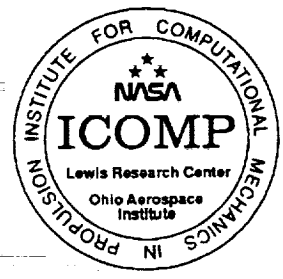
NASA Technical Memorandum 106781
ICOMP–94–28

Effects of Artificial Viscosity on the Accuracy of High-Reynolds-Number k-ε Turbulence Model

Tawit Chitsomboon
Institute for Computational Mechanics in Propulsion
Lewis Research Center
Cleveland, Ohio

November 1994

National Aeronautics and
Space Administration



Effects of Artificial Viscosity on the Accuracy of High-Reynolds-Number k-ϵ Turbulence Model

Tawit Chitsomboon

Institute for Computational Mechanics in Propulsion
NASA Lewis Research Center, Cleveland, Ohio

Abstract

Wall functions, as used in the typical high Reynolds number k-ϵ turbulence model, can be implemented in various ways. A least disruptive method (to the flow solver) is to directly solve for the flow variables at the grid point next to the wall while prescribing the values of k and ϵ. For the centrally-differenced finite-difference scheme employing artificial viscosity (AV) as a stabilizing mechanism, this methodolgy proved to be totally useless. This is because the AV gives rise to a large error at the wall due to too steep a velocity gradient resulting from the use of a coarse grid as required by the wall function methodology. This error can be eliminated simply by extrapolating velocities at the wall, instead of using the physical values of the no-slip velocities (i.e. the zero value). The applicability of the technique used in this paper is demonstrated by solving a flow over a flat plate and comparing the results with those of experiments. It was also observed that AV gives rise to a velocity overshoot (about 1 %) near the edge of the boundary layer. This small velocity error, however, can yield as much as 10% error in the momentum thickness. A method which integrates the boundary layer up to only the edge of the boundary (in stead of to ∞) was proposed and demonstrated to give better results than the standard method.

Nomenclature

C_f = friction coefficient

C_μ = constant for μ_t relation (= 0.09)

D_4 = forth-order dissipation

E = constant in the log law (= 9.0)

i,j,k = computational coordinates directions or indices

J = Jacobian of coordinates transformation

k = turbulent kinetic energy

p = pressure

P = production rate of k

q = conservative flow-variables

Re_x = Reynolds number based on x

Re_θ = Reynolds number based on momentum thickness

u, v, w = velocities in x, y, z directions

u', v', w' = fluctuating velocities in x, y, z directions

u_τ = frictional velocity

y^+ = law of the wall coordinate

δ_2 = momentum thickness

δ_{99} = boundary layer thickness at 99% of the freestream velocity

ϵ = dissipation rate of k

ϵ_2 = 2nd order artificial viscosity coefficient

ϵ_4 = 4th order artificial viscosity coefficient

κ = Von Karmann constant (= 0.41)

λ = spectral radius of Jacobian matrix

μ = viscosity

ρ = fluid density

τ = shear stress

$\triangle_j, \nabla_j$ = forward, backward differencing in j direction

Subscripts

i, j = tensoral directions

η = computational direction normal to the solid wall

l = laminar quantity

N = normal component

t = turbulent quantity

T	= tangential component
w	= wall
$1, 2, 3$	= grid indices at wall, next to wall, etc.
∞	= free stream value

Introduction

Over the past two decades, the k-ϵ turbulence model has enjoyed a considerable popularity in Computational Fluid Dynamics. The popularity is due to the fact that the model is not too complicated to work with and also can give good solutions to many complex turbulent flow problems of engineering interest.

There have been two main methods in using the k-ϵ equations in the near wall region: One is to solve the equations directly down to the wall; the other is to employ wall functions. The first method has been known as the low Reynolds (LR) number method and the latter has been called the high Reynolds (HR) number method. LR is potentially more accurate than HR since it aims at solving the entire boundary layer, including the thin laminar sublayer. HR skips the sublayer computation altogether and starts the calculation in the universal log-law region.

In a practical sense, the feature that differentiates the two methods the most is the fact that LR requires a lot more grid points than HR in order to resolve the laminar sublayer. The thickness of the first grid point for LR is of the order of y^+=1, whereas that for HR is of the order of $y^+ = 50$ to 100. There are two adverse effects in using fine grid: 1) there will be more grid points to solve for; and 2) the time step of integration will be relatively small. The first effect is obvious since the number of grid points is directly proportional to the grid size. The second effect, however, is not so obvious. The time step of integration is proportional to the Courant-Friedrichs-Lewy (CFL) number which, in turn, is proportional to the grid size. In solving the same flow using the same CFL number, the time step is smaller for a smaller grid size. Stability of a CFD code (even an implicit one) often depends on the criterion that its CFL number be below a certain critical value. Even if a large CFL number (hence time step) can be used without an adverse effect on stability, it still does not mean that the convergence rate of an implicit scheme will increase. Some implicit schemes show peak convergence rate at CFL numbers around 5 to 10. In general, it can be said that the use of a finer grid will hamper the convergence rate of computation to any flow. The trouble would become more acute if one

wants to solve a time dependent problem in a time-accurate manner in which a CFL number of order one (based on the smallest grid) must be maintained through out the flow domain. It is generally agreed that LR is still too expensive to be used in three dimensional (even in some two dimensional) engineering calculations.

