AIAA 95-1835
13th AIAA Applied Aerodynamics Conference
June 19–22, 1995 / San Diego, CA

EFFICIENCY AND ACCURACY OF TIME-ACCURATE TURBULENT NAVIER-STOKES COMPUTATIONS

Christopher L. Rumsey,[1,*] Mark D. Sanetrik,[2,*] Robert T. Biedron,[2] N. Duane Melson,[1,†] and Edward B. Parlette[3]

[1]*NASA Langley Research Center, Mail Stop 128, Hampton, VA 23681-0001,* [2]*AS&M, Inc., Hampton, VA, and* [3]*Vigyan, Inc., Hampton, VA*

Abstract

The accuracy and efficiency of two types of subiterations in both explicit and implicit Navier-Stokes codes are explored for unsteady laminar circular-cylinder flow and unsteady turbulent flow over an 18-percent-thick circular-arc (biconvex) airfoil. Grid and time-step studies are used to assess the numerical accuracy of the methods. Nonsubiterative time-stepping schemes and schemes with physical time subiterations are subject to time-step limitations in practice that are removed by pseudo time subiterations. Computations for the circular-arc airfoil indicate that a one-equation turbulence model predicts the unsteady separated flow better than an algebraic turbulence model; also, the hysteresis with Mach number of the self-excited unsteadiness due to shock and boundary-layer separation is well predicted.

Introduction

Although still prohibitively expensive, the routine inclusion of unsteady flow computations using the Navier-Stokes equations into the aircraft aerodynamic design cycle becomes more tractable as the speed and memory capacity of computers continues to increase. Many computer codes currently exist that can perform time-accurate (in addition to steady-state) computations; however, because most applications are still made in the steady-state realm, relatively little effort has been spent in the systematic analysis of the accuracy and efficiency of these codes for unsteady flows. In addition, little is currently known about the applicability of the new generation of eddy-viscosity field-equation turbulence models to unsteady flows.

Earlier works have addressed some of the considerations necessary in the application of Navier-Stokes codes to unsteady flow computations. For example, Pulliam[1] and Newsome[2] showed the importance of subiterations in enhancing the time accuracy of conventional implicit schemes. Jameson[3] and Melson et al.[4] developed a subiterative technique within the context of a multigrid methodology to allow practical time steps with the use of an explicit code.

The current work combines some of the concepts presented in these earlier papers into a comprehensive study that explores subiteration strategies as applied to explicit and implicit computer codes. The explicit central-difference code TLNS3D[5] and the implicit upwind code CFL3D[6] are both used in conjunction with subiterations to advance the equations of motion in a time-accurate manner. Two approaches to performing subiterations are explored and compared. Computations are performed for laminar flow over a circular cylinder and for turbulent flow over an 18-percent-thick circular-arc airfoil for which self-excited shock-induced flow oscillations occur. (See McDevitt[7] and Mabey.[8])

Previous computational studies with the Navier-Stokes equations were performed for the circular-arc airfoil by Levy,[9] Steger,[10] and Edwards and Thomas.[11] Levy used the MacCormack explicit scheme; Steger and Edwards and Thomas used implicit approximate-factorization schemes. All employed algebraic turbulence models. All three studies reproduced the self-excited oscillations, although the reduced frequencies were about 20 percent lower in these three studies than in the experiment of McDevitt.[7] Le Balleur and Girodroux-Lavigne[12] also predicted self-excited oscillations using a small-disturbance potential method with an interacted two-equation integral viscous model, but the computed reduced frequency was about 30 percent lower than in the experiment. More recently, Edwards[13] used a transonic small-disturbance potential code, coupled with a lag-entrainment integral boundary-layer method, to accurately predict reduced frequencies. He also accurately predicted the experimentally observed hysteresis of the oscillations with Mach number.

Levy blamed the poor prediction of the Navier-Stokes solvers on the failure of algebraic turbulence models to predict the flow features characteristic of shock-wave–boundary-layer interactions with relatively large regions of separated flow. Today, more advanced turbulence models are readily available for use with time-accurate Navier-Stokes solvers. In this paper, results with the algebraic Baldwin-Lomax (B-L) turbulence model[14] and the one-equation Spalart-Allmaras (S-A) turbulence model[15] are compared for this application.

Governing Equations

The governing equations are the thin-layer Navier-Stokes equations, written in generalized coordinates as

[*] Senior Member AIAA.
[†] Associate Fellow AIAA.

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial\left(\hat{F} - \hat{F}_v\right)}{\partial \xi} + \frac{\partial\left(\hat{G} - \hat{G}_v\right)}{\partial \eta} + \frac{\partial\left(\hat{H} - \hat{H}_v\right)}{\partial \zeta} = 0 \qquad (1)$$

where

$$\hat{Q} = \frac{Q}{J} \qquad (2)$$

and Q is the vector of conserved variables given as

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \qquad (3)$$

The inviscid flux terms are

$$\hat{F} = \frac{1}{J}\begin{bmatrix} \rho U \\ \rho U u + \xi_x p \\ \rho U v + \xi_y p \\ \rho U w + \xi_z p \\ (e+p)\,U - \xi_t p \end{bmatrix} \qquad (4)$$

$$\hat{G} = \frac{1}{J}\begin{bmatrix} \rho V \\ \rho V u + \eta_x p \\ \rho V v + \eta_y p \\ \rho V w + \eta_z p \\ (e+p)\,V - \eta_t p \end{bmatrix} \qquad (5)$$

and

$$\hat{H} = \frac{1}{J}\begin{bmatrix} \rho W \\ \rho W u + \zeta_x p \\ \rho W v + \zeta_y p \\ \rho W w + \zeta_z p \\ (e+p)\,W - \zeta_t p \end{bmatrix} \qquad (6)$$

The variable J represents the Jacobian of the transformation:

$$J = \frac{\partial(\xi, \eta, \zeta)}{\partial(x, y, z)} \qquad (7)$$

The contravariant velocities are given by

$$U = \xi_x u + \xi_y v + \xi_z w + \xi_t \qquad (8)$$

$$V = \eta_x u + \eta_y v + \eta_z w + \eta_t \qquad (9)$$

and

$$W = \zeta_x u + \zeta_y v + \zeta_z w + \zeta_t \qquad (10)$$

The viscous terms are

$$\hat{F}_v = \frac{1}{J}\begin{bmatrix} 0 \\ \xi_x \tau_{xx} + \xi_y \tau_{xy} + \xi_z \tau_{xz} \\ \xi_x \tau_{yx} + \xi_y \tau_{yy} + \xi_z \tau_{yz} \\ \xi_x \tau_{zx} + \xi_y \tau_{zy} + \xi_z \tau_{zz} \\ \xi_x b_x + \xi_y b_y + \xi_z b_z \end{bmatrix} \qquad (11)$$

$$\hat{G}_v = \frac{1}{J}\begin{bmatrix} 0 \\ \eta_x \tau_{xx} + \eta_y \tau_{xy} + \eta_z \tau_{xz} \\ \eta_x \tau_{yx} + \eta_y \tau_{yy} + \eta_z \tau_{yz} \\ \eta_x \tau_{zx} + \eta_y \tau_{zy} + \eta_z \tau_{zz} \\ \eta_x b_x + \eta_y b_y + \eta_z b_z \end{bmatrix} \qquad (12)$$

and

$$\hat{H}_v = \frac{1}{J}\begin{bmatrix} 0 \\ \zeta_x \tau_{xx} + \zeta_y \tau_{xy} + \zeta_z \tau_{xz} \\ \zeta_x \tau_{yx} + \zeta_y \tau_{yy} + \zeta_z \tau_{yz} \\ \zeta_x \tau_{zx} + \zeta_y \tau_{zy} + \zeta_z \tau_{zz} \\ \zeta_x b_x + \zeta_y b_y + \zeta_z b_z \end{bmatrix} \qquad (13)$$

where

$$b_{x_i} = u_j \tau_{x_i x_j} - \dot{q}_{x_i} \qquad (14)$$

The pressure is obtained by the equation of state for a perfect gas:

$$p = (\gamma - 1)\left[e - \rho\frac{(u^2 + v^2 + w^2)}{2}\right] \qquad (15)$$

Description of the Codes

The TLNS3D and CFL3D codes both solve the three-dimensional time-dependent thin-layer Navier-Stokes equations with a finite-volume formulation. Both can employ grid sequencing, multigrid, and local time stepping when accelerating convergence to steady state.

The TLNS3D code, described in detail in reference 5, is a central-difference code. Second-order central differences are used for all spatial derivatives, and a blend of second-difference and fourth-difference artificial dissipation terms is used to maintain numerical stability. These artificial dissipation terms can be added in either scalar or matrix form. The solution is advanced explicitly in time with either a four- or five-stage Runge-Kutta time-marching algorithm. For the results presented in this paper, both matrix dissipation and a five-stage Runge-Kutta scheme are used.

The CFL3D code, described in detail in reference 6, is an upwind code. For all results in this paper, upwind-biased spatial differencing is used for the inviscid terms, and flux limiting is used to obtain smooth solutions in the vicinity of shock waves. All viscous terms are centrally differenced. The equations are solved implicitly in time with the use of a three-factor approximate factorization (AF). Either the flux-difference-splitting (FDS) method of Roe[16] or the flux-vector-splitting (FVS) method of Van Leer[17] can be employed to obtain fluxes at the cell faces. For all results in this paper, the FDS approach is used.

The turbulence models (B-L[14] and S-A[15]) are described in detail in the given references. The one-equation turbulence model is decoupled from the Navier-Stokes equations; the resulting partial-differential equation is solved implicitly with three-factor AF.

