Journal of Computational Physics **179,** 313–329 (2002)
doi:10.1006/jcph.2002.7059

Implicit Time Integration Schemes for the Unsteady Compressible Navier–Stokes Equations: Laminar Flow

Hester Bijl,* Mark H. Carpenter,† Veer N. Vatsa,† and Christopher A. Kennedy‡

*Department of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands; †Computational Method and Simulation Branch, NASA Langley Research Center, Hampton, Virginia 23681-0001; and ‡Combustion Research Facility, Sandia National Laboratories, Livermore, California 94551-0969
E-mail: h.bijl@lr.tudelft.nl

Received July 10, 2001; revised March 5, 2002

The accuracy and efficiency of several lower and higher order time integration schemes are investigated for engineering solution of the discretized unsteady compressible Navier–Stokes equations. Fully implicit methods tested are either the backward differentiation formulas (BDF) or stage-order two, explicit, singly diagonally implicit Runge–Kutta (ESDIRK) methods. For this comparison an unsteady two-dimensional laminar flow problem is chosen: flow around a circular cylinder at $\mathrm{Re} = 1200$. At temporal error tolerances consistent with engineering simulation, $\epsilon \approx 10^{-1}$–10^{-2}, first-order implicit Euler (BDF1) is uncompetitive. While BDF3 is quite efficient, its lack of A-stability may be problematic in the presence of convection. At these same error tolerances, the fourth-order ESDIRK scheme is 2.5 times more efficient than BDF2. It is concluded that reliable integration is most efficiently provided by fourth-order Runge–Kutta methods for this problem where order reduction is not observed. Efficiency gains are more dramatic at smaller tolerances. © 2002 Elsevier Science (USA)

INTRODUCTION

The vast majority of flowfields encountered in engineering applications are unsteady. These flowfields may include inherently unsteady flow separation, unsteady boundary- and free-shear flows, as well as unsteady boundary conditions, possibly caused by flow actuators. In spite of this, since one is often concerned with only mean flow quantities and steady flowfields are less computationally demanding, unsteady flow simulations have historically been more rare. Contemporary computer resources now render three-dimensional unsteady flowfield computation viable [1, 4]. With this transition in focus from steady to unsteady simulations, it is appropriate to examine the relative efficiency of fully implicit

0021-9991/02 $35.00
© 2002 Elsevier Science (USA)
All rights reserved.

temporal integration techniques for unsteady compressible flows of engineering interest. As these simulations consume vast computational resources, their efficiency is of great importance. Further, the commonly held notion that higher order methods have no application in engineering contexts needs examination. Marx [12] has considered this same issue but only used methods of orders one and two.

For implicit methods, efficiency may be conveniently measured by the work required to maintain some particular fixed local integration error for any given simulation. Controlling the local error will tend to control the global error. As error tolerances are reduced, higher order methods are more efficient; hence, efficiency is tolerance dependent. For example, Kennedy *et al*. [8] compare explicit third-, fourth-, and fifth-order Runge–Kutta (RK) schemes and find that fourth-order methods are more efficient than third-order ones at $\epsilon < 10^{-1}$, but fifth-order methods are more efficient than fourth-order methods at $\epsilon < 10^{-3}$–10^{-5}. Efficient methods that are not reliable or have other significant shortcomings are of less interest.

The time step required to maintain a fixed temporal accuracy varies during a typical simulation. Therefore, a fixed time step is not optimal, as it must be chosen to achieve the desired local error tolerance during the most demanding portion of the simulation. To avoid this, efficient methods must also possess an error estimation procedure and an error controller to adjust the time step accordingly.

The hallmark of large-scale aerodynamics calculations, as well as many other engineering simulations, is that they seldom require relative error tolerances tighter than $\epsilon \approx 10^{-1}$–10^{-2}. Calculations that are accurate to one or two significant digits are frequently sufficient. Typical simulations utilize first- or second-order methods without error estimation/step-size control. For these aerodynamic calculations, the second-order-accurate BDF2 scheme is often preferred. The central question of this study then becomes whether higher order methods are more or less efficient than traditional methods at engineering error tolerances while maintaining the robustness of BDF2.

A production aerodynamics solver is needed for this study. For this, the extensively tested and well-documented solver of Vatsa (TLNS3D) [18] is chosen. This multiblock structured grid solver is representative of a broad class of commonly used flow solvers. The TLNS3D (Thin-Layer Navier–Stokes 3D) code utilizes a special form of the unsteady thin-layer Navier–Stokes equations. The spatial terms are discretized using a conventional cell-centered finite-volume scheme with artificial dissipation added for stability. Fully implicit time discretization is accomplished using either fixed coefficient BDF or ESDIRK methods. All simulations are conducted at fixed step size. The resultant nonlinear algebraic equations are solved iteratively in pseudotime with a multigrid acceleration used to speed up the convergence to (pseudotime) steady state.

In this paper, first, the governing flow equations and the space discretization are given. Thereafter, the time discretization techniques employed are extensively discussed. A brief description of the solution algorithm follows. After validation of the space-time method for unsteady laminar flow around a circular cylinder, accuracy and efficiency of the time integration schemes are discussed.

GOVERNING EQUATIONS/SPATIAL DISCRETIZATION

In the present work, a modified version of the thin-layer Navier–Stokes equations is used to model the flow. The equation set is obtained from the complete Navier–Stokes equations

by retaining only the viscous diffusion terms normal to the solid surfaces. For a body-fitted coordinate system (ξ, η, ζ) fixed in time, these equations can be written in the conservative form as

$$\frac{\partial(\mathbf{U})}{\partial t} = -\frac{\partial(\mathbf{F} - \mathbf{F}_\mathbf{v})}{\partial \xi} - \frac{\partial(\mathbf{G} - \mathbf{G}_\mathbf{v})}{\partial \eta} - \frac{\partial(\mathbf{H} - \mathbf{H}_\mathbf{v})}{\partial \zeta} = \hat{\mathbf{S}}(\mathbf{U}(t)), \tag{1}$$

where $\mathbf{U}$ represents a combination of the transformation Jacobian $\mathbf{J}$ and the conserved variable vector. The vectors $\mathbf{F}$, $\mathbf{G}$, $\mathbf{H}$, and $\mathbf{F}_\mathbf{v}$, $\mathbf{G}_\mathbf{v}$, $\mathbf{H}_\mathbf{v}$ represent the convective and diffusive fluxes in the three transformed coordinate directions, respectively. These equations represent a generalized form of the classical thin-layer Navier–Stokes equations and include all normal components of the viscous stress terms. The TLNS3D computer code is used in this study to solve Eq. (1). The spatial terms in Eq. (1) are discretized using a standard cell-centered finite-volume scheme. The convection terms are discretized with second-order central differences with scalar/matrix artificial dissipation (second- and fourth-order-difference dissipation) added to suppress the odd–even decoupling and oscillations in the vicinity of shock waves and stagnation points [18]. The viscous terms are central differenced with second-order formulas. Many references detail the discretization and implementation issues of TLNS3D (see the work of Vatsa and Wedan [18] for further details).

