A Runge-Kutta-Newton-Krylov Algorithm for Fourth-Order Implicit Time Marching Applied to Unsteady Flows

S. Isono [*] and D. W. Zingg [†]
Institute for Aerospace Studies
University of Toronto
4925 Dufferin St., Toronto
M3H 5T6, Canada

Two implicit time-marching methods are investigated for accuracy and efficiency in solving the unsteady Navier-Stokes equations. The methods considered are the second-order backwards differencing formula and the fourth-order explicit-first-stage, single-diagonal-coefficient, diagonally-implicit Runge-Kutta method. First, the efficiency of two strategies for solving the nonlinear problem arising at each time step, an approximate factorization algorithm and a Newton-Krylov algorithm, is investigated. The Newton-Krylov strategy is seen to be more efficient, especially on fine meshes. Next, the relative efficiency of the two time-marching methods is compared for two-dimensional unsteady laminar flows over a cylinder and an airfoil. The backwards differencing method with approximate factorization dual time stepping is very efficient on a coarse mesh, whereas the implicit Runge-Kutta scheme combined with the Newton-Krylov algorithm is more efficient on finer meshes and when lower errors are required. The combination of the implicit Runge-Kutta method with the Newton-Krylov algorithm is shown to be very efficient for high-fidelity time-accurate simulations.

Introduction

Higher-order time-marching methods, that is higher-order methods for numerical integration of first-order ordinary differential equations in time, are pertinent in several areas, including aeroacoustics, large-eddy simulations, and direct simulations of turbulent flows. Their study is motivated by the desire to choose the right method for a specific problem and to develop new and improved methods. Analysis provides a degree of understanding.[1] For example, Zingg[2] provides a methodology for comparing explicit time-marching methods that takes into account accuracy, computational work, and the stability bound. The application of implicit methods to nonlinear problems adds an additional complication in that a nonlinear algebraic system of equations arises at each time step. The system can be linearized in time; this introduces a linearization error that restricts the method to second-order accuracy. Alternatively, the nonlinear system of equations can be solved inexactly using an iterative technique. This approach is sometimes called a subiteration technique.[3] If a pseudo-time derivative is introduced to facilitate the solution of the nonlinear system, then the approach is usually termed a dual-time-stepping technique.[4] In either case, a problem in some ways equivalent to a steady flow problem is being solved at each time step. Hence acceleration techniques for steady problems, such as local preconditioning[4] and multigrid,[5, 6] can be applied to improve the convergence rate of the subiterations.

Systematic numerical experiments can provide a thorough assessment and characterization of implicit time-marching methods for nonlinear problems. Since problems of interest in fluid dynamics typically do not have known analytical solutions, this generally involves the computation of a reference solution using a very small time step. This solution is then used to calculate the errors in solutions computed using larger time steps. Studies of this kind have been reported by Rumsey *et al.*,[7] De Rango and Zingg,[8] Bijl *et al.*,[9, 10] and Carpenter *et al.*[11] De Rango and Zingg showed that a second-order dual-time-stepping scheme is much more efficient, that is it requires far less computational effort to achieve a given level of accuracy, than a scheme without subiterations. Although the original motivation for dual time stepping was to reduce linearization and in some cases factorization errors, it is also an effective way to reduce first-order errors arising from boundary and interface conditions and a loosely coupled turbulence model.

The focus of the studies of Bijl *et al.* and Carpenter *et al.* is on higher-order implicit Runge-Kutta methods, in particular explicit-first-stage, single-diagonal-coefficient, diagonally-implicit Runge-Kutta (ESDIRK) methods of various orders. It is well known

[*] Graduate Student, sammy@oddjob.utias.utoronto.ca.

[†] Professor, Senior Canada Research Chair in Computational Aerodynamics, Senior Member AIAA, dwz@oddjob.utias.utoronto.ca.

that unconditionally-stable (A-stable) linear multistep methods are restricted to second-order accuracy.[1] Implicit multi-stage schemes do not face this restriction, and ESDIRK schemes can be of arbitrarily high order while retaining A stability (in fact, L stability[10]). In general, higher-order schemes in both space and time become more attractive as the need for accuracy increases, i.e. at low error tolerances. Consequently, the ESDIRK family of schemes, especially the fourth-order variant, are potentially an efficient alternative to second-order methods (of which the second-order backwards formula is the most popular) for problems requiring high accuracy, such as direct turbulent simulations and long-range wave propagation.