Apparently, the industry has determined that the inherent inaccuracy of the method using the coarse grid associated with the use of wall functions is more than compensated for by economy and this is the reason why the wall function method will continue to be popular. It is paradoxical and interesting, though, that Refs. 1, 2 and 3 had reported the superiority in accuracy of HR over LR in solving many flow problems.

The base CFD code used in this study is the NPARC code[4] . The author was charged with the task of implementing wall-function boundary conditions to the existing LR k-ϵ turbulence model[5] which employs Chien's LR model[6]. It was found during the implementation that the drag coefficient of the flow over a flat plate was an order of magnitude lower than the experimental value. Naturally, much time had been spent to find the error in the implementation. Finally, it was found that the AV was the culprit.

The objectives of this paper are to: 1) detail how the wall function is implemented (despite its wide usage, it is hard to find a good reference on the technique); 2) demonstrate the mechanism by which AV causes trouble to the computation; and 3) show the methodology by which the difficulty was overcome.

Wall Function

The details of the Reynolds-averaged Navier-Stokes equations and the k-ϵ turbulence model will not be shown here, for the sake of brevity. The form of these equations can be found in the literature (see, e.g. , Ref. 7).

The specific detail in implementing the wall functions will depend partly on the main algorithm used in the flow solver as well as on the algorithm of the k-ϵ solver. The methodology presented here is specific to the centrally-differenced finite-difference scheme which employs AV as the stabilizing mechanism. The general philosophy used herein, however, should be applicable to other schemes as well.

The approach employed here is to prescribe k and ϵ values at the first grid point away from the wall and to solve for the flow variables, q, directly without any prescription of the variables. The advantage of this approach is two-fold: 1) the flow solver can still satisfy its strong conservation law form; 2) there is no disruption of the main flow algorithm at the wall.

The latter advantage makes the k-ϵ code more modular. The log-law profile, and the level of viscous drag associated with it, is imposed indirectly through the k and ϵ values that are imposed and through the fictitious turbulent viscosity imposed at the wall point. This latter point will be elaborated in detail later.

There are four major assumptions inherent in the wall function method:

1) The velocity profile satisfies the universal log-law:

$$\frac{u}{u_\tau} = \frac{1}{\kappa} ln(Ey^+) \tag{1}$$

2) the variables k and ϵ satisfy the Prandtl-Kolmogorov relation:

$$\mu_t = \frac{C_\mu \rho k^2}{\epsilon} \tag{2}$$

3) Local equilibrium: this means that the production of turbulence kinetic energy is equal to its dissipation, which can be written as

$$P = \rho\epsilon \tag{3}$$

where,

$$P = \overline{\rho u_i' u_j'} \frac{\partial u_i}{\partial x_j} \tag{4}$$

4) The mean flow is parallel to the solid wall so that the only production term which survives is that which multiplies with $\overline{u_1' u_2'}$ which is the wall shear stress itself.

It should be noted that implicit in the log-law profile assumption is also the assumption that turbulent shear stress is constant within the region (equal to ρu_τ^2) and that the turbulence length scale is directly proportional to the the distance from the wall.

The boundary condition for ϵ can be obtained from the local equilibrium relation:

$$\rho\epsilon = \overline{\rho u_i' u_j'} \frac{\partial u_i}{\partial x_j} \tag{5}$$

With the parallel flow assumption and the assumed Boussinesq's representation of the turbulent shear stresses, $\overline{\rho u_i' u_j'}$ becomes the turbulent stress term (ρu_τ^2) and the derivative of the mean flow reduces simply to $\frac{\partial u}{\partial y}$, so that the equation can now be written as

$$\epsilon = u_\tau^2 \frac{\partial u}{\partial y} \tag{6}$$

And finally, the derivative above is obtained by differentiating the log-law profile. This yields the relation:

$$\epsilon = \frac{u_\tau^3}{\kappa y} \tag{7}$$

The boundary condition for k is obtained by first writing the viscous drag relation at the first grid point away from the wall (point no. 2):

$$\tau_2 = \tau_w = \mu_t \frac{\partial u}{\partial y} \tag{8}$$

After substituting ρu_τ^2 for τ_w, eq. (2) for μ_t, eq. (1) for u, and eq. (7) for ϵ, and rearranging the resulting relation, the equation for k is obtained as

$$k = \frac{u_\tau^2}{\sqrt{C_\mu}} \tag{9}$$

It should be noted that by imposing the above values of k and ϵ , the log law profile is only partially satisfied (through its derivative) but the local equilibrium condition is strictly enforced.

As mention earlier, the strategy here is to not impose the log-law profile directly. Even if one wanted to do so, it would not be a very easy task in a general curvilinear coordinate system. the difficulty is that the choices of In addition, one may also have trouble in disrupting the strong conservation law form of the flow equations which is crucial for a shock capturing scheme.

So the strategy adopted in this study was to solve for the flow variables in a regular manner as one would normally do for the case of, say, an algebraic turbulence model. But the drag at the wall is imposed to be the drag dictated by the log-law profile. The procedure can be summarized as follows:

1. At any iteration step, solve for the velocity u, v, w as in the standard viscous method, with no-slip velocities applied at the wall.

2. From the values of the three velocities u, v and w, find the tangential velocity to the (in general, curved) wall.

3. Assume that this tangential velocity satisfies the log-law expression exactly and back out the value of u_τ from the log-law relation. In this study, the Newton's iteration was used to find u_τ from the log-law relation.