TEMPORAL DISCRETIZATION

Consider the integration of the system of ordinary differential equations (ODEs) represented by the equation [6, 7]

$$\frac{d\mathbf{U}}{dt} = \mathbf{S}(\mathbf{U}(t)).$$

In the present case, the vector $\mathbf{S}$ results from the semidiscretization of the equations of fluid mechanics as given by Eq. (1). It contains convection, and diffusion, but could include reaction, radiation, and/or buoyancy. The responsibility of the integrator is to integrate any $\mathbf{S}$ with which it is provided. Trouble often arises when the Jacobian of $\mathbf{S}$, $\partial\mathbf{S}/\partial\mathbf{U}$, has large eigenvalues. This may give rise to stiffness. Nontrivial near-wall stiffness is not uncommon in practical engineering problems due to grid clustering and increases with Reynolds number. A useful definition for stiffness states that a problem is stiff when the largest scaled eigenvalue of the Jacobian, $\|z = \lambda(\Delta t)\|$, contained in the complex left-half-plane (LHP) becomes much greater than unity. The resulting stiffness is then governed by both the Jacobian and the chosen time step. Ideally, the time step is selected solely based on error considerations and a good method simply executes this step size in a stable and robust fashion. Time integration methods that do not amplify *any* LHP-scaled eigenvalues are called A-stable. While A-stability is generally necessary, it is often not sufficient. We further demand that all eigenvalues, $\|z \to -\infty\|$, be completely damped. Hence, in the paper we consider only L-stable methods and the somewhat less-desirable and stable L(α) methods. Making this choice not only avoids numerical instability, it also facilitates convergence of the nonlinear equation solver.

Popular implicit ODE integration methods are generally either distinctly multistep or multistage methods. Each has different strengths and weaknesses. Implicit multistep BDF

methods compute each **U**-vector update to design order of accuracy using one nonlinear equation solve per step. Unfortunately, they are not A-stable above second order. Additionally, they are not self-starting and have diminished properties when used in a variable step-size context. Practical experience indicates that large-scale engineering computations are seldom stable if run with BDF4 [13]. The BDF3 scheme, with its smaller regions of instability, is often stable but diverges for certain problems and some spatial operators. Thus, a reasonable practitioner might use the BDF2 scheme exclusively for large-scale computations due to its L-stability rather than L(α)-stability. This explains why BDF2 is one of the current methods of choice in the computation of large-scale engineering flows. Practical RK methods such as ESDIRK methods can be made arbitrarily high order while retaining L-stability but possess intermediate **U**-vectors with reduced order of accuracy and lesser stability. This reduced stage order may give rise to order-reduction phenomena in the presence of substantial stiffness. ESDIRK schemes with s stages require $(s-1)$ nonlinear equation solves per step. Although it is possible to achieve progressively higher stage-order methods, such as the Radau IIA family, this is not likely to be practical in the current context. There is much less experience with implicit RK methods than BDF methods in the computation of large-scale engineering flows.

The general formula for a k-step, order-k, BDF scheme can be written as

$$\mathbf{U}^{(n+k)} = -\sum_{i=0}^{k-1} \alpha_i \mathbf{U}^{(n+i)} + (\Delta t)\beta_k \mathbf{S}^{(n+k)}. \tag{2}$$

At each time-step the BDF formulas involve the storage of $k+1$ levels of the solution vector **U**, and the implicit solution of one set of nonlinear equations. Stability diagrams for these methods may be found in Hairer and Wanner [7]. At order $k > 2$ one finds an unstable zone for scaled eigenvalues in the complex LHP. At orders $\{1, 2, 3, 4, 5, 6\}$ the methods are L(α)-stable where α is given by $\{90, 90, 86.03, 73.35, 51.84, \text{ and } 17.84^\circ\}$. For these same orders, β_k is given by $\{1, 2/3, 6/11, 12/25, 60/137, 60/147\}$. Smaller values of β_k facilitate iterative convergence of the nonlinear algebraic system at each step.

ESDIRK methods [9, 10] are implemented as

$$\begin{aligned}
\mathbf{U}^k &= \mathbf{U}^n + (\Delta t)\sum_{j=1}^{k} a_{kj}\mathbf{S}(\mathbf{U}^j), \quad k = 1, s, \\
\mathbf{U}^{n+1} &= \mathbf{U}^n + (\Delta t)\sum_{j=1}^{s} b_j \mathbf{S}(\mathbf{U}^j), \\
\hat{\mathbf{U}}^{n+1} &= \mathbf{U}^n + (\Delta t)\sum_{j=1}^{s} \hat{b}_j \mathbf{S}(\mathbf{U}^j),
\end{aligned} \tag{3}$$

where s is the number of stages, a_{kj} are the stage weights, and b_i and $\hat{b}_j$ are the main and embedded scheme weights. The vectors **U** and $\hat{\mathbf{U}}$ are the pth-order and $(p-1)$th-order solutions at time level $n+1$. The vector $\hat{\mathbf{U}}$ is used solely for estimating error and is virtually free. The Butcher tableau for stiffly accurate ESDIRK schemes (here represented

with $s = 5$) takes the form

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ c_2 & a_{21} & \gamma & 0 & 0 & 0 \\ c_3 & a_{31} & a_{32} & \gamma & 0 & 0 \\ c_4 & a_{41} & a_{42} & a_{43} & \gamma & 0 \\ 1 & b_1 & b_2 & b_3 & b_4 & \gamma \\ \hline & b_1 & b_2 & b_3 & b_4 & \gamma \\ \hline & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \hat{b}_4 & \hat{b}_5 \end{array} ,$$

where c_i are the abscissas that denote the point in the time, $t + c_i \Delta t$, where the stage i is evaluated. ESDIRK schemes differ from traditional SDIRK methods (see Section IV.6 in Hairer and Wanner [7]) by the choice $a_{11} = 0$, which permits stage-order-two methods. The stiffly accurate assumption ($a_{sj} = b_j$) makes the new solution $\mathbf{U}^{n+1}$ independent of any *explicit* process within the integration step. Three representative ESDIRK methods were chosen for this study: a third-, fourth-, and fifth-order method. The first two, ESDIRK3 and ESDIRK4, are taken directly from Kennedy and Carpenter [9] while ESDIRK5 was constructed specifically for this study. ESDIRK3, ESDIRK4, and ESDIRK5 have $s = 4, 6, 7$ and $\gamma \approx 0.4359$, $\gamma = 0.2500, 0.1840$, respectively. The coefficients for ESDIRK3 and ESDIRK4 are included in the Appendix. While no major problems were encountered with ESDIRK5 in this laminar study, design shortcomings surfaced when applying the scheme to turbulent flows. As such the coefficients for the ESDIRK5 scheme are not included in this work.