Time-Advancement and Subiteration Algorithms

For a nondeforming mesh, equation (1) can be written as

$$\frac{1}{J}\frac{\partial Q}{\partial t} = R(Q) \tag{16}$$

where

$$R = -\left[\frac{\partial\left(\hat{F}-\hat{F}_v\right)}{\partial\xi} + \frac{\partial\left(\hat{G}-\hat{G}_v\right)}{\partial\eta} + \frac{\partial\left(\hat{H}-\hat{H}_v\right)}{\partial\zeta}\right] \tag{17}$$

The time term can be discretized with backward differencing:

$$\frac{(1+\phi)\,(Q^{n+1}-Q^n) - \phi\,(Q^n - Q^{n-1})}{J\Delta t} = R(Q^{n+1}) \tag{18}$$

where the superscripts indicate time level. When $\phi = 0$ the method is first-order temporally accurate; when $\phi = 1/2$ the method is second-order accurate. This equation is implicit because the right-hand side is a function of the unknown flow variables at time level $n+1$. Different methods for solving this equation with the TLNS3D and CFL3D codes are explored below.

TLNS3D

The TLNS3D code is normally explicitly advanced in time (with a Runge-Kutta scheme); the right-hand side of equation (18) is taken at time level n. However, this time-advancement scheme is generally impractical for time-accurate runs with a fixed Δt due to severe stability-restricted time-step limitations. Therefore, as shown in reference 4, for time-accurate problems an additional level of iteration is introduced that is iteratively implicit. This additional iteration level allows the code to take a physical time step of nearly any size. Basically, multigrid and pseudo time steps are used to iterate to the next *physical* time step.

A pseudo time term is added to equation (18) as

$$\frac{1}{J}\frac{\partial Q}{\partial\tau} + \frac{(1+\phi)\,(Q^{n+1}-Q^n) - \phi\,(Q^n - Q^{n-1})}{J\Delta t} = R(Q^{n+1}) \tag{19}$$

where τ represents pseudo time. As shown, the addition of the pseudo time term alters the Navier-Stokes equations, but the $\partial Q/\partial\tau$ term is treated in such a way that it vanishes as Q approaches its correct value at time level $n+1$.

A five-stage Runge-Kutta integration scheme ($K = 5$) is used to iterate equation (19) in pseudo time as

$$V^0 = Q^m \tag{20}$$

$$V^k = V^0 + \alpha_k J\Delta\tau\left[R(V^k) - \frac{(1+\phi)\,(V^k - Q^n) - \phi\,(Q^n - Q^{n-1})}{J\Delta t}\right], k = 1, \ldots, K \tag{21}$$

$$Q^{m+1} = V^K \tag{22}$$

where m is the subiteration counter and $Q^{m+1} \to Q^{n+1}$ as $m \to \infty$. Note that the physical time step Δt is a constant (the same for all computational cells); however, $\Delta\tau$ is based on a constant CFL number (currently taken as 7) and varies as a function of cell size and local flow quantities.

Further algorithm modifications must be made in equation (21) for efficiency and stability. The residual vector R is split into a convective and diffusive part $R = C - D$, where the diffusive part includes both physical and artificial diffusion terms. The convective part is evaluated at the last Runge-Kutta step $(k-1)$ of this subiteration; the diffusive part is evaluated at some combination of one or more previous Runge-Kutta time steps (signified by $\overline{k-l}$) of this subiteration:

$$C(V^k) \cong C(V^{k-1}) \tag{23}$$

$$D(V^k) \cong D\left(V^{\overline{k-l}}\right) \tag{24}$$

Unlike the explicit treatment in Jameson,[3] $(1+\phi)\,V^k$ is treated implicitly in the Runge-Kutta integration. In order that the residual (i.e., the term in the brackets in equation (21)) goes to zero as the subiterations converge, $(1+\phi)\,\alpha_k\Delta\tau/(\Delta t)\,\Delta Q^{k-1}$ is added and subtracted from the right-hand side. The final iterative-implicit Runge-Kutta algorithm in TLNS3D is

$$V^0 = Q^m \tag{25}$$

$$\left(1 + (1+\phi)\,\alpha_k\frac{\Delta\tau}{\Delta t}\right)V^k = V^0 + (1+\phi)\,\alpha_k\frac{\Delta\tau}{\Delta t}Q^{k-1} + \alpha_k J\Delta\tau\left[C(Q^{k-1}) - D\left(Q^{\overline{k-l}}\right) - \frac{(1+\phi)\,(Q^{k-1} - Q^n) - \phi\,(Q^n - Q^{n-1})}{J\Delta t}\right], k = 1, \ldots, K \tag{26}$$

$$Q^{m+1} = V^K \tag{27}$$

CFL3D

The CFL3D code is advanced in time with an implicit approximate-factorization method. The implicit derivatives are written as spatially first-order accurate, which results in block-tridiagonal inversions for each sweep. However, for solutions that utilize FDS the block-tridiagonal inversions are further simplified with a diagonal algorithm (with a spectral radius scaling of the viscous terms).

Because of the method with which the left-hand side is treated for computational efficiency in steady-state runs (approximate factorization, first-order accuracy), second-order temporal accuracy is forfeited for unsteady computations. One method for recovering the desired accuracy is through the use of subiterations. Two different subiteration strategies have been implemented in CFL3D. The first method, which is termed "pseudo time subiteration (τ-TS)," is similar to the strategy employed in TLNS3D. The other method, termed "physical time subiteration (t-TS)," follows Pulliam.[1]

For the τ-TS method in CFL3D, as in TLNS3D, a pseudo time term is added to the time-accurate Navier-Stokes equations. See equation (19). This equation is then discretized and iterated in m, where m is the subiteration counter.

$$\frac{(1+\phi')\,(Q^{m+1}-Q^{m})-\phi'\,(Q^{m}-Q^{m-1})}{J\Delta\tau}+ \qquad (28)$$
$$\frac{(1+\phi)\,(Q^{m+1}-Q^{n})-\phi\,(Q^{n}-Q^{n-1})}{J\Delta t}=R\,(Q^{m+1})$$

In equation (28), ϕ and ϕ' govern the order of accuracy of the physical and pseudo time terms, respectively. In practice the pseudo time term is treated as first order (i.e., $\phi' = 0$), but the general form is shown here for completeness. As $m \to \infty$, the pseudo time term vanishes and $Q^{m+1} \to Q^{n+1}$. If R is linearized with

$$R\,(Q^{m+1}) \cong R\,(Q^{m}) + \frac{\partial R}{\partial Q}\Delta Q^{m} \qquad (29)$$

and the quantity $-(1+\phi)\,Q^{m}/\,(J\Delta t)$ is added to both sides of equation (28), then equation (28) becomes

$$\left[\left(\frac{1+\phi'}{J\Delta\tau}+\frac{1+\phi}{J\Delta t}\right)I+\delta_{\xi}A+\delta_{\eta}B+\delta_{\zeta}C\right]\Delta Q^{m} = \qquad (30)$$
$$\frac{\phi'\Delta Q^{m-1}}{J\Delta\tau}+\frac{\phi\Delta Q^{n-1}}{J\Delta t}-\frac{(1+\phi)\,(Q^{m}-Q^{n})}{J\Delta t}+R\,(Q^{m})$$

where $\Delta Q^{m} = Q^{m+1}-Q^{m}$, $A = \partial(\hat{F}-\hat{F}_{v})/\partial Q$, $B = \partial(\hat{G}-\hat{G}_{v})/\partial Q$, and $C = \partial(\hat{H}-\hat{H}_{v})/\partial Q$. Equation (30) is approximately factored and written in primitive variable form; it is solved as a series of sweeps in each coordinate direction as

$$\left[\left(\frac{(1+\phi')\,M}{J\Delta\tau}+\frac{(1+\phi)\,M}{J\Delta t}\right)+\delta_{\xi}\tilde{A}\right]\Delta q' = \frac{\phi' M\Delta q^{m-1}}{J\Delta\tau} \qquad (31)$$
$$+\frac{\phi M\Delta q^{n-1}}{J\Delta t}-\frac{(1+\phi)\,M\,(q^{m}-q^{n})}{J\Delta t}+R\,(q^{m})$$

$$\left[\left(\frac{(1+\phi')\,M}{J\Delta\tau}+\frac{(1+\phi)\,M}{J\Delta t}\right)+\delta_{\eta}\tilde{B}\right]\Delta q'' = \qquad (32)$$
$$\left(\frac{(1+\phi')\,M}{J\Delta\tau}+\frac{(1+\phi)\,M}{J\Delta t}\right)\Delta q'$$

$$\left[\left(\frac{(1+\phi')\,M}{J\Delta\tau}+\frac{(1+\phi)\,M}{J\Delta t}\right)+\delta_{\zeta}\tilde{C}\right]\Delta q^{m} = \qquad (33)$$
$$\left(\frac{(1+\phi')\,M}{J\Delta\tau}+\frac{(1+\phi)\,M}{J\Delta t}\right)\Delta q''$$

$$q^{m+1} = q^{m}+\Delta q^{m} \qquad (34)$$

where the primitive variables are $q = [\rho, u, v, w, p\,]^{T}$, $M = \partial Q/\partial q$, $\tilde{A} = \partial(\hat{F}-\hat{F}_{v})/\partial q$, $B = \partial(\hat{G}-\hat{G}_{v})/\partial q$, and $\tilde{C} = \partial(\hat{H}-\hat{H}_{v})/\partial q$. The quantity $\Delta\tau$ is based on a constant CFL number of 10. Multigrid is used to drive Δq^{m} to zero in a reasonable number of subiterations.