4. This value of u_τ (which is wrong if the flow does not yet reach a convergence) is used to fix the value of μ_t at the wall such that the total shear stress at the wall (laminar plus turbulent) is equal to ρu_τ^2

5. Proceed to the next iteration step.

Step 4 needs to be elaborated further. After u_τ is obtained from the Newton's iteration, the following relation should be satisfied:

$$\rho u_\tau^2 = 0.5[(\mu_{\tau_1} + \mu_{l_1}) + (\mu_{\tau_2} + \mu_{l_2})]\frac{\partial u_T}{\partial y_N} \tag{10}$$

Where, subscripts 1 and 2 denote the grid points at and next to the wall. This is nothing but the centrally-differenced viscous stress relation using the turbulent viscosity at the half-point. Note that the velocity derivative is evaluated by using the tangential velocity u_T and the normal distance y_N

The above equation can be used to solve for μ_{τ_1} since it is the only unknown in the equation. The derivative in the above equation needs to be scrutinized further. A careful examination revealed that this should be the finite-difference derivative (not the analytical derivative) and that the formula of the derivative must be the same as that used in the flow solver. With that, the value of μ_{τ_1} is obtained as

$$\mu_{\tau_1} = \frac{2y_N \rho u_\tau^2}{u_T} - (\mu_{l_1} + \mu_{\tau_2} + \mu_{l_2}) \tag{11}$$

This value of μ_{τ_1} will give the right value of the viscous drag at the wall without the programmer having to interfere with the logic of the algorithm of the flow solver. It was

found that the value of μ_{T_1} can become negative. But this is permissible since the average value of the viscosity is positive and the drag is the right physical drag. For other types of algorithms, eq. (9) should be changed to be the same as the viscous stress formula of the flow solver.

Description of the Test Case

Since the code used in this investigation was designed primarily for compressible flow problems, it was determined that a Mach 0.2 flow over a flat plate should be appropriate for the first test case because the solution can still converge reasonably fast while compressibility effect on turbulence should be negligible. The model must be able to solve this most simple case first before it will have any chance of solving other complex flows.

The domain of computation was a flat plate 50 cm. long and 2.5 cm wide. The upper boundary was 2.5 cm. above the plate. The grid employed was 31x31x5, uniformly spaced in the streamwise (x) and spanwise (z) directions. The five computational planes in the z-direction were used merely to expedite the two-dimensional flow since the CFD code was a three-dimensional one. In the normal (y) direction, the first grid point was placed at 0.0417 cm. away from the wall. The third grid point was 0.02085 cm. away from the second grid point. Thereafter, the grid expanded at the rate of 10% per grid point as it moved away from the wall. A convenient rule of thumb for determining the location of the first grid point is to place the point at 10% of the momentum thickness which is defined as[8]

$$\frac{\delta_2}{x} = 0.036 Re_x^{-0.2} \tag{12}$$

This will almost gaurantee that the grid point falls within the log-law region.

Boundary Conditions

For the flow equations, characteristic boundary conditions were imposed at the inflow and outflow boundaries (BC number 0 in the NPARC code). The total pressure and total temperature at inflow were set at 104192.5 Pascals and 302.4 °K, respectively. These conditions were derived from the static conditions of the Mach 0.2 flow at 101325 Pascals static pressure and 300 °K static temperature. The static pressure at the outflow boundary was set at 101325 Pascals.

In the spanwise direction, symmetric boundary conditions (BC number 50) were imposed to simulate two-dimensional flow. The conditions at the wall were the standard no-slip, adiabatic conditions (BC number 60). An extrapolation condition (BC number 3) was used for the upper boundary in order to allow the developing boundary layer to push the fluid out in a natural manner.

For the k-ϵ equations, the equations are purely hyperbolic in nature and the inflow conditions was thus fixed. The value of k was fixed at 1% of the free stream kinetic energy. The value of μ_t was assumed to be uniform at five times the laminar viscosity value. These are adhoc assumptions which could be evaluated to see if they have any effects on the solution when the turbulence intensity and viscosity takes on other assumed values. With k and μ_t fixed, the value of ϵ at inflow can be obtained from the Prandtl-Kolmogorov relation:

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \tag{13}$$

At the outflow and upper boundaries, extrapolation conditions were used. At the solid wall, the wall function conditions as described in the previous section were used.

Results and Discussion

With all the above mentioned implementation strategies, the flat plate test case was run using the NPARC code. The drag coefficient obtained from the results is compared with the experimental data of Klebanoff[9] in figure 1. It can be seen that the computational results differ from the experimental data by an order of magnitude. The velocity profile at the trailing edge of the plate reflects this erratic result, as can be seen in figure 2, which is the comparison of the computational results with the experimental data of Weighardt[10]. Much time was spent in finding the source of the error that caused these inaccurate results until finally the source was identified to be the AV terms. It should be noted that the k-ϵ solver contains no AV terms since it is based on an upwind TVD scheme.[11,12]

A proof that the error was caused by AV can be seen by progressively reducing the value of the fourth order AV coefficient (ϵ_4) from the default value of 0.64 to 0.0001. The results of these numerical experiments are shown in figures 3 and 4. Note that in this particular flow, the second order AV is not active because the pressure variation is too low, resulting in the pressure switch of the second-order AV being turned off all the time. It can be seen from the figures that at $\epsilon_4 = 0.0001$, the drag coefficient and the velocity profile from the computation

compare very well with those of the experiment. It is surprising that the code still remained stable at this extremely low value of the AV coefficient. It is believed that this was so because this was a very simple flow over a flat plate. Experience in more complex flows had shown that by reducing ϵ_4 to only 0.1 (from 0.64), code failure can occur.