SOLUTION ALGORITHM

Discretizing Eq. (1) with an s-stage ESDIRK scheme represented in Eq. (3) yields for stage k the expression

$$\frac{\mathbf{U}^k - \mathbf{U}^n}{\Delta t} + \sum_{j=1}^{k} a_{kj} \mathbf{S}^j = 0, \quad k = 1, s \tag{4}$$

with

$$\mathbf{S}^j = \frac{\partial(-\mathbf{F} + \mathbf{F}_\mathbf{v})}{\partial \xi} + \frac{\partial(-\mathbf{G} + \mathbf{G}_\mathbf{v})}{\partial \eta} + \frac{\partial(-\mathbf{H} + \mathbf{H}_\mathbf{v})}{\partial \zeta}.$$

The vector $\mathbf{U}^k$ is the solution at stage k, $\mathbf{U}^n$ is the solution at the previous time level n, and a_{kj} are the Butcher parameters for the ESDIRK method used. Again note that in our case the last stage gives the solution at the new time level; that is, $\mathbf{U}^{n+1} = \mathbf{U}^s$. Advancing the solution vector $\mathbf{U}^k$ from time level $n \rightarrow n + 1$ requires solving the sequential set of $s - 1$ nonlinear algebraic equations defined in Eq. (4).

PSEUDO-TIME-ITERATIVE ALGORITHM

We follow the work of Melson *et al.* [13], which was originally developed for a BDF algorithm. Equation (4) can be difficult to solve in its present form. Thus, the pseudotime

term $\partial \mathbf{U}^k/\partial\tau$ is added to each stage k, yielding the expression

$$\frac{\partial \mathbf{U}^k}{\partial \tau} + \frac{\mathbf{U}^k - \mathbf{U}^n}{\Delta t} + \sum_{j=1}^{k} a_{kj}\mathbf{S}^j = 0, \quad k = 1, s. \tag{5}$$

In this form, Eq. (5) is amenable to all the nonlinear solving machinery available in TLNS3D. Each nonlinear equation is marched in pseudotime with multigrid acceleration until a predetermined convergence criterion is satisfied. Equation (5) can be rewritten in the form

$$\frac{\partial \mathbf{U}^k}{\partial \tau} + \frac{\mathbf{U}^k}{\Delta t} + \gamma \mathbf{S}^k + E = 0, \quad k = 1, s. \tag{6}$$

with

$$E = \frac{-\mathbf{U}^n}{\Delta t} + \sum_{j=1}^{k-1} a_{kj}\mathbf{S}^j,$$

where E includes all the iteration-independent information at stage k of the ESDIRK scheme. Discretizing Eq. (6) in the variable τ using an Runge–Kutta (RK_τ) scheme yields

$$\frac{\mathbf{U}^{k^q} - \mathbf{U}^{k^0}}{\Delta \tau} + \alpha_q \left(\frac{\mathbf{U}^{k^q}}{\Delta t} + \gamma \mathbf{S}^{k^{(q-1)}} + E \right) = 0, \quad q = 1, 5, \tag{7}$$

where the q is the stage number of the RK_τ scheme. Note that the contribution from the time term is treated implicitly, that the inviscid and viscous flux terms are treated explicitly, and that first-order temporal accuracy is sufficient for this scheme. Adding, subtracting, rearranging terms, and accounting for residual smoothing in Eq. (7) yields

$$\left(1 + \frac{\alpha_q \Delta \tau}{\Delta t}\right)\mathbf{U}^{k^q} = \mathbf{U}^{k^0} + \frac{\alpha_q \Delta \tau}{\Delta t}\mathbf{U}^{k^{q-1}} - \alpha_q \Delta \tau \mathbf{L}_{irs}^{-1}\left[\frac{\mathbf{U}^{k^{q-1}}}{\Delta t} + \gamma \mathbf{S}^{k^{q-1}} + E\right], \tag{8}$$

where $\mathbf{L}_{irs}^{-1}$ is the implicit residual smoothing matrix. Note that the bracketed term in Eq. (8) is still the residual of the physical time ESDIRK scheme. Bijl *et al*. [2] give a pseudotime analysis of Eq. (8) and determine the optimal values of the pseudo-time-step $\Delta\tau$ and the residual smoothing parameters. Only their major conclusions are presented in this work.

The stability analysis for each stage is identical because the diagonal coefficient γ is the same on each stage of an ESDIRK. The stability analysis for the BDF scheme given by Eq. (2) is similar, with the variable γ replaced by β_k. The independent parameters in Eq. (8) are $R = \Delta\tau/\Delta t$, the ratio of time steps; $\hat{\xi} = \gamma CFL$, the scaled pseudo-time-step CFL condition; and $\mathcal{R}$, the implicit residual smoothing parameter. An exhaustive study of R, $\hat{\xi}$, and $\mathcal{R}$ reveals that the residual smoothing is most productive when the effective stability limit of the subiteration is increased by a factor of 2–3. The pseudotime subiteration is stable for $\hat{\xi} \approx 7$ and has the best damping characteristics in the $\frac{\pi}{2} \leq \theta \leq \pi$ range for values of $\hat{\xi}$ near the stability limit. These parameters are nearly identical to those used in the steady-state solver TLNS3D. Extensive details of the underlying steady-state solver can be found in the work of Vatsa and Wedan [18]. The same concept for the computation of unsteady flows is used by other researchers (see, for example, Martinelli and Jameson [11] and Swanson and Turkel [16]).

DIAGONAL COEFFICIENTS

The pseudotime subiteration is strongly influenced by the diagonal coefficients γ and β_k. In TLNS3D the pseudotime subiteration is always advanced with the maximum allowable *scaled* pseudotime step, $\hat{\xi} \approx 7$. The rate of relaxation in *nonscaled* pseudotime is therefore inversely proportional to the diagonal coefficient γ: the smaller the value of γ, the more rapidly the pseudotime subiteration progresses. The values γ and β_k vary significantly for different physical time-advancement schemes. The β_k coefficients for the BDF schemes vary by a factor of approximately two. The γ coefficients for the ESDIRK schemes vary by approximately a factor of two and are generally much smaller than the BDF β_k values.