In the t-TS method, equation (18) is merely iterated in m, where m is the subiteration counter:

$$\frac{(1+\phi)\,(Q^{m+1}-Q^{n})-\phi\,(Q^{n}-Q^{n-1})}{J\Delta t} = R\,(Q^{m+1}) \qquad (35)$$

The quantity $-(1+\phi)\,Q^{m}/\,(J\Delta t)$ is added to both sides, the residual is linearized, and the equation is approximately factored and written in primitive variable form as

$$\left[\frac{(1+\phi)\,M}{J\Delta t}+\delta_{\xi}\tilde{A}\right]\Delta q' = \frac{\phi M\Delta q^{n-1}}{J\Delta t}- \qquad (36)$$
$$\frac{(1+\phi)\,M\,(q^{m}-q^{n})}{J\Delta t}+R\,(q^{m})$$

$$\left[\frac{(1+\phi)\,M}{J\Delta t}+\delta_{\eta}\tilde{B}\right]\Delta q'' = \frac{(1+\phi)\,M}{J\Delta t}\Delta q' \qquad (37)$$

$$\left[\frac{(1+\phi)\,M}{J\Delta t}+\delta_{\zeta}\tilde{C}\right]\Delta q^{m} = \frac{(1+\phi)\,M}{J\Delta t}\Delta q'' \qquad (38)$$

$$q^{m+1} = q^{m}+\Delta q^{m} \qquad (39)$$

As $m \to \infty$, $q^{m+1} \to q^{n+1}$. When only one series of sweeps is performed, $q^{m} = q^{n}$ and the standard time-accurate CFL3D scheme is recovered (i.e., no subiterations). Unlike the τ-TS method, this subiteration procedure (equations (36)–(39)) utilizes only one time step: the physical time step Δt (= constant). Multigrid is currently not used for the t-TS subiteration procedure.

Results

For all results in this paper, second-order temporal accuracy is employed ($\phi = 1/2$). In addition, although the computations above are formulated for three-dimensional (3-D) flows, only two-dimensional (2-D) computations are performed for this validation study.

Laminar Flow Over Circular Cylinder

Laminar flow over a circular cylinder is computed at $M = 0.3$ and $Re = 1200$ (based on diameter) on a 101×49 grid, with minimum spacing at the body of 0.002 diameters and a far-field extent of 20 diameters. The relative accuracy and efficiency of the subiterative schemes are evaluated by establishing periodic vortex shedding and then comparing the Strouhal number $St = nd/u_{\infty}$ (where n is the shedding frequency and d is the diameter) with the limit of St as $\Delta t \to 0$ on this grid.

Because the primary focus of this paper is on the temporal accuracy and efficiency of the subiterative time-stepping schemes described earlier, grid density and grid extent studies are not performed for the circular cylinder flow. The authors recognize that the 101 × 49 grid is relatively coarse and that the computed unsteady flow is not grid converged for this case. Also, the use of the thin-layer equations on a bluff-body flow is questionable and will deviate to some degree from a full Navier-Stokes solution, as indicated in a previous study by Rumsey et al.[18] However, the effect of grid density and extent is investigated below for the circular-arc airfoil case, for which the use of the thin-layer assumption is more appropriate.

The number of subiterations required to converge the subiterative schemes differs for the t-TS and τ-TS methods. An example is shown in figure 1 in which the CFL3D code is used with Δt = 0.2 (nondimensionalized by d/a_∞). The quantity $St/St_{\Delta t \to 0}$ is plotted; as $\Delta t \to 0$, St approaches a value of 0.235 for CFL3D and 0.247 for TLNS3D on this grid. For this problem, three subiterations are sufficient to converge the t-TS method; eight three-level multigrid cycles are necessary for the τ-TS method. Although not shown, the results are similar for the TLNS3D code. Also, if no multigrid is employed, the τ-TS method takes as many as twenty subiterations to converge to the same level.

When significantly smaller or larger time steps are taken, the number of subiterations required for τ-TS can vary. (Smaller time steps require fewer subiterations; larger time steps require more subiterations.) However, this variation has not been explored in detail in this study.

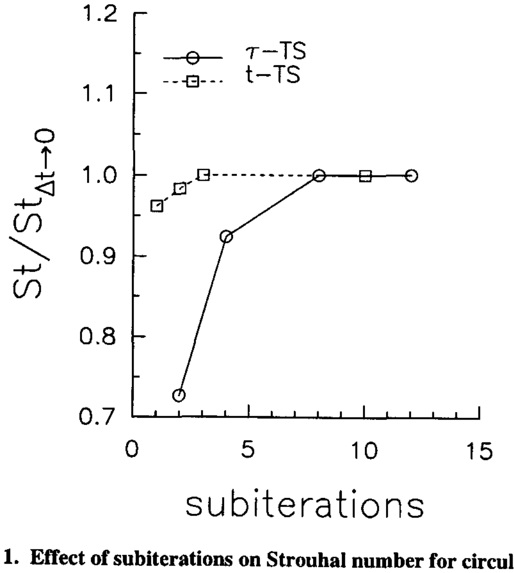

Fig. 1. Effect of subiterations on Strouhal number for circular cylinder; CFL3D, Δt = 0.2 .

Figure 2 shows the effect of time step Δt on the Strouhal number for CFL3D with no subiterations (unmodified scheme), CFL3D with τ-TS and eight three-level multigrid subiterations, CFL3D with t-TS and three subiterations, and the TLNS3D code with τ-TS and eight three-level multigrid subiterations. When no subiterations are employed, use of the explicit TLNS3D code is not practical for this unsteady problem because of the severe stability restriction ($\Delta t < O(0.001)$). Although CFL3D is an implicit code, it also has a time-step limitation in practice when τ-TS is not employed; for this particular case, the limitation is approximately $\Delta t \le 0.4$ (which yields a maximum CFL number in the flow field of about 400). This limit is indicated in the figure by a vertical bar.

When no subiterations are employed in CFL3D, the accuracy of the code quickly diminishes (roughly linearly with Δt, which is indicative of first-order temporal accuracy) as the time step is increased. In contrast, when subiterations of the type τ-TS are used in either code, not only are larger time steps possible, but the accuracy is also significantly enhanced. Even at a time step as high as Δt = 1.0 (roughly 15 steps per period), $St/St_{\Delta t \to 0} > 0.9$ for both codes. The CFL3D code with the t-TS method is still restricted by the upper time-step limit of Δt = 0.4. However, the accuracy is dramatically improved in comparison with the noniterative results and is comparable with the τ-TS results.

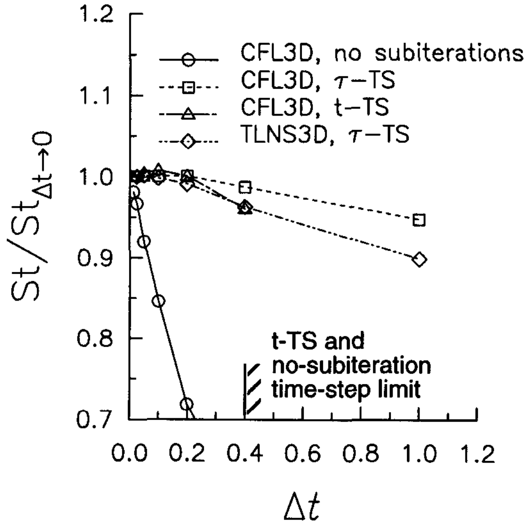

Fig. 2. Effect of subiteration type and time step on Strouhal number for circular cylinder.

The second-order temporal accuracy of the codes is difficult to verify with St because the number of steps per period is obtained by counting the number of time steps from maximum c_l to maximum c_l and, hence, can only be accurate to ±1 unless the true location of maximum c_l is interpolated. For example, at a time step of Δt = 0.2, 71 steps per period indicates that St lies somewhere in the range 0.231–0.238. Other quantities, such as peak lift and drag coefficients, are more amenable to this type of analysis because they are less sensitive to number of steps per period. Table 1 shows $|c_{l,max}|$ and $c_{d,min}$ as a function of time step for the t-TS method in the CFL3D code. The quantity $c_{d,max}$ is not shown because its variation from its extrapolated value of 1.7714 is smaller than the fourth decimal place at the smallest time steps. Note that the error is approximately quadrat-

Table 1. Verification of Second-Order Temporal Accuracy for t-TS in CFL3D, with 101×49 Cylinder Grid

Δt	steps/period	$\lvert c_{l,max}\rvert$	% error	$c_{d,min}$	% error
0.2	71	approx 1.60	+11.0	approx 1.41	+3.7
0.1	141	1.451	+0.694	1.3649	+0.338
0.05	283	1.435	-0.416	1.3573	-0.221
0.025	568	1.4378	-0.222	1.3588	-0.110
0.0125	1137	1.4400	-0.069	1.3599	-0.029
0.00625	2274	1.4408	-0.014	1.3602	-0.007
0 (extrapolated)	∞	1.4410	-	1.3603	-

ic (i.e., it approximately quadruples when the time step is doubled) only for the smallest three time steps taken. Although not shown, results with the τ-TS method in CFL3D are similar. In other words, for this problem on the 101×49 grid, at least 500 time steps per shedding cycle are necessary to ensure that second-order temporal accuracy is achieved. If fewer steps per cycle are taken, then temporal errors no longer necessarily vary with the square of the time step.

The true test of a scheme's efficiency is the central processing unit (CPU) time required to obtain a result for a given level of accuracy. In figure 3, $St/St_{\Delta t \to 0}$ is plotted versus CPU time per shedding cycle for CFL3D with the standard (unmodified) scheme with no subiterations, the τ-TS method with eight multigrid subiterations, and the t-TS method with three subiterations. Each curve is a plot of the results obtained by varying Δt. Clearly, the t-TS method is the most efficient; $St/St_{\Delta t \to 0}$ is nearly equal to 1.0 in as little as 4 CPU sec/period (on NASA Ames Research Center's Aerodynamic Consolidated Supercomputer Facility (ACSF) CRAY C-90 computer). This result corresponds to a time step of 0.2 and roughly 11 μ sec per grid point per time step. In contrast, the τ-TS method requires about 30 sec/period, or roughly 85 μ sec per grid point per time step, to obtain the same level of accuracy (this result also corresponds to a time step of 0.2). Not only is each subiteration more expensive in the τ-TS method because of the use of multigrid, but more subiterations are necessary to converge the iterative scheme far enough. Surprisingly, in terms of efficiency, the τ-TS method is not significantly more efficient than CFL3D with no multigrid and no subiterations; this latter method can run in roughly 4 μ sec per grid point per time step. The efficiency of the TLNS3D code is not shown in this figure because its CPU times (roughly 360 μ sec per grid point per time step for τ-TS with eight multigrid subiterations) are significantly larger than those of CFL3D. Unlike the CFL3D code, the TLNS3D code has not been optimized for 2-D problems. For 3-D problems, the two methods are expected to be comparable in efficiency.