Obviously, the practice of varying the AV coefficients to get a good solution match with experimental data is not a good practice. It is highly desirable that a method be found to get a good solution while still using the default, theoretically-determined AV coefficients.

The AV terms used in NPARC code were based upon the model introduced by Jameson, et al[13]. The terms for the j computational coordinate can be written as

$$D = \nabla_j(\lambda_j(\epsilon_2\triangle_j - \epsilon_4\triangle_j\nabla_j\triangle_j))(Jq) \tag{14}$$

Terms for the i and k coordinates can be written by simply changing the index j to i and k, respectively. The coefficients ϵ_2 and ϵ_4 are defined as

$$\epsilon_2 = \kappa_2\delta_p \tag{15}$$

$$\epsilon_4 = Max(0, \kappa_4 - \epsilon_2) \tag{16}$$

where κ_2 and κ_4 are numerical constants, hithereto referred to as AV coefficients. Their values should not exceed 0.25 and 0.64, respectively (actually, 0.64 is the value used in NPARC which is further scaled down to about 1/16). δ_p is called pressure switch which turns the second order AV on only when there are strong pressure fluctuations. The term is defined as

$$\delta_p = \frac{\mid p_{j+1} - 2p_j + p_{j-1} \mid}{p_{j+1} + 2p_j + p_{j-1}} \tag{17}$$

By considering the AV formula, it can be seen that when j equals to two (i.e., the point next to the wall) the AV formula involves derivatives of the flow variables at the wall as well as those beyond the wall which are fictitious grid points (since they do not really exist). In particlur, $u_2 - u_1$ takes on an unrealistically large value. This happens because the second grid point is too far away from the wall, as required by the wall function. If the near wall grid is fine enough, as in the low Reynolds number turbulence model case, then there should be no problem and the AV model should need no modifications. This unrealistically large value

of $u_2 - u_1$ adds error to the computation through the AV terms. One strategy to get around this is to get the derivative value at and beyond the wall points by extrapolation from the interior points as follows:

$$u_2 - u_1 = u_3 - u_2 \tag{18}$$

also,

$$u_1 - u_{-1} = u_3 - u_2 \tag{19}$$

where the subscript -1 refers to the fictitious point beyond the wall. This means that the second order AV terms vanish at the wall and that the fourth order AV terms reduce to:

$$D_4 = \epsilon_4(u_4 - 2u_3 + u_2) \tag{20}$$

which can be seen as equivalent to the second order AV term centering at point number 3 (but with ϵ_4 as the coefficient). Note also that the subscript 4 on D and ϵ are coincidental and does not represent the grid point number four.

With the above modifications of the AV terms and the AV coefficient, ϵ_4, sets at the default value (0.64), NPARC now produces good drag coefficients and velocity profiles, as can be seen in figures 5 and 6. Unlike the results shown in figure 3, the C_f distribution here shows no streamwise oscillations, due to the larger AV coefficient being used. Cross-stream profiles of k , ϵ and μ_t at the trailing edge are plotted in figures 7 ,8 and 9, respectively. Also shown in these figures are their counterparts obtained by using the original AV model. It is seen in the figures that k and ϵ of the modified AV method behave as expected in that k remains more or less constant in the near wall region (due to local equilibrium) before dropping off further away from the wall, while ϵ drops off immediately as it moves away from the wall. The k and ϵ of the original AV model, on the other hand, exhibit non-physical, very high peaks near the wall. Contours of μ_t are shown in figure 10. There are as many contour levels in the original AV case (figure 10.a) as in the modified AV case (figure 10.b) but the contour values of the former is ten times larger than those of the latter. It can be seen that the original AV model gives a much thicker boundary layer than the modified AV model. For the sake of

demonstration, velocity vectors of the last five streamwise stations are shown in figure 11. It should be noted that there are only 15 grid points in the entire boundary layer. This should be compared to 50 or more grid points needed if the low Reynolds number model were used.

It is the opinion of the author that this kind of error due to AV is not unique to central difference schemes. Upwind schemes would probably exhibit a similar error insofar as upwind schemes can be interpreted as central difference schemes plus some extra AV terms. If these extra AV terms involve derivatives of conservative variables at the wall then it is to be expected that the upwind schemes would suffer a similar type of error as explained above for the central difference scheme.

Another minor effect of AV on solution quality was found by inspecting the velocity profile near the edge of the boundary layer. There appeared to be a velocity overshoot (over that of the free stream) of the order of 1%. This overshoot extended over a distance of about one boundary layer thickness before gradually dropping off to the free stream value. This overshoot would be very hard to notice in a casual observation. In fact, the overshoot would be barely noticeable in a normal-scale plotting. Figure 12 shows the velocity overshoot near the edge of boundary layer as compared to the case run with the same modified AV model but using ϵ_4=0.0001. It can be seen clearly that the overshoot was caused by the high (regular) value of the AV coefficient. Unfortunately, the low values of the AV coefficients cannot always be used in flows more complex than the flow over a flat plate.