The ESDIRK schemes can be optimized to improve their efficiency. The important parameters are γ and the number of implicit stages, $s - 1$. A fortuitous trend observed with the stiffly accurate, L-stable ESDIRK schemes is a general decrease in the value of γ with increasing order and stage number. Increasing the number of implicit stages does not always decrease the efficiency of the scheme; it sometimes yields *greater* efficiency. The value $\gamma = \frac{1}{4}$ in the ESDIRK4 scheme is an example of five implicit stages producing a more efficient fourth-order scheme than four implicit stages produce with $\gamma \approx 0.5728$.

VALIDATION OF THE SPACE–TIME METHOD

The accuracy of the space–time integration methods is investigated for an unsteady laminar flow test case. The test problem is laminar flow around a two-dimensional circular cylinder at a Reynolds number of 1200 and a Mach number of 0.3. Reference nondimensionalization is based on infinity values and cylinder diameter. The initial flow is symmetric with zero lift. As the wake behind the cylinder starts to grow, it becomes unstable and begins to shed from alternate sides of the cylinder. Detailed numerical and experimental investigations of this flow have been performed by several authors [3, 5, 15, 16]. The variety of grids used in this study ranged from 73×49 to 289×193. The computational boundary is located at a distance of 20 diameters, while the distance between the wall and the first grid point is 0.001 times the diameter of the cylinder on the 97×65 grid. Grid points are clustered in the wake. Contour plots of the density reveal that the near-wall vortical structures appear to be sufficiently resolved even on the 97×65 grid, but that vortex resolution is lost as the grid expands in the farfield.

Figure 1 shows a comparison of the lift history for one shedding cycle as calculated on the 97×65 and 193×129 grids. The 193×129 data is shifted so that the first zero in lift coincides on both curves. Small differences in the lift are seen near the peaks of the cycle. A time-step $\Delta t = \frac{1}{2}$ is chosen, ensuring that the dominant error component is the spatial contribution. Stepwise temporal error in these calculations is 10^{-4}, 10^{-5}, for L_∞, L_2, respectively. Cumulative error increases by approximately one order of magnitude in each case.

Figure 2 shows the convergence of the Strouhal number on the sequence of five grids. The convergence rate of the spatial algorithm is approximately 1.64 (based on a Richardson extrapolated exact solution of 0.2467 and a least-squares fit of the data). The values of the Strouhal number are larger than 0.21, reported from experiments [4, 15]. As Mittal and Balachandar report [14] this is likely caused by the onset of three-dimensional effects that are not captured in our two-dimensional computations. Two-dimensional computations performed by other researchers [5, 17] resulted in larger Strouhal numbers too, in the range of 0.23–0.24. This study and an exhaustive study of inviscid vortex propagation support the

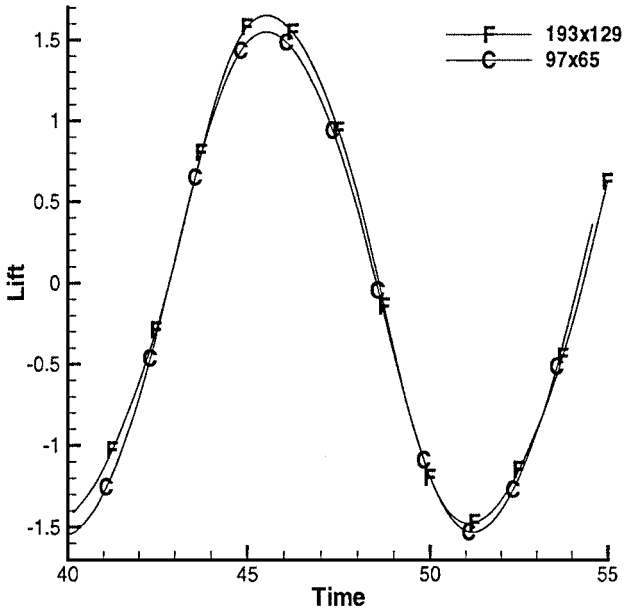

FIG. 1. Comparison of lift over one shedding cycle between coarse and fine grids.

conclusion that the space–time operator is working properly, and that the spatial operator converges on these stretched grids at nearly the design rate of two.

The objective of this study is to compare temporal schemes at engineering relative error tolerances (1–10%). It is therefore consistent to choose a spatial grid having a comparable level of error. The 97×65 grid has approximately 5% error ($0.24/0.23 - 1$) and provides spatial resolution adequate for capturing the relevant large-scale features in the shedding

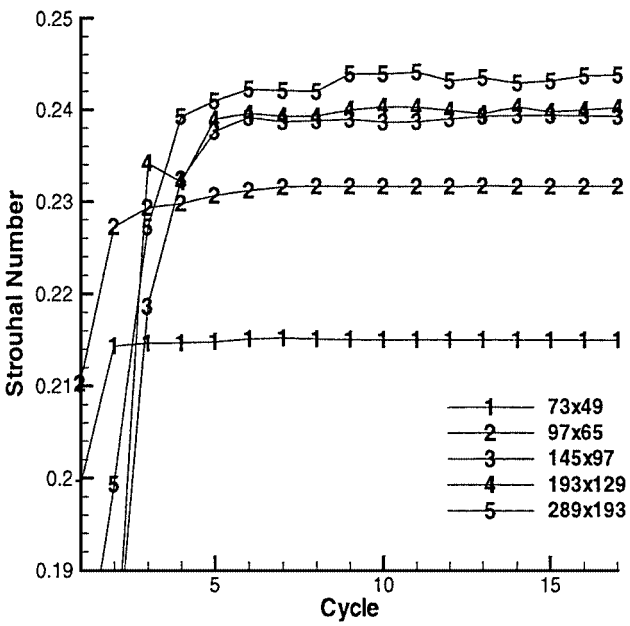

FIG. 2. Convergence of Strouhal number.

process; as such, it is chosen as the basis of most of the temporal refinement studies presented in this work.

TEMPORAL ACCURACY

A temporal refinement study is performed to assess the accuracy of various ESDIRK and BDF schemes. The initial condition for the study was obtained by simulating the limit cycle behavior of the flow for approximately 20 shedding cycles, with a relatively small time step ($\Delta t = \frac{1}{2}$). After 20 cycles, the solution was stored in a restart file for use as the initial condition in the subsequent studies. A classical temporal study was then performed from this initial condition. A typical practitioner is interested in lift, drag, pitching moment, skin friction, and the frequency spectrum over several cycles. Thus, the time interval of the study includes approximately $1\frac{1}{4}$ shedding cycles. This interval is sufficient in length to allow accumulation of temporal error during the shedding cycle. No exact solution is known for this problem, so a "numerically exact" solution was obtained using a small time step and a small iteration tolerance on the nonlinear solves. The numerically exact time step used was $\Delta t = 0.05$, and a stopping criterion for the iterative solve of the implicit equations was $max(residual) < 10^{-6}$. The "exact" solution is accurate to approximately six significant digits in the lift when run with the ESDIRK4 scheme.