A sketch that summarizes the effects of subiterations and time step is shown in figure 4. The dashed line indicates the converged solution at a given time step. It is achieved when a sufficient number of subiterations are performed. However, note that the converged solution at a given time step still deviates from the "true" solution at an infinitely small time step. This error is generally greater at larger time steps, but the region of true second-order temporal accuracy exists only over a portion of the time-step axis near $\Delta t = 0$.

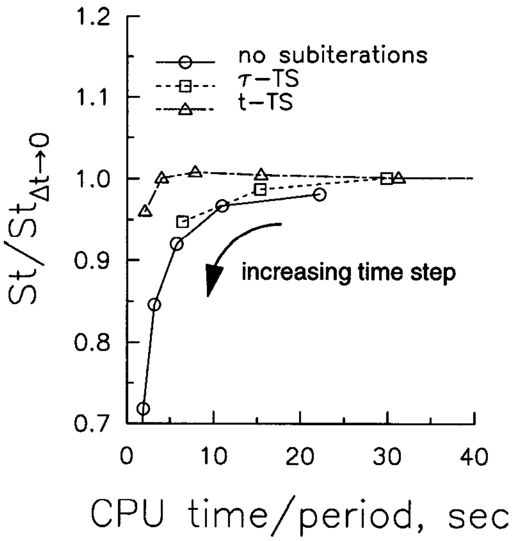

Fig. 3. Efficiency of subiteration types for circular cylinder with CFL3D.

Turbulent Flow Over 18-Percent-Thick Circular-Arc Airfoil

Turbulent flow over an 18-percent-thick circular-arc airfoil is computed at Mach numbers between $M = 0.72 - 0.76$ and $Re = 11$ million based on chord length; the computational results are compared with the experimental results of McDevitt.[7] At these conditions, the circular-arc airfoil can experience self-induced periodic shock oscillations.

Assessments are made of the effect of time step, subiteration type, grid size, grid extent, Mach number, turbulence model, computer code, and tunnel walls on the computed unsteady solution. A complete list of parameters and results for the various circular-arc airfoil runs are given in Table A.1 in the appendix. The CPU times in seconds per period are given for the ACSF CRAY C-90 computer (designated by a "C"), and for NASA Langley Research Center's CRAY Y-MP computer (designated by a "Y").

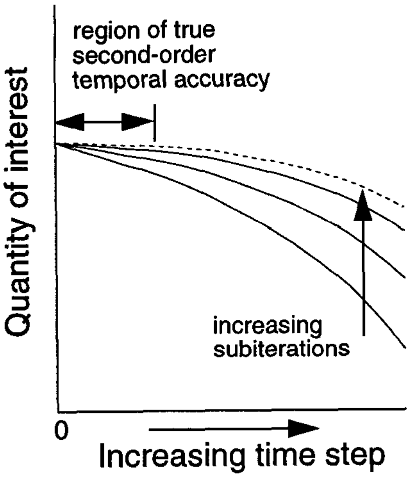

Fig. 4. Sketch of time step and subiteration effect.

Because the S-A turbulence model is a field-equation model, a partial-differential equation is solved for a turbulence quantity related to the eddy viscosity. This equation is solved iteratively in both codes (decoupled from the Navier-Stokes equations) with a three-factor implicit AF scheme. Because the model contains advective terms, the turbulence equation must be solved time accurately. Currently, first-order temporal differencing is employed, and the advective terms are discretized with first-order upwind differencing. If the S-A turbulence model is instead advanced in time with local time stepping (i.e., the turbulence quantity is treated in a quasi-steady manner with respect to the instantaneous unsteady flow field at each time step), then the frequencies are predicted to be about 10 percent lower for this problem.

Most of the computations in this study are performed on a 185×65 C mesh with average minimum normal spacing at the wall of approximately 4.3×10^{-6} chords, a maximum grid extent of approximately 18 chords, and 129 points on the airfoil. This minimum spacing corresponds to $y+$ values between about 1.8 (near the leading edge) and 0.5 (near the trailing edge) for a typical unsteady computation at $M = 0.76$. Part of the lower half of this grid is shown in figure 5, along with pressure contours over the upper half of the grid at one point in the unsteady cycle of run #7 of Table A.1 at $M = 0.76$. This grid is made by taking every other point from a 369×129 grid. Grid density studies are performed on this finer grid, and grid extent studies are performed by adding points at the outer boundary, which extends the far-field extent to about 45 chords. A second grid, of a different family, is a 257×81 C mesh with average minimum normal spacing at the wall of approximately 3.6×10^{-6} chords, a maximum grid extent of approximately 18 chords, and 177 points on the airfoil. It is used to assess the effect of Mach number.

Boundary conditions at the body surface are no slip, with an adiabatic wall temperature and $\partial p/\partial n = 0$. At the curved outer boundary, far-field Riemann invariant boundary conditions are employed. At the downstream flat portion of the C mesh, all flow quantities are extrapolated. The free-stream temperature is taken to be 486 R.

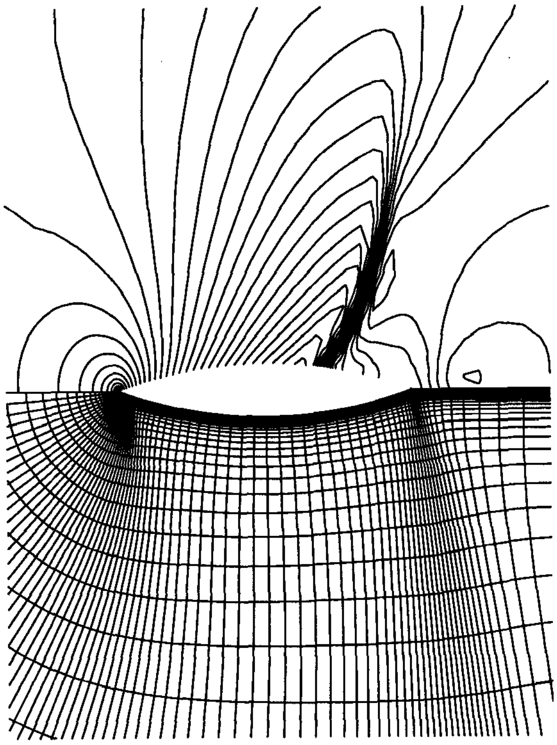

Fig. 5. Typical unsteady pressure contours and one-half of 185×65 C mesh for circular-arc airfoil.

The effect of tunnel walls is assessed on a two-block ($2B$) grid (one above and one below the airfoil) with 241×97 grid points in each block. Part of the lower block of this grid is shown in figure 6, along with pressure contours in the upper block at one point in the unsteady cycle of run #22. Minimum normal spacing at the wall is 3.9×10^{-6} chords, and the grid extends 2 chords upstream of the leading edge and 3 chords downstream of the trailing edge. A total of 258 points are used on the airfoil surfaces. The upper and lower tunnel walls are treated as inviscid surfaces, and the exit pressure is prescribed (with all other quantities extrapolated from the interior of the grid) in order to obtain the desired average Mach number at a specified location in the tunnel. The following boundary conditions are used at the inflow boundary: the velocity normal to the streamwise direction is set to zero, pressure is extrapolated from the interior of the grid, and reference entropy and enthalpy conditions are imposed.

Effect of Subiteration Type and Time Step

Figure 7 shows the effect of number of subiterations on the reduced frequency k $(=2\pi n(c/2)/u_\infty)$ for this problem on the 185×65 grid, where n is the unsteady frequency and c is the chord length. For the τ-TS method, at least fifteen three-level multigrid subiterations are sufficient for both CFL3D and TLNS3D at a time step of $\Delta t = 0.05$ (about 170 steps per period). The time step is nondimensionalized by c/a_∞. Note that this number of subiterations is roughly twice that required for the

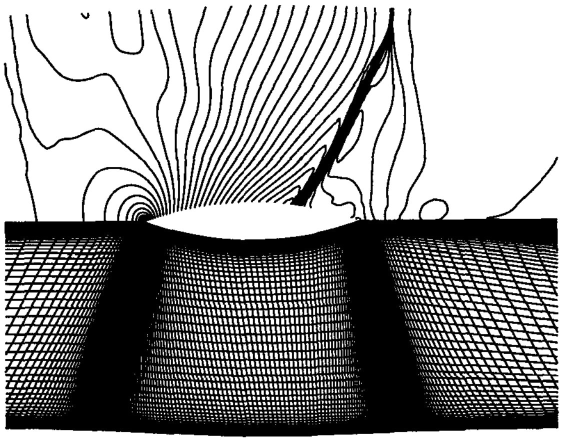

Fig. 6. Typical unsteady pressure contours and one-half of $241 \times 97 \times 2B$ tunnel grid for circular-arc airfoil in tunnel.

laminar circular cylinder computations. For the t-TS method, even twenty subiterations is not sufficient to fully converge the solution at a time step of $\Delta t = 0.005$ (over 1700 steps per period). This slow convergence with subiterations for t-TS is peculiar to the use of the field-equation turbulence model. Note from run #20 in Table A.1 that when the B-L turbulence model is employed, only three t-TS subiterations are required to predict reduced frequency in good agreement with the τ-TS result (run #12). This inconsistent behavior of the t-TS method warrants further study.