Upon further investigation, it was found that this small velocity overshoot can cause up to 10% error in the Re_θ (hence C_f) calculation. Re_θ is the Reynolds number based on the momentum thickness which is defined by the following formula:

$$\delta_2 = (1/\rho_\infty u_\infty^2) \int_0^\infty \rho u(u_\infty - u)dy \tag{21}$$

Note that the upper integration limit for y is ∞ which means the integration must go through the velocity overshoot region; this is the source of error mentioned above. At any streamwise station, the overshoot will render δ_2 (hence Re_θ) to be smaller than it should be. An adhoc remedy was used in this investigation wherein the upper limit of the integration was taken at $y = \delta_{99}$ (method 1) instead of integrating to ∞ (method 2). The results of using methods 1 and 2 in computing C_f as a function of Re_θ are plotted in figure 13 where it can be seen that method 2 yields a lower value of C_f than method 1 which is already slightly lower than the experimental value.

Figure 14 shows Re_θ using the two methods of integration versus Re_x. As can be seen, method 2 gives about a 10% smaller value of Re_θ. It would be interesting to see if upwind schemes would give this same overshoot when using a coarse grid as was used in this study.

Conclusion

The study has revealed that artificial viscosity (AV) terms can induce a very large error in the drag coefficient value and the shape of velocity profiles of a turbulent flow when turbulence is simulated by the k-ϵ equations in conjunction with the wall-function boundary conditions. The study has demonstrated that this kind of error can be circumvented simply by extrapolating the velocity profiles from the interior points toward the boundary points and using these extrapolated values in the AV formula while still retaining the no-slip boundary conditions in the flow solver as usual.

The AV terms were also found to cause a small (about 1%) velocity overshoot near the boundary layer edge. This small error, however, can cause a fairly large error (about 10%) in momentum thickness. This type of error can be reduced by integrating the boundary layer only up to δ_{99}, and not to ∞.

References

[1]J.R. Viegas and M.W. Rubesin, "Wall-Function Boundary Conditions in the Solution of the Navier-Stokes Equations for Complex Compressible Flows," AIAA Paper 83-1694, July 1983.

[2]J.R. Viegas, M.W. Rubesin, and C.C. Horstman, "On the Use of Wall Functions as Boundary Conditions for Two-Dimensional Separated Compressible Flows," AIAA Paper 85-0180, Jan. 1985.

[3] C.J. Steffen, Jr., "A Critical Comparison of Several Low Reynolds Number k-ϵ Turbulence Models for Flow Over a Backward-Facing Step," AIAA Paper 93-1927, June 1993.

[4]Cooper, G., and Sirbaugh, J., "The PARC Distinction: A Practical Flow Simulator," AIAA Paper 90-2002, July 1990.

[5]Chien, K.Y., "Predictions of Channel and Boundary-Layer Flows with a Low-Reynolds-Number Turbulence Model," AIAA Journal, Vol. 20, No. 1, 1980, pp. 33-38.

[6]Georgiadis, N.J., Chitsomboon, T., and Zhu, J., "Modification of the Two-Equation Turbulence Model in NPARC to a Chien Low-Reynolds Number k-ϵ Formulation," NASA TM-106710, Sept., 1994.

[7] Launder, B.E., and Spalding, D.B., "The Numerical Computation of Turbulent Flows," Computer Methods in Applied Mechanics and Engineering 3(1974), pp. 269-289.

[8]Kays, W.M., and Crawford, M.E., "Convective Heat and Mass Transfer," McGraw-Hill, 1980, p. 175.

[9]Klebanoff, P.S., "Characteristics of Turbulence in a Boundary Layer with Zero Pressure Gradient," NACA Rept. 1247, 1955.

[10]Weighardt, K., and Tillmann, W., "On the Turbulent Friction Layer for Rising Pressure," NACA TM 1314, 1951.

[11] Nichols, R.H., "A Two-Equation Model for Compressible Flows," AIAA Paper 90-0494, Jan. 1990.

[12] Gorski, J.J, Chakravarthy, S.R., and Goldberg, U.C., "High Accuracy TVD Schemes for the k-ϵ Equations of Turbulence," AIAA Paper 85-1655, July 1985.

[13]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods using Runge-Kutta Time-Stepping Scheme," AIAA Paper 81-1259, June 1981.