The lift on the body is used as the representative measure of error in all calculations. Other integral measures including viscous drag, skin friction, total drag, pitching moment, and L_2 and L_∞ norms over the domain were studied. All integral measures yield nearly the same qualitative conclusions, although the quantitative errors are different for each case. A detailed investigation of the location of the maximum temporal error reveals that the vortex generation process in the near-wall regions is the most temporal demanding portion of the cycle. It is not surprising, therefore, that the lift integral is a good measure of total error in the calculation.

Detailed results from the study are now presented. Figure 3 shows a detailed refinement study with the ESDIRK4 scheme. Shown are logarithms of the solution errors in the lift,

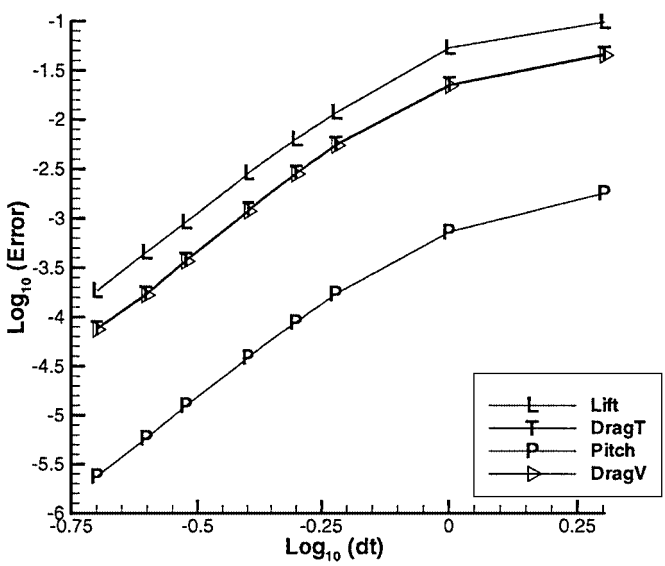

FIG. 3. Convergence behavior for lift, drag, and pitching moment as calculated with the ESDIRK4 scheme.

viscous drag, total drag, and pitching moment as a function of logarithm of time step. In all cases, the nonlinear system is solved to strict tolerances to eliminate "iteration" error as a contaminating variable in the study. At coarse time steps the solution accuracy deteriorates away from design accuracy. For sufficiently small time steps, design accuracy is clearly demonstrated in all variables. For example, a least-squares fit of the finest four data points on each curve reveals that the convergence rates for lift, drag V, drag T, and pitch are 3.9628, 4.0257, 4.0478, and 4.0251, respectively. The coarsest data point corresponds to a time step of $\Delta t = 2$, for which 12 points resolve the temporal shedding cycle. Note that 15–20 points in time are needed to resolve the cycle before the fourth-order scheme converges with design accuracy. From spatial experience, this resolution is consistent with conventional estimates of "points-per-wavelength" needed to resolve a periodic wave.

The stiffness in this problem can be inferred by comparing the physical and pseudotime steps, rather than calculating the eigenvalues λ of the spatial operator. TLNS3D explicitly advances the spatial terms $\mathbf{S}(\mathbf{U}(t))$ in pseudotime, which requires $\|Z\| = \|\lambda \Delta\tau\| \approx 1$. Accuracy considerations require that the physical time step satisfies $\Delta t = \mathcal{O}(1)$. Thus, the stiffness is $\mathcal{O}((\Delta\tau)^{-1})$. The pseudotime step satisfies $\Delta\tau \approx 0.0005$ on the 97×65 grid, which translates to a stiffness of $\mathcal{O}(10^3)$. Stiffness increases on finer grids.

Implicit Runge–Kutta schemes are susceptible to order reduction for stiff problems. Order reduction results from stiff components of the solution converging at a rate which is governed by the stage order rather than the design order of accuracy. (The ESDIRK3,4,5 schemes have a stage order of two and design orders of 3, 4, 5, respectively.) The lower convergence rate of the stiff components may or may not adversely affect the accuracy of the nonstiff components and is strongly problem dependent. The discrete solution of the cylinder ($\mathrm{Re} = 1200$, $\mathrm{Ma} = 0.3$) converges at the design order rate (3, 4, 5) for all variables of interest, including lift, drag, and pitching moment. Order reduction of the stiff components is not a critical issue in the solution of the cylinder at the stated flow conditions.

Figure 4 presents the error in lift versus the time step (log–log) for three BDF schemes and three ESDIRK schemes. Again, the nonlinear equations at each stage (step) are solved to tolerances which are small compared with the absolute error in the calculation. Accuracy increases dramatically in going from BDF1 (sometimes referred to as implicit Euler) to ESDIRK5. For example, a relative error tolerance of 10^{-1} is achieved with a time step of $\Delta t = 1$ for the ESDIRK4 and ESDIRK5 schemes, while the BDF1 and BDF2 require $\Delta t = 10^{-2}$ and 10^{-1}, respectively. The BDF3 and the ESDIRK3 schemes have nearly the same absolute level of error, although the convergence behavior of the BDF3 scheme is sporadic. No explanation of this behavior was identified.

As mentioned previously in this work, the BDF2 scheme is consistently used by practitioners because of its robustness, simplicity, and efficiency. The results in Fig. 4 clearly show that the ESDIRK4 scheme can be used at time steps that are a factor of 10 larger than those used in the BDF2 and still achieve similar accuracy. The BDF2 scheme achieves design accuracy at approximately $\Delta t = 0.1$–0.2. At larger time steps the second-order convergence behavior is not realized. At fine tolerances, this difference becomes even greater. However, Fig. 4 cannot be used to conclude that the ESDIRK4 scheme is more *efficient* than the BDF2 scheme. To do so requires detailed accounting of the work involved in each algorithm and is not a simple task.