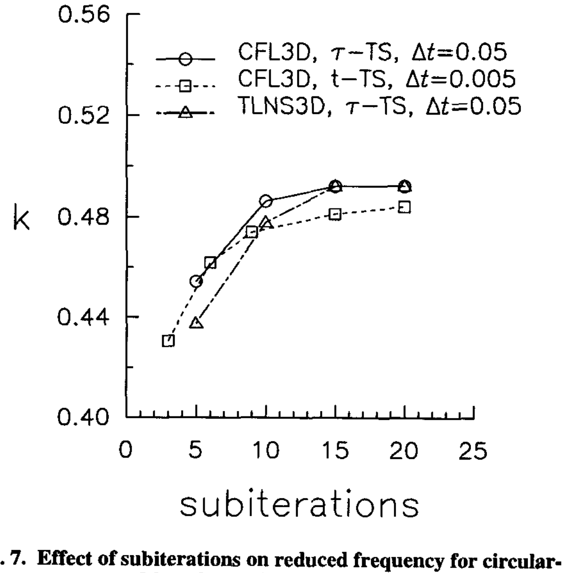

Fig. 7. Effect of subiterations on reduced frequency for circular-arc airfoil; $M = 0.76$, 185×65 grid, S-A model.

Figure 8 shows the reduced frequency as a function of time step Δt for the two codes on the 185×65 grid. Results with the t-TS method are not shown in the figure because of the excessive number of subiterations that are required to converge it when the field-equation turbulence model is employed. As was the case for the circular cylinder flow, the time step for the CFL3D code with no subiterations (or with t-TS) is limited in practice; on this grid, a time step no larger than $\Delta t = 0.005$ (which yields a maximum CFL number in the flow field of about 2000) can be taken. This time-step limit is indicated in figure 8 by a vertical bar. The τ-TS method removes the time-step limitation. All results approach the same value of approximately $k = 0.49-0.50$ for an infinitely small time step on this grid. At a time step of $\Delta t = 0.10$, which corresponds to about 90 steps per period, the frequency is predicted to within approximately 7 percent of its value with an infinitely small time step on this grid. If $\Delta t = 0.05$ (approximately 170 steps per period), then k is predicted to within about 1 percent. With no subiterations, if $\Delta t = 0.000625$ (14067 steps per period), then k is predicted to within approximately 5 percent. Recall that the number of steps per period can be obtained only to ± 1. For example, at a time step of 0.1, 87 steps per period indicates that k lies somewhere in the range of 0.470–0.481 (it is given as 0.475). The smaller the time step, the more accurately k can be determined. All τ-TS results in this figure were obtained with fifteen three-level multigrid subiterations.

At first glance, the reduced frequency values predicted by τ-TS for the two codes in figure 8 might appear to vary approximately quadratically with Δt (as should be the case for a temporally second-order-accurate scheme). However, a significant uncertainty associated with k at the larger time steps still exists, and the earlier circular cylinder results suggest that the computations are probably not truly temporally second-order accurate until on the order of 500 time steps per cycle are taken. Table 2, which shows the effect of time step on the maximum and minimum force coefficients and on k for the 185×65 grid, confirms the fact that second-order temporal accuracy has not been achieved at the larger two time steps. Further refinement in time is necessary to determine the time step below which the error varies quadratically with Δt. Note from the table that, although k is underpredicted at the largest time step by as much as 6 percent from its value at the smallest time step, peak lift is in error by as much as 19 percent and peak drag is in error by as much as 23 percent on this grid.

The reduced frequency is plotted as a function of CPU time per period in figure 9 for results on the 185×65 grid. Each curve is generated by varying the time step for a given method. Clearly, the reduced frequency is predicted to within a given level of accuracy in far less CPU time when τ-TS subiterations are employed as opposed to no subiterations. This turbulent-flow result is different than the earlier laminar-flow result (of figure 3) that τ-TS subiterations are comparable in efficiency to no subiterations. Results using t-TS subiterations again are not plotted because of the excessive number of subiterations required when the S-A model is employed. The efficiency of the TLNS3D code is not shown in this figure for the same reason mentioned in the circular cylinder study.

The approximate speeds of the various methods in μ s per grid point per time step for this turbulent flow computation on the CRAY C-90 are as follows: 6 for CFL3D with no subiterations, 170 for CFL3D with fifteen three-level multigrid τ-TS subiterations, and 470 for

Table 2. Effect of Time Step on k and Lift and Drag Peak Values; $M = 0.76$, τ-TS, 185×65 Grid, S-A Model

Code	Δt	steps/period	k	$\lvert c_{l,max} \rvert$	$c_{d,min}$	$c_{d,max}$
CFL3D	0.10	87	0.475	0.369	0.066	0.082
	0.05	168	0.492	0.337	0.061	0.075
	0.025	334	0.495	0.327	0.059	0.073
	0.0125	670	0.494	0.323	0.056	0.072
TLNS3D	0.10	89	0.465	0.394	0.076	0.091
	0.05	168	0.492	0.363	0.068	0.084
	0.025	332	0.498	0.342	0.064	0.079
	0.0125	665	0.497	0.330	0.062	0.077

Table 3. Effect of Grid on k and Lift and Drag Peak Values; $M = 0.76$, τ-TS, $\Delta t = 0.05$, S-A Model

Code	Grid	steps/period	k	$\lvert c_{l,max} \rvert$	$c_{d,min}$	$c_{d,max}$
CFL3D	185×65	168	0.492	0.337	0.061	0.075
	369×129	176	0.470	0.361	0.068	0.079
TLNS3D	185×65	168	0.492	0.363	0.068	0.084
	369×129	180	0.459	0.378	0.073	0.085

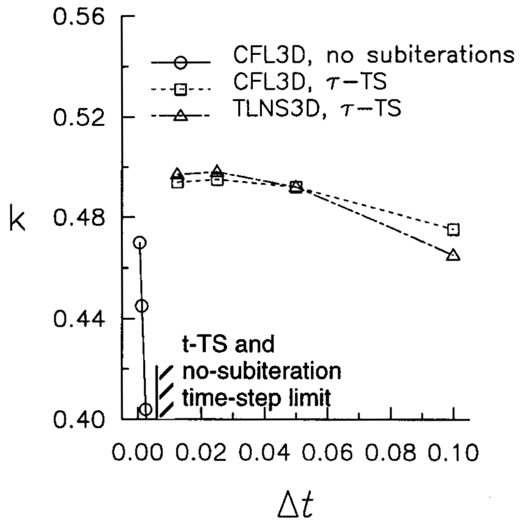

Fig. 8. Effect of time step on reduced frequency for circular-arc airfoil; $M = 0.76$, 185×65 grid, S-A model.

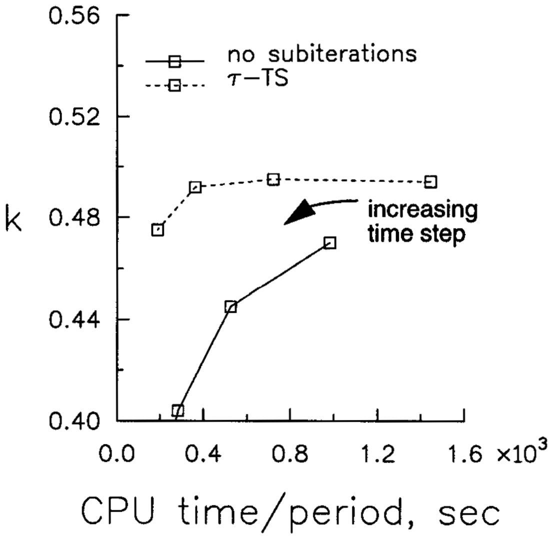

Fig. 9. Efficiency of subiterations for circular-arc airfoil; $M = 0.76$, CFL3D, 185×65 grid, S-A model.

TLNS3D with fifteen three-level multigrid τ-TS subiterations. The speed for CFL3D with t-TS subiterations averages about 5.5 μs per subiteration. Hence, if three subiterations are required, then the t-TS method costs about 17 μs per grid point per time step. If twenty subiterations are required, then it costs about 110 μs per grid point per time step.

Effect of Grid

The effect of doubling the grid density is shown in Table 3 with the τ-TS method, a time step of $\Delta t = 0.05$, and fifteen three-level multigrid subiterations for both codes. The reduced frequency is overpredicted by the coarse grid by as much as 7 percent from its value on the finest 369×129 grid, peak lift is in error by as much as 7 percent, and peak drag is in error by as much as 10 percent. Note that at this time step, even results on the finest 369×129 grid are not completely code converged. The reduced frequencies predicted by CFL3D and TLNS3D differ by about 2 percent, and the peak lift and drag coefficients differ by 5–8 percent.

The effect of grid extent is also investigated. As indicated in runs #14 and #35 in Table A.1, an increase in the grid extent from 18 chords to 45 chords has no effect for the CFL3D code (compared with run #7), and only a small effect (a decrease in k of less than 1 percent) for the TLNS3D code (compared with run #29).

Comparison with Experiment

The effect of Mach number is explored on the 257×81 grid with CFL3D. In particular, the ability of the Navier-Stokes code to predict the hysteresis effect noted in the experiment of McDevitt[7] and the computations of Edwards[13] is tested. In the experiment, the flow remained steady as the Mach number was increased in increments of 0.001 up to $M = 0.76$, at which point the oscillatory flow was established. After the oscillatory flow was established, unsteady flow persisted as the Mach number was lowered to approximately 0.733, at which point the flow became steady again. In the interactive boundary-layer/small-disturbance potential code results of Edwards, the computed hysteresis range extended from $M = 0.755-0.735$.

The hysteresis effect is tested by running CFL3D in steady-state mode; $M = 0.70$ at the start and is and increased in increments of 0.01. At every Mach number up to $M = 0.75$, the lift coefficient oscillates about zero with a magnitude of order 10^{-13} (near machine zero) and the residual (right-hand side) of the equation for density decreases at least 5 orders of magnitude (i.e., the solutions converges). Note that to obtain machine-zero levels of lift, the grid must be symmetric about the x-axis. The computations must also be initiated at a Mach number *below* the hysteresis region and then restarted at successively higher Mach numbers from fully converged lower Mach-number solutions.

