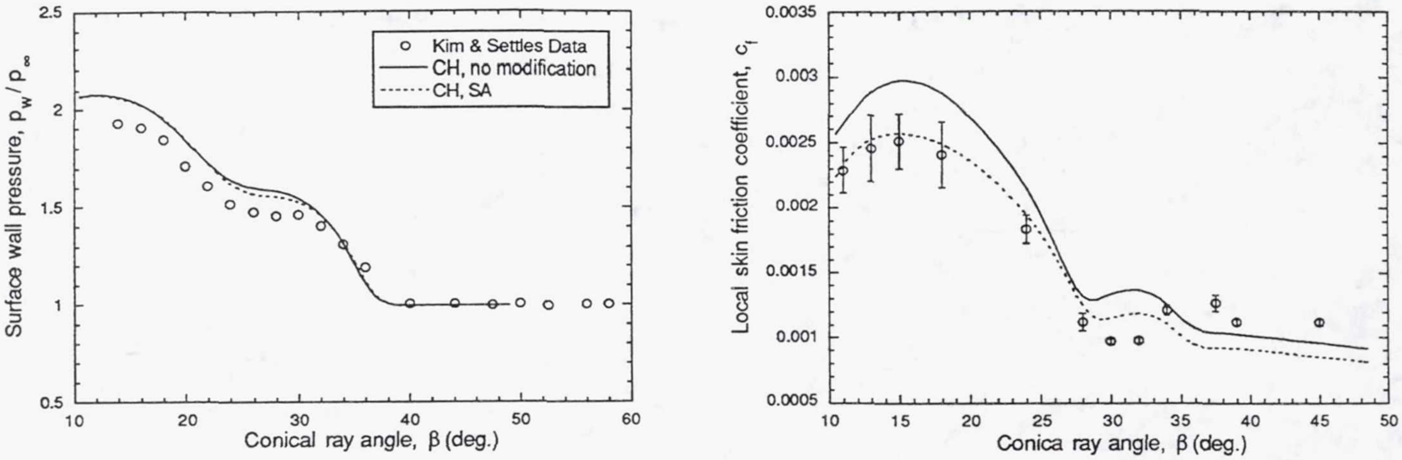
a. Surface wall pressure, CH model

b. Local skin friction, CH model



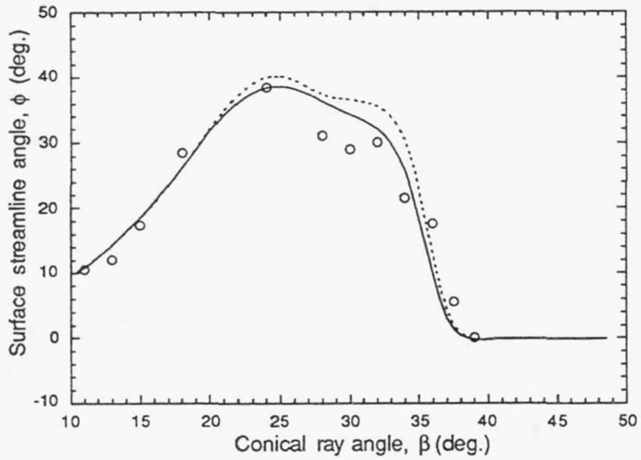
c. Surface streamline angle, CH model

Fig. 15 Glancing shock wave/turbulent boundary layer interaction results — The CH k-ε model with effects of the turbulent length scale correction