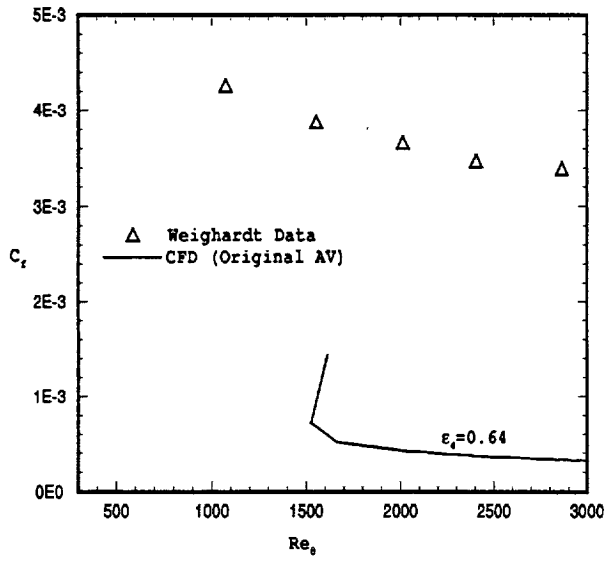


Fig. 1. Drag coefficient using the original AV model

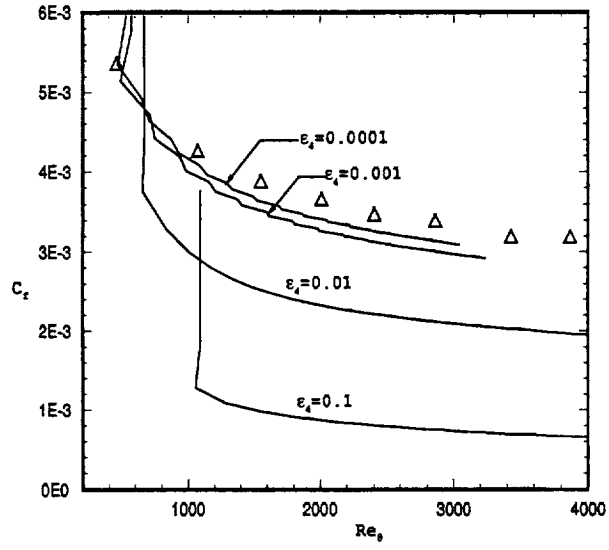


Fig. 3. Drag coefficients using original AV with reduced values of AV coefficient

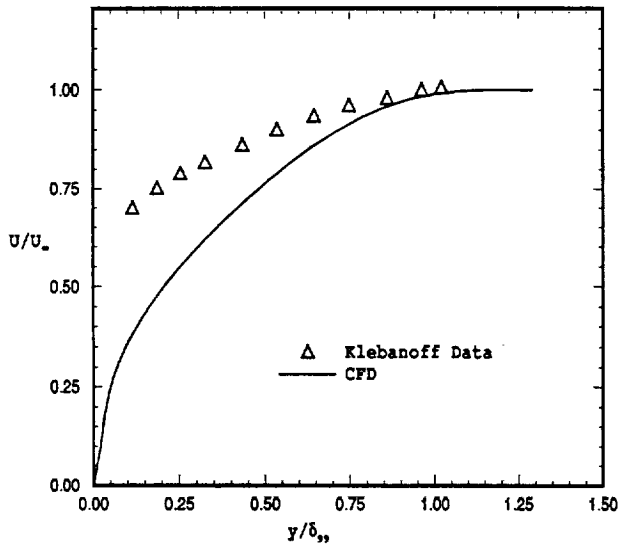


Fig. 2. Velocity profile at traling edge using original AV model

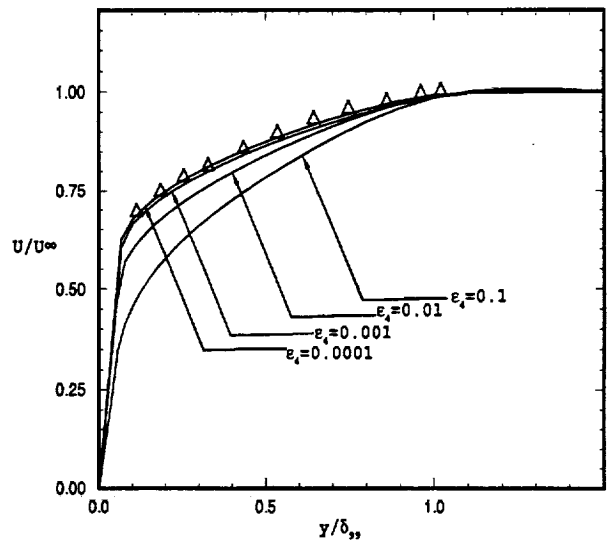


Fig. 4. Velocity profiles at trailing edge uisng original AV with reduced values of AV coefficient

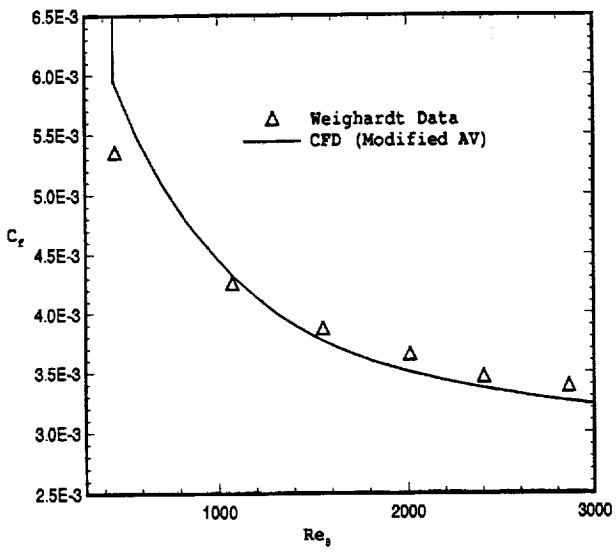


Fig. 5. Drag coefficient using modified AV with default value of AV coefficient ($\epsilon_4 = 0.64$)