As shown in figure 10, as the Mach number is increased, the separation point on the airfoil for this grid moves forward from $x/c = 0.94$ when $M = 0.70$ to $x/c = 0.84$ when $M = 0.75$. A shock first appears at $x/c = 0.61$ when $M \approx 0.72$. It moves slightly downstream as the Mach number is increased. When $M = 0.76$, the separation point jumps forward to the base of the shock at $x/c = 0.69$, and the lift starts to oscillate about zero with exponentially increasing magnitude. Eventually, the solution settles into a periodic cycle. This intersection of the trailing-edge separation point with the shock is probably the physical mechanism that triggers the onset of the unsteady flow with increasing Mach number.

Starting from the unsteady solution at $M = 0.76$, the code is then run time accurately at decreasing Mach numbers. The flow remains unsteady at every Mach number down to $M = 0.73$. Below this, the oscillations damp out and the solution becomes steady. These results are shown in figure 11. The reduced frequencies and the hysteresis region are predicted with reasonable accuracy. Although not shown, the prediction of the hysteresis region was sensitive to the grid density and the discretization scheme.

It is not known whether use of the current time-advancement schemes are appropriate for computing the hysteresis region. If one assumes that hysteresis is associated with an unstable fixed point of the Navier-Stokes equations, then a direct solver rather than a time-marching algorithm may be necessary to insure that the lower (steady) branch is indeed a fixed-point solution. This issue warrants further study.

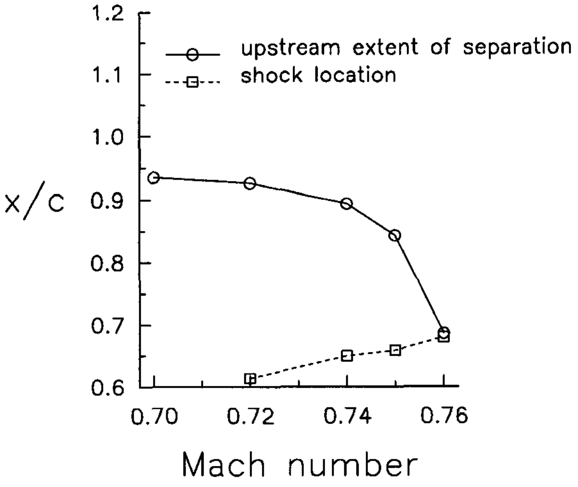

Fig. 10. Trailing-edge separation and shock locations; CFL3D, 257×81 grid, S-A model.

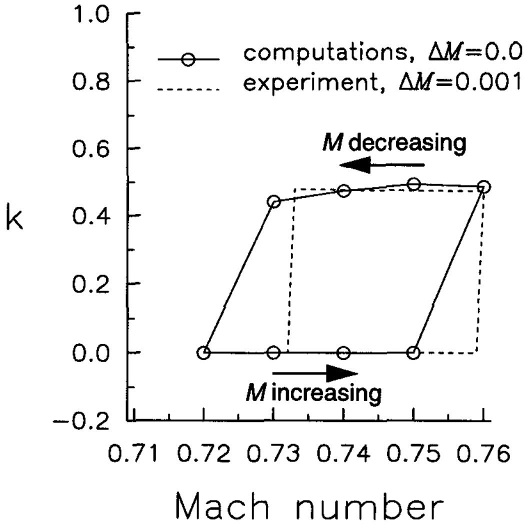

Fig. 11. Computed hysteresis region for circular-arc airfoil; CFL3D, 257×81 grid, S-A model.

The lift histories from two different turbulence models are shown as a function of t^* ($= t/(c/u_\infty)$) in figure 12 for $M = 0.76$. The S-A model with both CFL3D and TLNS3D yields a reduced frequency that agrees well with the experimental value of approximately 0.472. The B-L model predicts a lower frequency and lower amplitudes, consistent with previous results that utilized algebraic turbulence models. (See Edwards and Thomas.[11]) Note, however, that previous Navier-Stokes results[11] with the B-L model predicted steady flow at a similar Mach number of $M = 0.754$, whereas computations at $M = 0.78$ exhibited unsteady flow. These results suggest that the present B-L computations at $M = 0.76$ may be close to the lower end of the Mach-number range in which the unsteady flow occurs for this turbulence model.

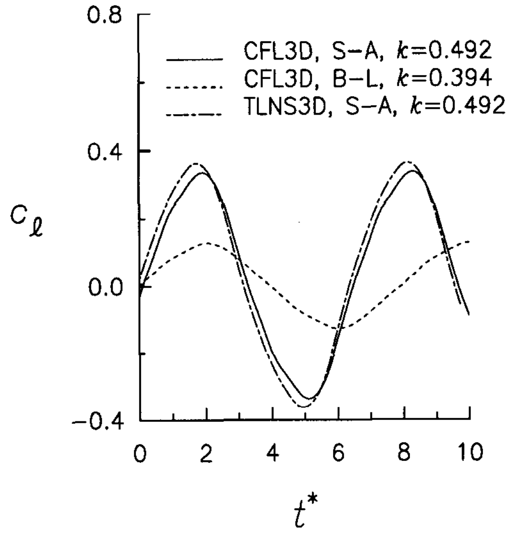

Fig. 12. Lift-coefficient variation for circular-arc airfoil; $M = 0.76$, 185×65 grid.

Figure 13 shows the shock position on the airfoil surface, nondimensionalized by chord, as a function of $\bar{t}$, where $\bar{t}$ is the fractional cyclic time ($\bar{t} = 0 \rightarrow 1$ represents one period of motion). Following McDevitt,[7] the time $\bar{t} = 0$ is taken to be the time when the shock is at midchord. (If the shock does not reach midchord in the computations, then $\bar{t} = 0$ is the time when the shock is closest to midchord.) The S-A model follows the experimental data of McDevitt closely, although it does not predict motion of the shock forward of midchord. (Note that the discrete jumps in the computed results in the figure are a result of assigning a shock location to the nearest streamwise grid point.) Results with both CFL3D and TLNS3D are nearly identical. The B-L model predicts a significantly smaller movement of the shock than the S-A model.

The effect of the grid density on the predicted shock position is shown in figure 14. The coarser grid is sufficient to capture the general character of the shock motion for this flow. Upper surface pressure coefficients taken at five different times during the unsteady cycles for CFL3D with both the S-A and B-L models and for TLNS3D with the S-A model are shown in figures 15, 16, and 17, respectively.

The CFL3D code is also run at $M = 0.74$ on the 185×65 grid. The predicted k is 0.477, similar to the earlier results on the 257×81 grid. Variation of shock position is shown in comparison with results at $M = 0.76$ in figure 18. At the lower Mach number, the shock travels slightly further forward of midchord, in better agreement with the experimental results. However, the lower Mach number computation also changes the character of the shock motion between $\bar{t} = 0.4$–0.8.

The effect of the tunnel walls is investigated by using the tunnel grid shown in figure 6. In the experiment, the local side-wall Mach number (1.34 chords upstream of the leading edge) was used to estimate Mach number in the center of the channel, and an empirical relationship was then employed to estimate M_∞. For $M_\infty = 0.76$, the Mach number in the tunnel was $M_{x/c=-1.34} \approx 0.744$. (See McDevitt et al.[19]) In the computation (run #22), the back pressure boundary condition was adjusted until an average Mach number of $M \approx 0.743$ was obtained 1.33 chords upstream of the leading edge.

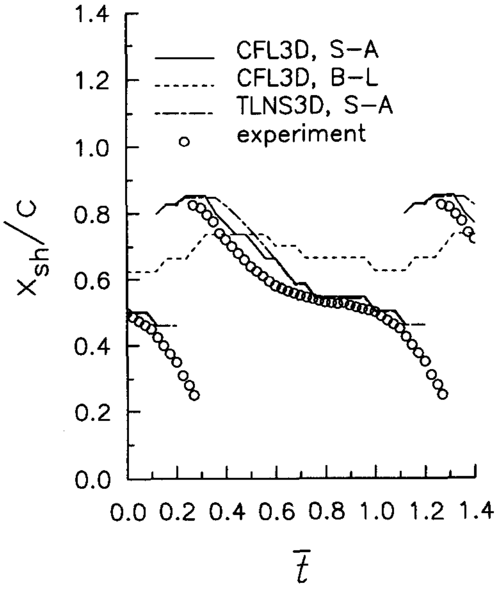

Fig. 13. Variation of shock location on surface of circular-arc airfoil; $M = 0.76$, 185×65 grid.

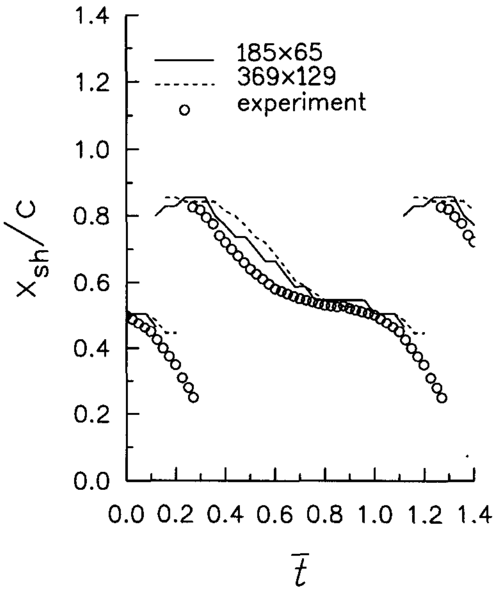

Fig. 14. Effect of grid on variation of shock location on surface of circular-arc airfoil; $M = 0.76$, CFL3D, S-A model.