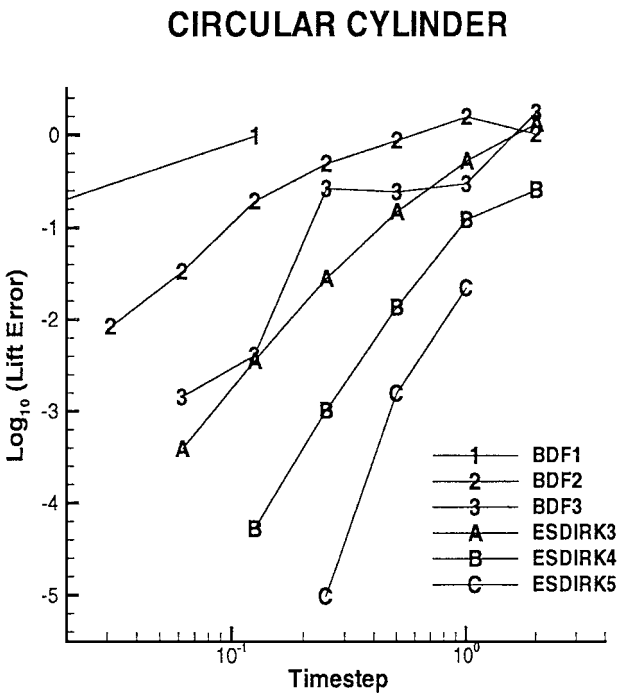

FIG. 4. Convergence as a function of time step for BDF and ESDIRK schemes.

TEMPORAL EFFICIENCY

For large computations, the work involved in advancing the solution over the interval $T_0 \leq t \leq T_f$ is proportional to the number of nonlinear solves required over that interval. The work also depends strongly on how quickly each nonlinear solve converges. The number of nonlinear solves is simply $N_L = \frac{T_f - T_0}{\Delta t} I_s$, where I_s is the number of implicit stages per time step. For the BDF schemes $I_s = 1$. How rapidly the nonlinear solve converges is a much more difficult aspect to predict. The BDF schemes benefit slightly from the smaller physical time steps because they provide a better initial guess for the nonlinear iteration. (Even though the trivial initital guess is used, effects of initial conditions are rapidly swept out of the domain by the multigrid algorithm and only slightly influence the total work.) In addition, decreasing Δt alters the parameter $R = \frac{\Delta \tau}{\Delta t}$, which slightly improves the smoothing rate of the RK_τ pseudotime solver. Conversely, the ESDIRK schemes have a much smaller diagonal coefficient γ, which significantly increases the asymptotic convergence rate of the multigrid process.

Figure 5 presents the convergence of the six schemes as a function of the required work. Three accuracy levels are chosen, 10^{-1}, 10^{-2}, and 10^{-3}, as representative of desired engineering accuracy levels. The appropriate values of Δt needed for each method are obtained from Fig. 4. For these calculations the tolerance ratio is $\mathcal{T} = \frac{1}{200}$ where $\mathcal{T}$ is defined as the ratio of *nonlinear algebraic error* to *temporal integration error* at each stage/time step. The algebraic error for the nonlinear iteration is based on the L_∞ norm of the residual. The work from each method is measured as the total number of multigrid cycles used in the entire time interval.

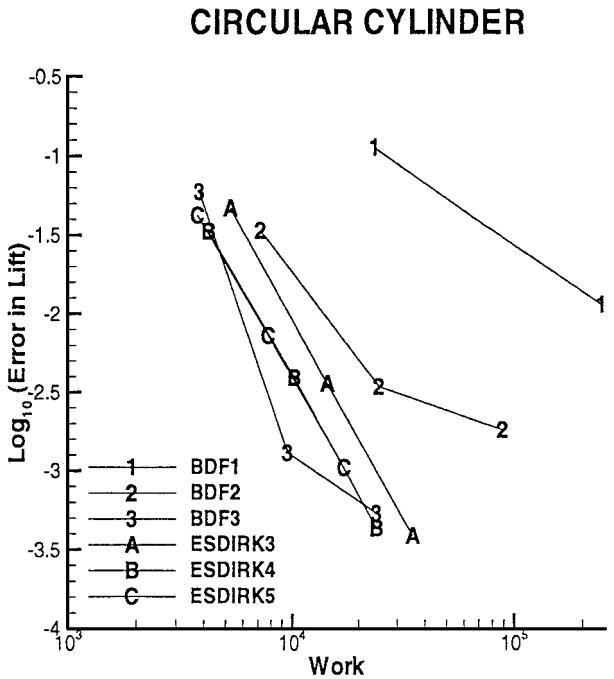

FIG. 5. Convergence as a function of work for BDF and ESDIRK schemes.

An obvious conclusion from the study presented in Fig. 5 is that the BDF1 scheme (Euler implicit) will never compete with the higher order schemes in terms of efficiency. Similarly, the BDF2 scheme is only competitive with the third-, fourth-, and fifth-order schemes at extremely coarse tolerances. For example, the BDF2 is 2.5 times less efficient than the ESDIRK4 scheme at an error tolerance of 10% (one significant digit in the solution accuracy). As the error tolerance becomes more strict, the high-order schemes easily outperform the BDF1 and the BDF2 schemes. For a desired temporal error of 1% in lift, the fourth-order integration method ESDIRK4 only requires 1.5% of the work required by BDF1, an efficiency increase with a factor of 70. For a temporal error of 0.1%, ESDIRK4 requires 10% of the work of BDF2, an efficiency increase with a factor of 10. Note that again the BDF3 scheme shows an irregular behavior. Perhaps this is because the BDF3 solution oscillates around the exact solution for different time steps and sometimes by coincidence yields unusually low levels of error. The work required for the BDF schemes appears to increase dramatically for small error tolerances. A possible explanation is that the asymptotic convergence rate of the multigrid algebraic solver is sensitive to the diagonal coefficients γ, (β_k). The ESDIRK schemes have smaller diagonal coefficients than do the BDF schemes.

The fourth- and fifth-order ESDIRK schemes have exactly the same accuracy–work ratio, and their convergence behavior appears to be logarithmic in nature. Both have different time steps, stages, and diagonal coefficients γ; there is no reason why they should lie on the same line or have logarithmic convergence behavior. Though both have the same efficiency, the ESDIRK5 scheme requires more storage and is less robust than the ESDIRK4 scheme (internal stability problems surface at huge time steps). As robustness is a major concern, the ESDIRK4 scheme is recommended over the ESDIRK5 scheme.