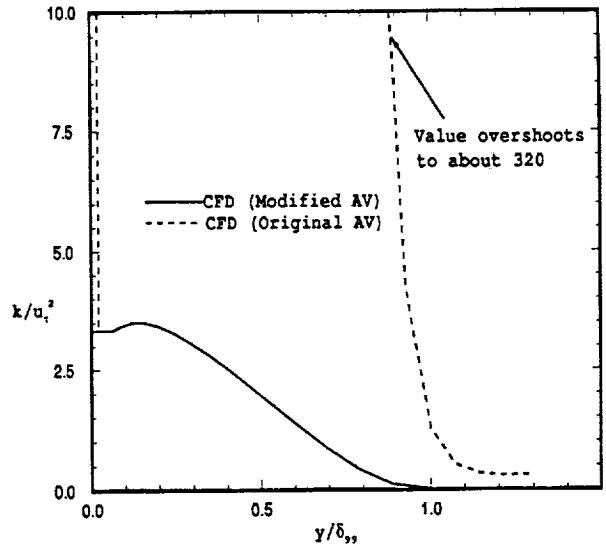


Fig. 7. k distributions of the original and modifed AV cases

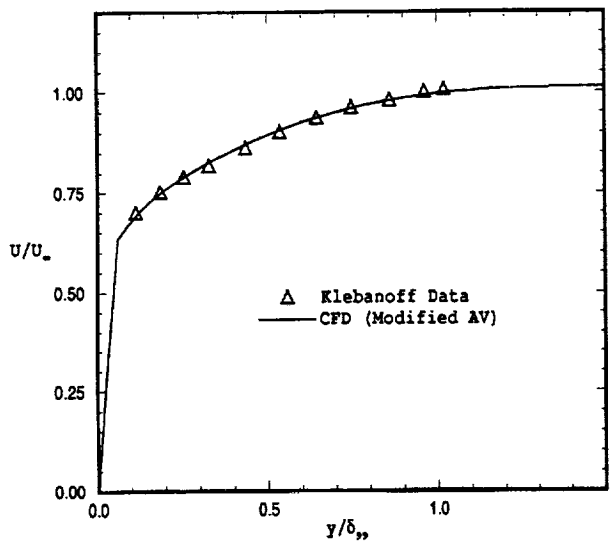


Fig. 6. Velocity profile at trailing edge using modified AV with default value of AV coefficient

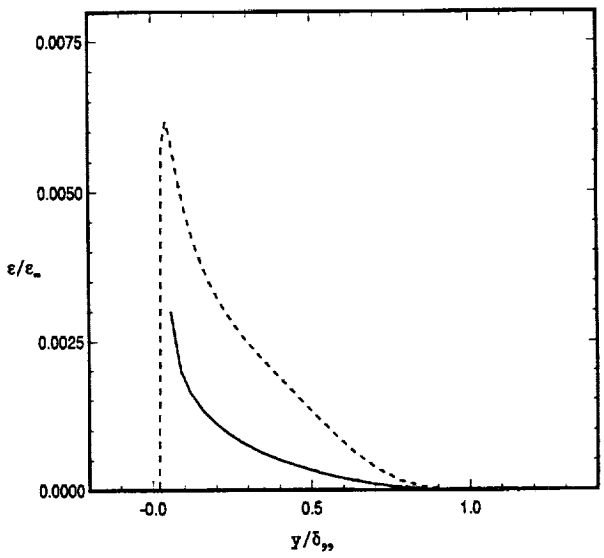


Fig. 8. ϵ distributions of the original and modifed AV cases

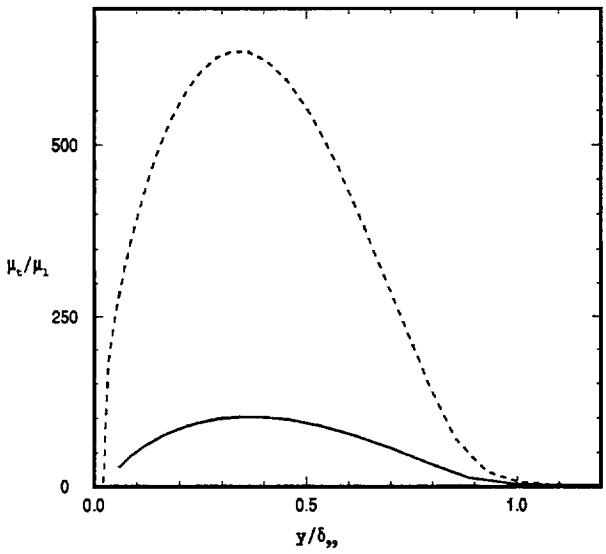


Fig. 9. μ_t distributions of the original and modifed AV cases

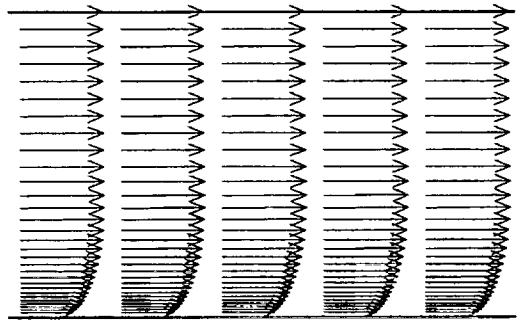


Fig. 11. Velocity vectors of the modifed AV case

Fig. 10. Contours of μ_t for original AV (a) and modified AV (b)