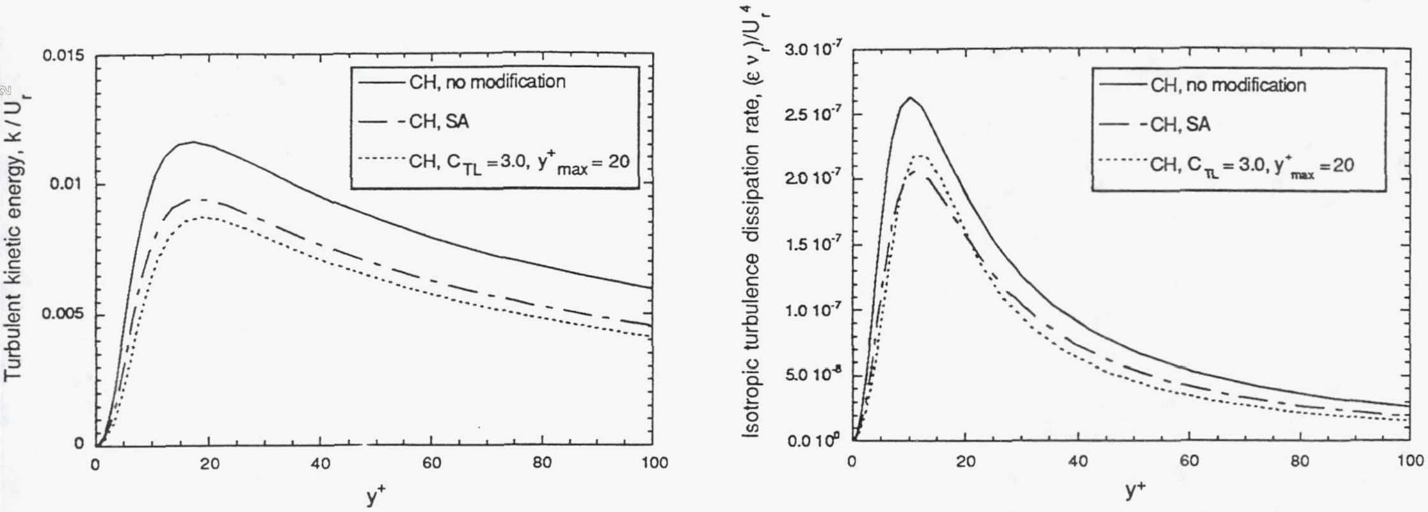
a. Surface wall pressure, CH model

b. Local skin friction, CH model



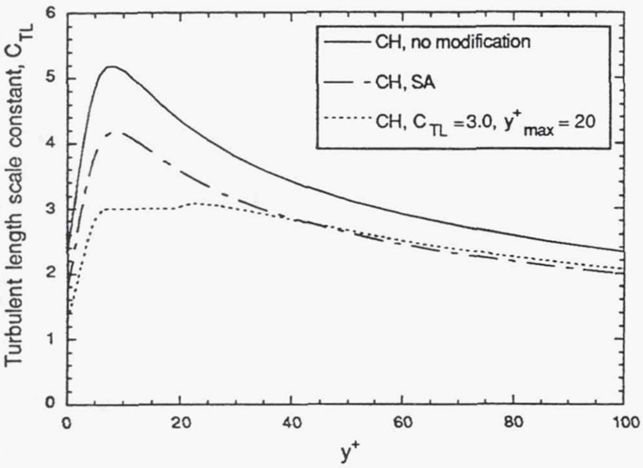
c. Surface streamline angle, CH model

Fig. 16 Glancing shock wave/turbulent boundary layer interaction results — The CH k-ε model and the effects of the Sarkar compressibility correction