Tunnel-grid results are shown in figures 19, 20, and 21. In figure 19, the $M = 0.76$ results on the 185×65 C mesh (free-air grid) are included for comparison. The tunnel-grid shock motion is in excellent agreement with the experimental results, including the movement of the shock forward of midchord between $\bar{t} = 0$ and 0.2. How-

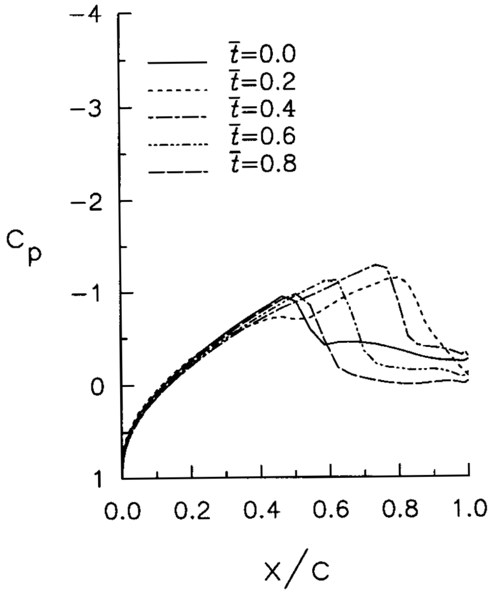

Fig. 15. Upper surface pressure coefficients for circular-arc airfoil; $M = 0.76$, CFL3D, 185×65 grid, S-A model.

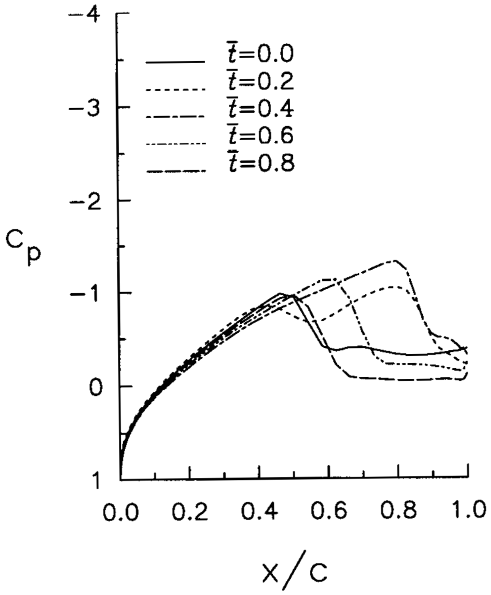

Fig. 17. Upper surface pressure coefficients for circular-arc airfoil; $M = 0.76$, TLNS3D, 185×65 grid, S-A model.

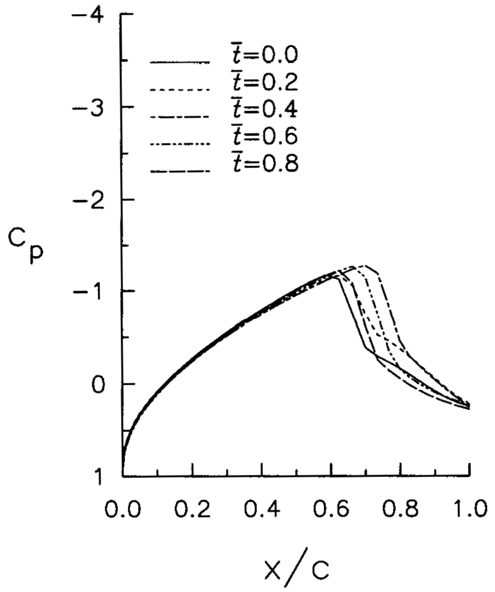

Fig. 16. Upper surface pressure coefficients for circular-arc airfoil; $M = 0.76$, CFL3D, 185×65 grid, B-L model.

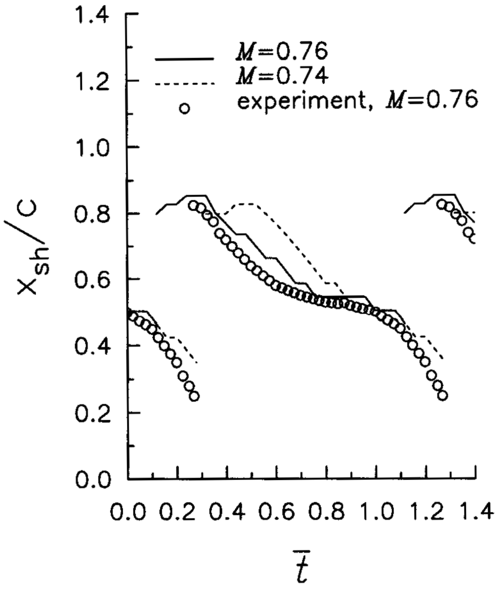

Fig. 18. Effect of Mach number on shock location on surface of circular-arc airfoil; CFL3D, 185×65 grid, S-A model.

ever, the predicted frequency of $k = 0.514$ is somewhat higher than the free-air computations and the experimental results. The upper surface pressure coefficients taken at five times in the unsteady cycle are given in figure 20. Figure 21 is a plot of pressure contours over the entire airfoil at one time in the cycle; the figure shows the relative positions of the upper and lower surface shocks. This figure corresponds to $\bar{t}$ of about 0.15, when the shock is beginning to “disappear” forward of the midchord on the upper surface at the same time that a new shock is forming near the trailing edge.

Conclusions

Accuracy and Efficiency of Subiterative Schemes

The use of subiterations in both explicit and implicit Navier-Stokes codes has been explored for unsteady flows. Two different methods have been examined. One uses pseudo time subiterations (τ-TS) in combination

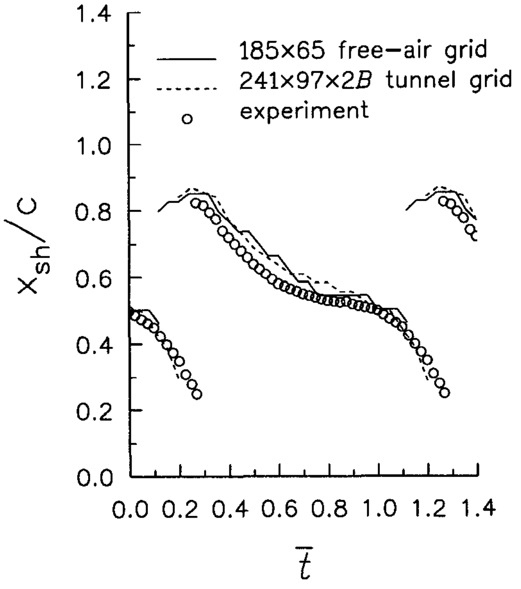

Fig. 19. Effect of tunnel grid on shock location on surface of circular-arc airfoil; $M_\infty = 0.76$, CFL3D, S-A model.

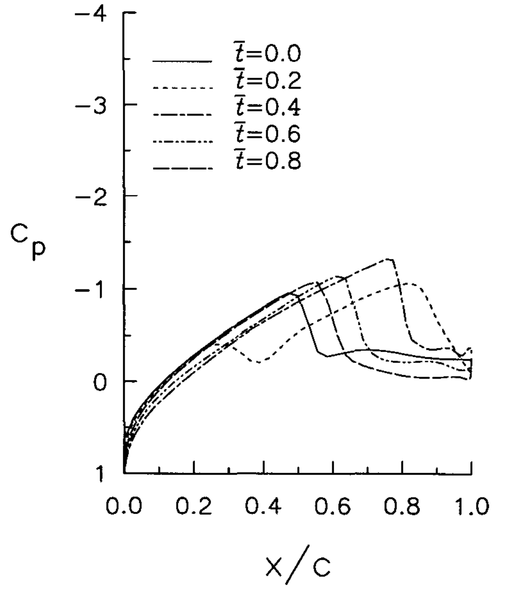

Fig. 20. Upper surface pressure coefficients for circular-arc airfoil in tunnel; $M_{x/c=-1.33} \approx 0.743$, CFL3D, 241 × 97 × 2$B$ grid, S-A model.

with multigrid both to improve accuracy and to overcome any inherent physical time-step limitations; the other method, of practical use only in the implicit code, uses physical time subiterations (t-TS) to improve accuracy.

Both laminar flow over a circular cylinder and turbulent flow over an 18-percent-thick circular-arc airfoil have been examined. For laminar flow, the choice of type of subiterative scheme (if any) in an implicit approximate-factorization code such as CFL3D is a tradeoff between accuracy and central processing unit (CPU) time. In general, the t-TS method (which requires three subiterations per time step for the time-step ranges and grids utilized in this study) yields the most accurate result for a given CPU time. The method with no subiterations can be as accurate for a given CPU time as the τ-TS method (which requires eight three-level multigrid subiterations). However, the method with no subiterations is at best first-order accurate in time due to linearization and factorization errors. In practice, the t-TS method and the method with no subiterations possess time-step limits that may be impractically small for certain problems. In such cases, the τ-TS method allows larger time steps. For an explicit code such as TLNS3D, only a pseudo time subiterative technique such as τ-TS, which removes the restrictive explicit-code time-step limitation, is practical.

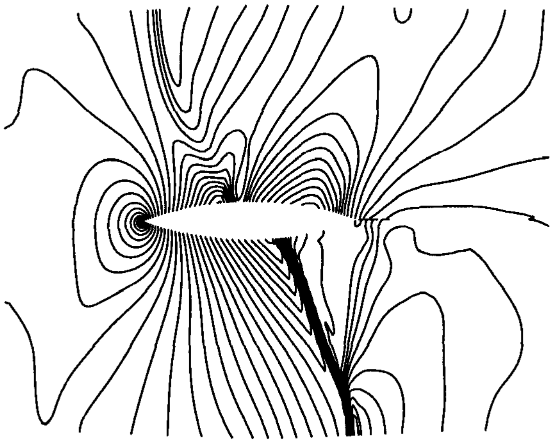

Fig. 21. Unsteady pressure contours; $M_{x/c=-1.33} \approx 0.743$, CFL3D, 241 × 97 × 2$B$ grid, S-A model.

In spite of the fact that second-order temporal discretizations are used in the numerical methods, small time steps are necessary to fully realize second-order temporal accuracy. For the circular cylinder flow, on the order of 500 time steps per period are required.