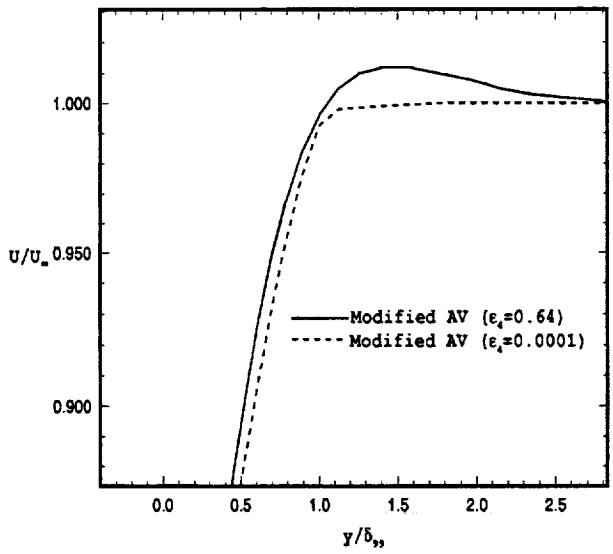


Fig. 12. Velocity profiles near the boundary layer edge

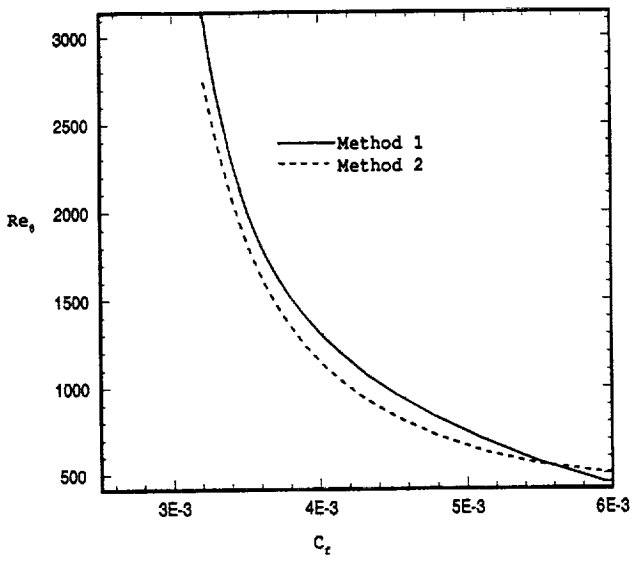


Fig. 13. Friction coefficient vs. Re_θ using the two methods of integration

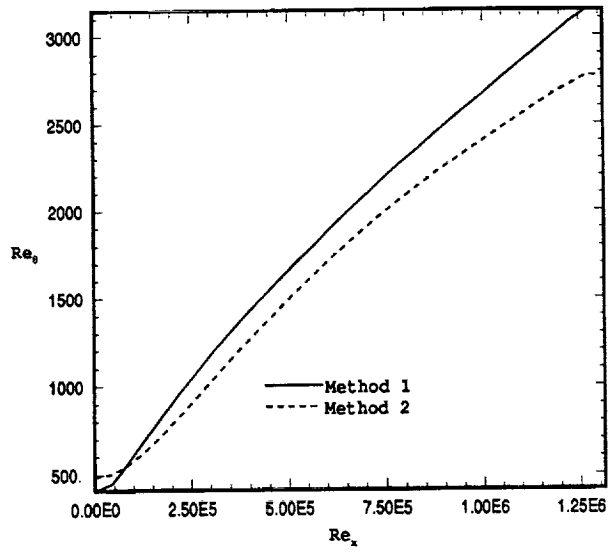


Fig. 14. Re_θ vs. Re_x using the two methods of integration

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 1994	Technical Memorandum

4. TITLE AND SUBTITLE

Effects of Artificial Viscosity on the Accuracy of High-Reynolds-Number k-ε Turbulence Model

5. FUNDING NUMBERS

WU-505-90-5K

6. AUTHOR(S)

Tawit Chitsomboon

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-9234

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-106781
ICOMP-94-28

11. SUPPLEMENTARY NOTES

Tawit Chitsomboon, Institute for Computional Mechanics in Propulsion, NASA Lewis Research Center (work funded under NASA Cooperative Agreement NCC3-233). ICOMP Program Director, Louis A. Povinelli, organization code 2600, (216) 433-3106.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 34

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Wall functions, as used in the typical high Reynolds number k-ε turbulence model, can be implemented in various ways. A least disruptive method (to the flow solver) is to directly solve for the flow variables at the grid point next to the wall while prescribing the values of k and ε. For the centrally-differenced finite-difference scheme employing artificial viscocity (AV) as a stabilizing mechanism, this methodolgy proved to be totally useless. This is because the AV gives rise to a large error at the wall due to too steep a velocity gradient resulting from the use of a coarse grid as required by the wall function methodology. This error can be eliminated simply by extrapolating velocities at the wall, instead of using the physical values of the no-slip velocities (i.e. the zero value). The applicability of the technique used in this paper is demonstrated by solving a flow over a flat plate and comparing the results with those of experiments. It was also observed that AV gives rise to a velocity overshoot (about 1 %) near the edge of the boundary layer. This small velocity error, however, can yield as much as 10% error in the momentum thickness. A method which integrates the boundary layer up to only the edge of the boundary (instead of to ∞) was proposed and demonstrated to give better results than the standard method.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Turbulence modeling; Wall functions; Artificial viscosity	20
	16. PRICE CODE A03

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

National Aeronautics and
Space Administration

Lewis Research Center

Cleveland, OH 44135-3191