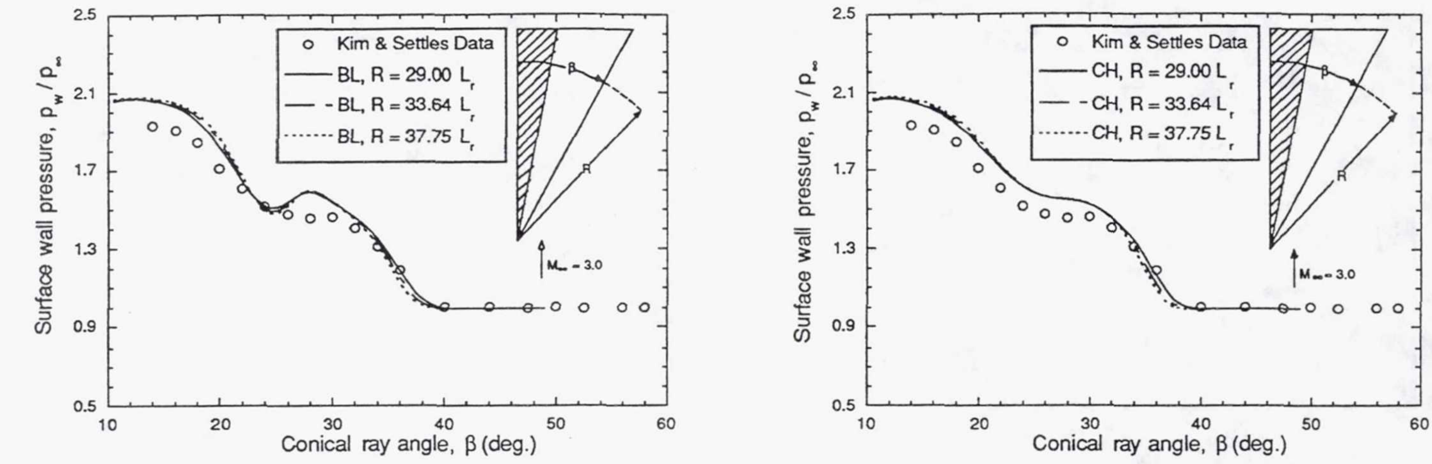
a. Turbulence kinetic energy

b. Isotropic turbulent dissipation rate



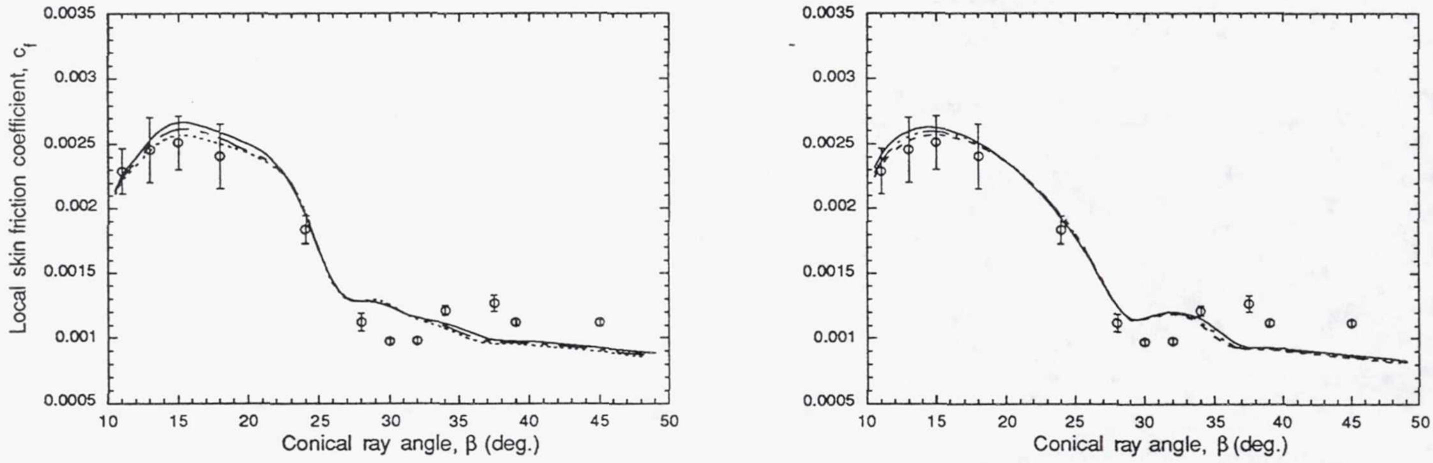
c. Turbulent length scale constant

Fig. 17 Glancing shock wave/turbulent boundary layer interaction results — Effects of the Sarkar compressibility and length scale corrections on some turbulence quantities