For turbulent flow over the circular-arc airfoil, approximately fifteen three-level multigrid τ-TS subiterations are required for the time-step ranges and grids utilized in this study. Unlike the laminar-flow results, the τ-TS subiterations yield a given level of accuracy in significantly less CPU time than the scheme with no subiterations when the Spalart-Allmaras (S-A) turbulence model[15] is employed. The t-TS method in conjunction with the S-A model requires an excessive number of subiterations; the t-TS method in conjunction with the Baldwin-Lomax (B-L) turbulence model[14] does not. This discrepancy requires further study. Both the CFL3D and TLNS3D codes yield similar reduced frequencies; the effect of time step and grid size has been assessed for both codes.

Comparison with 18-Percent-Thick Circular-Arc Airfoil Experiment

Thin-layer Navier-Stokes computations used with the S-A turbulence model accurately predict the experimental reduced frequency of oscillations on the circular-arc airfoil, as well as the Mach-number range within which hysteresis takes place. The motion of the shock over the airfoil surface at $M = 0.76$ with both CFL3D and TLNS3D agrees well with experimental results, except that the motion of the shock forward of midchord is not predicted. Computations at a slightly lower Mach number of $M = 0.74$ indicate a sensitivity of the shock motion to Mach number, including the extent of the forward motion. When wind-tunnel walls are modeled and the experimental Mach number is duplicated, the predicted motion of the shock agrees well with experimental results at a reduced frequency that is approximately 9 percent higher than the experimental reduced frequency. The B-L turbulence model yields a lower lift amplitude (about one-third the magnitude) and a lower frequency (by about 20 percent) than the S-A model. Also, the B-L model does not predict as large a range of shock motion as the S-A model.

Acknowledgments

The authors thank John Edwards, Jim Thomas, Jerry South, and Robert Bartels for their discussions during the course of this investigation.

References

[1] Pulliam, T., "Time Accuracy and the Use of Implicit Methods," AIAA 93-3360-CP, July 1993.

[2] Newsome, R., "Navier-Stokes Simulation of Wing Tip and Wing Juncture Interactions for a Pitching Wing," AIAA 94-2259, June 1994.

[3] Jameson, A., "Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings," AIAA 91-1596, June 1991.

[4] Melson, D., Sanetrik, M., and Atkins, H., "Time-Accurate Navier-Stokes Calculations with Multigrid Acceleration," *Sixth Copper Mountain Conference on Multigrid Methods*, Part 2, NASA CP-3224, 1993, pp. 423–437. Also in *11th AIAA CFD Conference*, Part 2, Orlando, Florida, July 1993, pp. 1041–1042.

[5] Vatsa, V., and Wedan, B., "Development of a Multigrid Code for 3-D Navier-Stokes Equations and its Application to a Grid-Refinement Study," *Computers and Fluids*, Vol. 18, No. 4, 1990, pp. 391–403.

[6] Thomas, J., Krist, S., and Anderson, W., "Navier-Stokes Computations of Vortical Flows Over Low Aspect-Ratio Wings," *AIAA Journal*, Vol. 28, No. 2, 1990, pp. 205–212.

[7] McDevitt, J., "Supercritical Flow About a Thick Circular-Arc Airfoil," NASA TM-78549, January 1979.

[8] Mabey, D., "Oscillatory Flows From Shock-Induced Separations on Biconvex Aerofoils of Varying Thickness in Ventilated Wind Tunnels," AGARD CP 296, *Proceedings of AGARD Conference on Boundary Layer Effects on Unsteady Airloads*, Aix-en-Provence, September 1980, pp. 11-1–11-14.

[9] Levy, Jr., L., "Experimental and Computational Steady and Unsteady Transonic Flows about a Thick Airfoil," *AIAA Journal*, Vol. 16, No. 6, 1978, pp. 564–572.

[10] Steger, J., "Implicit Finite-Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries," *AIAA Journal*, Vol. 16, No. 7, 1978, pp. 679–686.

[11] Edwards, J., and Thomas, J., "Computational Methods for Unsteady Transonic Flows," *Unsteady Transonic Aerodynamics, AIAA Progress in Astronautics and Aeronautics*, Nixon, D. (ed.), Vol. 120, Washington, D.C., 1989, pp. 211–261.

[12] Le Balleur, J., and Girodroux-Lavigne, P., "A Viscous-Inviscid Interaction Method for Computing Unsteady Transonic Separation," ONERA TP 1985-5, 1985.

[13] Edwards, J., "Transonic Shock Oscillations Calculated with a New Interactive Boundary Layer Coupling Method," AIAA 93-0777, January 1993.

[14] Baldwin, B., and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flow," AIAA 78-257, January 1978.

[15] Spalart, P., and Allmaras, S., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA 92-0439, January 1992.

[16] Roe, P., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[17] Van Leer, B., "Flux Vector Splitting for the Euler Equations," *Proceedings of the Eighth International Conference on Numerical Methods in Fluid Dynamics*, Krause, E. (ed.), Springer-Verlag, New York, 1982, pp. 501–512.

[18] Rumsey, C., Thomas, J., Warren, G., Liu, G., "Upwind Navier-Stokes Solutions for Separated Periodic Flows," *AIAA Journal*, Vol. 25, No. 4, 1987, pp. 535–541.

[19] McDevitt, J., Levy, Jr., L., Diewert, G., "Transonic Flow About a Thick Circular-Arc Airfoil," *AIAA Journal*, Vol. 14, No. 5, 1976, pp. 606–613.

Appendix

Table A.1. Summary of Circular-Arc Airfoil Computations Performed

#	Code	Grid	Turb	*M*	Method	Δt	Subits	MGlev	*k*	steps/period	sec/period
1	CFL3D	185 × 65	S-A	0.76	no sub	0.000625	N/A	1	0.470	14067	983(C)
2	CFL3D	185 × 65	S-A	0.76	no sub	0.00125	N/A	1	0.445	7437	526(C)
3	CFL3D	185 × 65	S-A	0.76	no sub	0.0025	N/A	1	0.404	4096	284(C)
4	CFL3D	185 × 65	S-A	0.76	no sub	0.003	N/A	1	0.389	3540	246(C)
5	CFL3D	185 × 65	S-A	0.76	τ-TS	0.05	5	3	0.454	182	314(Y)
6	CFL3D	185 × 65	S-A	0.76	τ-TS	0.05	10	3	0.486	170	584(Y)
7	CFL3D	185 × 65	S-A	0.76	τ-TS	0.05	15	3	0.492	168	364(C)
8	CFL3D	185 × 65	S-A	0.76	τ-TS	0.05	20	3	0.492	168	1150(Y)
9	CFL3D	185 × 65	S-A	0.76	τ-TS	0.1	15	3	0.475	87	189(C)
10	CFL3D	185 × 65	S-A	0.76	τ-TS	0.025	15	3	0.495	334	724(C)
11	CFL3D	185 × 65	S-A	0.76	τ-TS	0.0125	15	3	0.494	670	1445(C)
12	CFL3D	185 × 65	B-L	0.76	τ-TS	0.05	15	3	0.394	210	938(Y)
13	CFL3D	369 × 129	S-A	0.76	τ-TS	0.05	15	3	0.470	176	3445(Y)
14	CFL3D	193 × 69 (45c)	S-A	0.76	τ-TS	0.05	15	3	0.492	168	968(Y)
15	CFL3D	185 × 65	S-A	0.76	t-TS	0.005	3	1	0.431	1920	397(C)
16	CFL3D	185 × 65	S-A	0.76	t-TS	0.005	6	1	0.462	1790	1745(Y)
17	CFL3D	185 × 65	S-A	0.76	t-TS	0.005	9	1	0.474	1744	1033(C)
18	CFL3D	185 × 65	S-A	0.76	t-TS	0.005	15	1	0.481	1718	1693(C)
19	CFL3D	185 × 65	S-A	0.76	t-TS	0.005	20	1	0.484	1708	2218(C)
20	CFL3D	185 × 65	B-L	0.76	t-TS	0.005	3	1	0.395	2092	755(Y)
21	CFL3D	185 × 65	S-A	0.74	τ-TS	0.05	15	3	0.477	178	914(Y)
22	CFL3D	241 × 97 × 2*B*	S-A	0.743 @ $x/c = -1.33$	τ-TS	0.05	15	3	0.514	161	1291(C)
23	CFL3D	257 × 81	S-A	0.76	τ-TS	0.05	15	3	0.486	170	1481(Y)
24	CFL3D	257 × 81	S-A	0.75 ↓	τ-TS	0.05	15	3	0.493	170	1481(Y)
25	CFL3D	257 × 81	S-A	0.74 ↓	τ-TS	0.05	15	3	0.474	179	1568(Y)
26	CFL3D	257 × 81	S-A	0.73 ↓	τ-TS	0.05	15	3	0.441	195	692(C)
27	TLNS3D	185 × 65	S-A	0.76	τ-TS	0.05	5	3	0.437	189	1018(Y)
28	TLNS3D	185 × 65	S-A	0.76	τ-TS	0.05	10	3	0.478	173	1712(Y)
29	TLNS3D	185 × 65	S-A	0.76	τ-TS	0.05	15	3	0.492	168	1183(C)
30	TLNS3D	185 × 65	S-A	0.76	τ-TS	0.05	20	3	0.492	168	3224(Y)
31	TLNS3D	185 × 65	S-A	0.76	τ-TS	0.1	15	3	0.465	89	617(C)
32	TLNS3D	185 × 65	S-A	0.76	τ-TS	0.025	15	3	0.498	332	2335(C)
33	TLNS3D	185 × 65	S-A	0.76	τ-TS	0.0125	15	3	0.497	665	4741(C)
34	TLNS3D	369 × 129	S-A	0.76	τ-TS	0.05	15	3	0.459	180	9010(Y)
35	TLNS3D	193 × 69 (45c)	S-A	0.76	τ-TS	0.05	15	3	0.489	169	1123(C)