A major contributor to the inefficiency of implicit methods is solving the nonlinear systems at each stage (step) to inappropriate subiteration tolerance levels. If the nonlinear

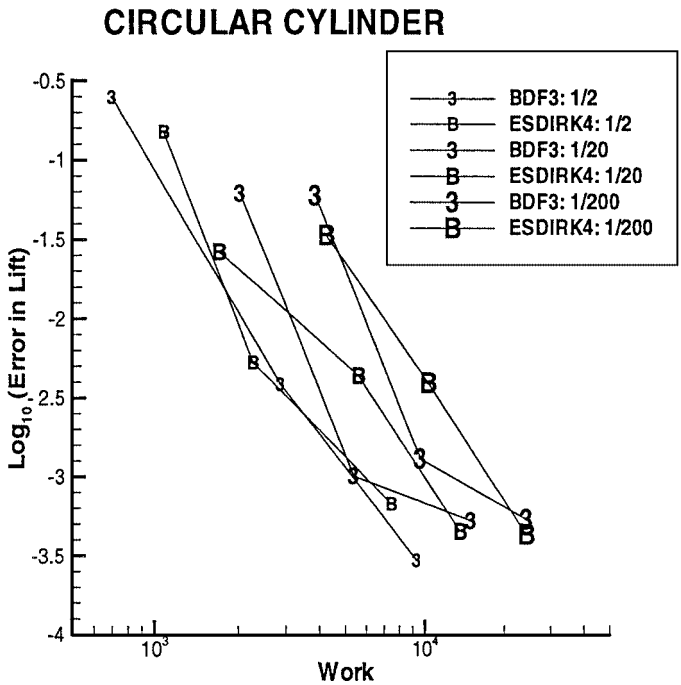

FIG. 6. The effects of tolerance ratio $\mathcal{T}$ on solution accuracy for the BDF3 and ESDIRK4 schemes.

solve is iterated too many times, the additional work does not increase the solution accuracy or robustness, but does increase the cost. If the nonlinear solve is not iterated enough, then the solution error will be dominated by the algebraic errors in the nonlinear solve, robustness will suffer, and the entire solution will be in jeopardy. The solution *cannot* be more accurate than the algebraic errors in the nonlinear equations. Unfortunately, it is not known a priori to what levels the nonlinear equations should be solved. A final study is performed by varying the tolerance ratio $\mathcal{T}$ over the interval $\frac{1}{200}$, $\frac{1}{20}$, $\frac{1}{2}$. The temporal error tolerances used in the previous study, 10^{-1}, 10^{-2}, and 10^{-3}, are again used.

Figure 6 shows a plot of the solution accuracies for the BDF3 and the ESDIRK4 schemes. Ideally, increasing the tolerance ratio should move the curves uniformly to smaller values of work, until at a critical tolerance ratio the solution accuracy begins to deteriorate. The $\mathcal{T} = \frac{1}{20}$ and $\frac{1}{200}$ yield essentially the same accuracy levels but significantly differ in the amount of work required. For $\mathcal{T} = \frac{1}{2}$ the solution accuracy begins to degrade for both schemes. Similar results are exhibited with the other ESDIRK schemes. It is concluded that the nonlinear subiteration in TLNS3D should be converged so that the tolerance ratio $\mathcal{T} = \frac{1}{10}$ is achieved, independent of the temporal integration method used.

TEMPORAL ERROR PREDICTOR

Any modern, efficient time advancement scheme should not only be capable of estimating temporal error, it should also have an error controller that adjusts step size to maintain fixed local error. An error estimate is obtained using $\delta^{n+1} = \mathbf{U}^{n+1} - \hat{\mathbf{U}}^{n+1}$. Normalization of the $\mathbf{U}$-vector components to order unity is recommended so that the error measure over all equations may be compared. As the main and embedded schemes differ in order by one, δ^{n+1} is then an error estimate of the lower order solution. The

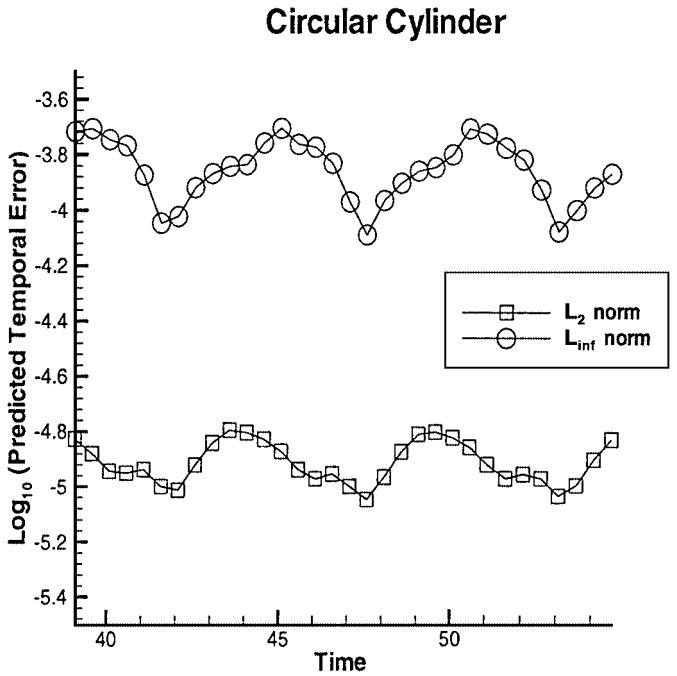

FIG. 7. Cycle variation of predicted temporal error as calculated with ESDIRK4.

ESDIRK schemes used in this study have stage-order two, and experience shows that δ^{n+1} is more reliable than with stage-order one SDIRKs. In this study it was found that there was only a minor need for step-size adjustment and hence an I-controller was deemed sufficient [7]. A *significant* additional benefit of the error estimate is that a rational stopping criterion for the nonlinear subiteration can be imposed. Specifically, the algebraic and temporal errors can be systematically balanced without having to make multiple runs.

Figure 7 shows the predicted temporal error as calculated by the ESDIRK4 scheme. The L_2 and L_∞ are presented for the coarse grid case shown in Fig. 1. A fixed time step of $\Delta t = \frac{1}{2}$ was used with a nonlinear subiteration tolerance of 0.5×10^{-5}. The temporal error correlates highly with the maximum and minimum lift. Note that about one-half an order variation in temporal error is observed over the shedding cycle. Calculations were successfully performed in this case in a variable time-stepping mode using a controller (see Kennedy and Carpenter [9] for details on controllers). The results were encouraging, but further work is needed to optimize the coupling between the ESDIRK4 scheme and the controller.

CONCLUSIONS/FUTURE WORK

The accuracy and efficiency of several time integration schemes have been investigated for the unsteady compressible Navier–Stokes equations. Time is discretized implicitly, while the spatial discretization is a conventional cell-centered finite-volume scheme with artificial dissipation added for stability. The nonlinear equations are solved at each step with a multigrid

algorithm. This study focuses on the efficiency of higher order Runge–Kutta schemes in comparison with the popular backward differentiation formulas (BDF). For this comparison, an unsteady two-dimensional laminar flow problem was chosen, i.e., flow around a circular cylinder at Re = 1200. On this problem for all realistic error levels (smaller than 10^{-1}) fourth- and fifth-order Runge–Kutta schemes are more efficient than the popular second-order backward differentiation formula (BDF2) scheme. For reasons of robustness and computer storage, the fourth-order Runge–Kutta method is recommended. The efficiency of the fourth-order Runge–Kutta scheme exceeds that of the BDF2 by a factor of 2.5 at engineering error tolerance levels (10^{-1}–10^{-2}). Efficiency gains are more dramatic at smaller tolerances.