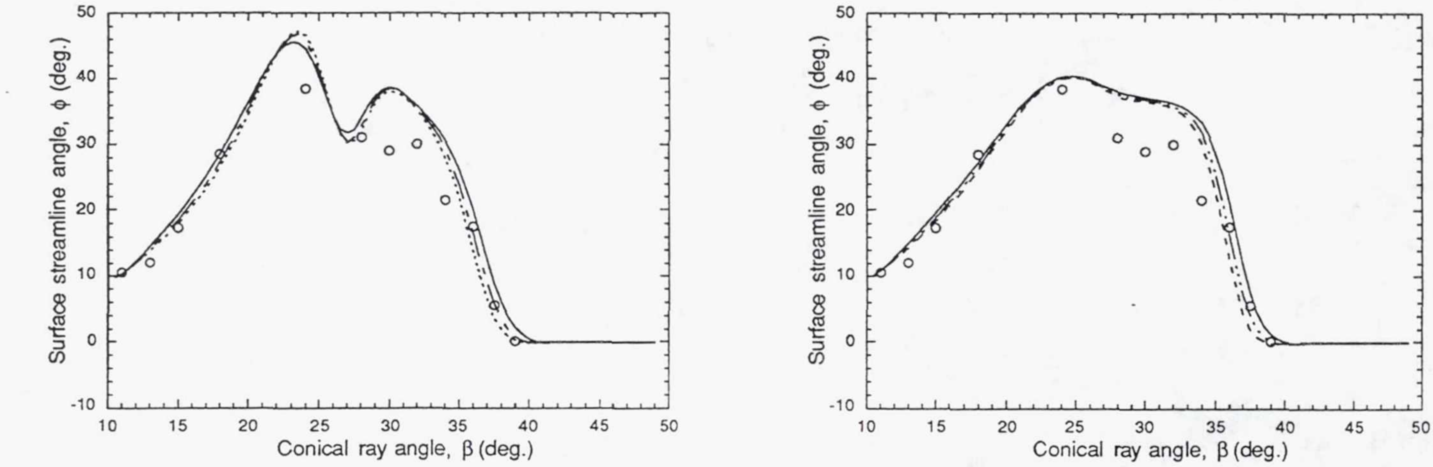
a. Surface pressure distribution, BL model

d. Surface pressure distribution, CH model



b. Local skin friction, BL model

e. Local skin friction, CH model



c. Surface streamline angle, BL model

f. Surface streamline angle, CH model

Fig. 18 Glancing shock wave / turbulent boundary layer interaction results — Quasiconical similarities in computed flow properties

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** November 1993	**3. REPORT TYPE AND DATES COVERED** Technical Memorandum

4. TITLE AND SUBTITLE Some Practical Turbulence Modeling Options for Reynolds-Averaged Full Navier-Stokes Calculations of Three-Dimensional Flows	**5. FUNDING NUMBERS** WU-505-62-52
6. AUTHOR(S) Trong T. Bui	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191	**8. PERFORMING ORGANIZATION REPORT NUMBER** E-8231
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, D.C. 20546-0001	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER** NASA TM-106399 AIAA-93-2964

11. SUPPLEMENTARY NOTES

Prepared for the 24th Fluid Dynamics Conference sponsored by the American Institute of Aeronautics and Astronautics, Orlando, Florida, July 6-9, 1993. Responsible person, Trong T. Bui, (216) 433-5639.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 02	**12b. DISTRIBUTION CODE**

13. ABSTRACT ***(Maximum 200 words)***

New turbulence modeling options recently implemented for the 3-D version of Proteus, a Reynolds-averaged compressible Navier-Stokes code, are described. The implemented turbulence models include: the Baldwin-Lomax algebraic model, the Baldwin-Barth one-equation model, the Chien k-ε model, and the Launder-Sharma k-ε model. Features of this turbulence modeling package include: well documented and easy to use turbulence modeling options, uniform integration of turbulence models from different classes, automatic initialization of turbulence variables for calculations using one- or two-equation turbulence models, multiple solid boundaries treatment, and fully vectorized L-U solver for one- and two-equation models. Validation test cases include the incompressible and compressible flat plate turbulent boundary layers, turbulent developing S-duct flow, and glancing shock wave/turbulent boundary layer interaction. Good agreement is obtained between the computational results and experimental data. Sensitivity of the compressible turbulent solutions with the method of y^+ computation, the turbulent length scale correction, and some compressibility corrections are examined in detail. The test cases show that the highly optimized one- and two-equation turbulence models can be used in routine 3-D Navier-Stokes computations with no significant increase in CPU time as compared with the Baldwin-Lomax algebraic model.

14. SUBJECT TERMS Turbulence modeling; Navier-Stokes; 3-D; L-U factorization	**15. NUMBER OF PAGES** 40
	16. PRICE CODE A03

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT**

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102