Runge–Kutta schemes are susceptible to order reduction for stiff systems, although none was experienced for this laminar problem with stiffness levels of $\mathcal{O}(10^3)$. Future studies will investigate the convergence behavior of ESDIRK schemes on other stiff problems. The specific focus of our next study will be on the stiffness generated by popular turbulence models. It is well-known that turbulence models exhibit considerable stiffness at moderate to high Reynolds numbers (10^6). The ultimate objective will be to compare the efficiency of ESDIRK schemes with the BDF family of schemes.

A final observation is relevant, as it helps to focus future work. The TLNS3D solver technology used for the nonlinear algebraic system is nearly state of the art but *cannot* be considered efficient. For example, one time step of the implicit operator is $\mathcal{O}(10^2)$ times more expensive than one explicit time step. Significant potential improvements [O(10)] in the temporal efficiency of implicit schemes could be achieved from algebraic solver developments. One could speculate that solver improvements could be more dramatic than improvements in integration techniques. Moderate advances in both areas as well as computer hardware advances will make 3D temporal aerodynamic simulations an engineering tool in the next decade.

APPENDIX

ESDIRK3

0	0	0	0	0
$\frac{1767732205903}{2027836641118}$	$\frac{1767732205903}{4055673282236}$	$\frac{1767732205903}{4055673282236}$	0	0
$\frac{3}{5}$	$\frac{2746238789719}{10658868560708}$	$\frac{-640167445237}{6845629431997}$	$\frac{1767732205903}{4055673282236}$	0
1	$\frac{1471266399579}{7840856788654}$	$\frac{-4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$
b_i	$\frac{1471266399579}{7840856788654}$	$\frac{-4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$
$\hat{b}_i$	$\frac{2756255671327}{12835298489170}$	$\frac{-10771552573575}{22201958757719}$	$\frac{9247589265047}{10645013368117}$	$\frac{2193209047091}{5459859503100}$

ESDIRK4

0	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0
$\frac{83}{250}$	$\frac{8611}{62500}$	$\frac{-1743}{31250}$	$\frac{1}{4}$	0	0	0
$\frac{31}{50}$	$\frac{5012029}{34652500}$	$\frac{-654441}{2922500}$	$\frac{174375}{388108}$	$\frac{1}{4}$	0	0
$\frac{17}{20}$	$\frac{15267082809}{155376265600}$	$\frac{-71443401}{120774400}$	$\frac{730878875}{902184768}$	$\frac{2285395}{8070912}$	$\frac{1}{4}$	0
1	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$\frac{-2260}{8211}$	$\frac{1}{4}$
b_i	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$\frac{-2260}{8211}$	$\frac{1}{4}$
$\hat{b}_i$	$\frac{4586570599}{29645900160}$	0	$\frac{178811875}{945068544}$	$\frac{814220225}{1159782912}$	$\frac{-3700637}{11593932}$	$\frac{61727}{225920}$

ACKNOWLEDGMENTS

This work was performed while the first author was in residence at NASA Langley Research Center, Hampton, VA 23681-0001. The fourth author acknowledges support of Sandia National Laboratories, Laboratory Directed Research and Development, for support. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94-AL85000.

REFERENCES

1. K. J. Badcock, F. Cantariti, I. Hawkins, M. Woodgate, L. Dubuc, and B. E. Richards, Simulation of unsteady turbulent flows around moving aerofoils using the pseudo-time method, *Int. J. Numer. Methods Fluids* **32**(5), 585 (2000).
2. H. Bijl, M. H. Carpenter, and V. N. Vatsa, *Time Integration Schemes for the Unsteady Navier–Stokes Equations*, AIAA Paper 2001-2612 (2001).
3. G. Bosch and W. Rodi, Simulation of vortex shedding past a square cylinder with different turbulence models, *Int. J. Numer. Methods Fluids* **28**(4), 601 (1998).
4. R. Bouard and M. Coutanceau, The early stage of the wake behind an impulsively started cylinder for $40 < Re < 104$, *J. Fluid Mech.* **101**, 583 (1980).
5. J. R. Cox, K. S. Brentner, and C. L. Rumsey, Computation of vortex shedding and radiated sound for a circular cylinder: sub-critical to trans-critical Reynolds number, *Theor. Comput. Fluid Dyn.* **12**(4), 233 (1998).
6. E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations. I: Nonstiff Problems* (Springer-Verlag, Berlin, 1993), 2nd ed.
7. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations. II: Stiff and Differential-Algebraic Problems* (Springer-Verlag, Berlin, 1996), 2nd ed.
8. C. A. Kennedy, M. H. Carpenter, and R. M. Lewis, Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations, *Appl. Numer. Math.* **35**(3), 177 (2000).

9. C. A. Kennedy and M. H. Carpenter, Additive Runge–Kutta schemes for convection-diffusion-reaction equations, submitted for publication.
10. A. Kværnø, *More, and to Be Hoped for, Better DIRK Methods for the Solution of Stiff ODEs*, Technical Report (Mathematical Sciences Division, Norwegian Institute of Technology, Trondheim, Norway, 1992).
11. L. Martinelli and A. Jameson, *Validation of a Multigrid Method for the Reynolds Averaged Equations*, AIAA Paper 88-0414 (1988).
12. Y. P. Marx, Time integration for the unsteady incompressible Navier–Stokes equations, *J. Comput. Phys.* **112**, 182 (1994).
13. N. D. Melson, M. D. Sanetrik, and H. L. Atkins, Time-accurate Navier–Stokes calculations with multigrid acceleration, in *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods, Copper Mountain, Colorado, April, 1993*, edited by N. D. Melson, T. Manteuffel, and S. McCormick, NASA Conference Publication 3224 (Washington D.C.), Part 2, p. 423, 1993.
14. R. Mittal and S. Balachandar, Effect of three-dimensionality on the lift and drag of nominally two-dimensional cylinders, *Phys. Fluids* **7**(8), 1841 (1995).
15. A. Roshko, *The Flow Past a Circular Cylinder at Low Speeds*, NACA Report TN-2913 (1953).
16. R. C. Swanson and E. Turkel, *A Multistage Time-Stepping Scheme for the Navier–Stokes Equations*, AIAA Paper 85-0035 (1985).
17. S. J. Tang and N. Aubry, On the symmetry breaking instability leading to vortex shedding, *Phys. Fluids* **9**(9), 2550 (1997).
18. V. N. Vatsa and B. W. Wedan, Development of a multigrid code for 3-d Navier–Stokes equations and its application to a grid-refinement study, *Comput. Fluids* **18**(4), 391 (1990).