The efficiency of a time-marching method that involves subiterations depends on the iterative technique used to solve the nonlinear problem or problems arising at each time step. The degree to which the nonlinear problem is converged represents another factor affecting the efficiency. Bijl *et al.* and Carpenter *et al.* used a multigrid technique which is efficient for steady flows. In this paper, we compare the second-order backwards differencing method with the fourth-order ESDIRK method using both an approximate-factorization algorithm[3] and a Newton-Krylov algorithm[12] to solve the nonlinear problems. Our objectives are to develop efficient implementations of the two methods and then to compare their efficiency.

Governing Equations

The two-dimensional thin-layer Navier-Stokes equations are solved. In generalized curvilinear coordinates, the equations take the following form:

$$\partial_t \hat{Q} + \partial_\xi \hat{E} + \partial_\eta \hat{F} = \mathcal{R}e^{-1} \partial_\eta \hat{S} \tag{1}$$

where $\hat{Q}$ is a vector containing the conservative variables, mass, momentum and energy, per unit volume:

$$\hat{Q} = J^{-1} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix} \tag{2}$$

$\hat{E}$ and $\hat{F}$ contain the inviscid fluxes in the ξ and η directions respectively,

$$\hat{E} = J^{-1} \begin{bmatrix} \rho U \\ \rho U u + \xi_x p \\ \rho U v + \xi_y p \\ (e+p)U - \xi_t p \end{bmatrix} \tag{3}$$

$$\hat{F} = J^{-1} \begin{bmatrix} \rho V \\ \rho V u + \eta_x p \\ \rho V v + \eta_y p \\ (e+p)V - \eta_t p \end{bmatrix} \tag{4}$$

where U and V are the contravariant velocities given by:

$$\begin{aligned} U &= \xi_t + \xi_x u + \xi_y v \\ V &= \eta_t + \eta_x u + \eta_y v \end{aligned}$$

and J represents the metric Jacobian of the transformation:

$$J^{-1} = (x_\xi y_\eta - x_\eta y_\xi) \tag{5}$$

The viscous terms are:

$$\hat{S} = J^{-1} \begin{bmatrix} 0 \\ A \\ B \\ C \end{bmatrix} \tag{6}$$

where

$$\begin{aligned} A &= \eta_x m_1 + \eta_y m_2 \\ B &= \eta_x m_2 + \eta_y m_3 \\ C &= \eta_x (u m_1 + v m_3 + m_4) + \eta_y (u m_2 + v m_3 + m_5) \end{aligned}$$

with

$$\begin{aligned} m_1 &= \mu(4\eta_x u_\eta - 2\eta_y v_\eta)/3 \\ m_2 &= \mu(\eta_y u_\eta + \eta_x v_\eta) \\ m_3 &= \mu(-2\eta_x u_\eta + 4\eta_y v_\eta)/3 \\ m_4 &= \mu \mathcal{P}r^{-1} (\gamma - 1)^{-1} \eta_x \partial_\eta (a^2) \\ m_5 &= \mu \mathcal{P}r^{-1} (\gamma - 1)^{-1} \eta_y \partial_\eta (a^2) \end{aligned}$$

Pressure is related to the conservative variables by the equation of state for a perfect gas, which leads to:

$$p = (\gamma - 1)\left(e - \frac{1}{2}\rho(u^2 + v^2)\right) \tag{7}$$

All of the above variables are nondimensionalized as described by Pulliam.[13]

Second-order centered-differencing is used to discretize in space with second- and fourth-difference scalar artificial dissipation.[13, 14]

Temporal Discretization

The time-marching methods investigated are the second-order backwards difference formula (BDF2)[1] and the explicit-first-stage, single-diagonal-coefficient, diagonally-implicit Runge-Kutta scheme of fourth order (ESDIRK4).[9] The BDF2 method is given by the following nonlinear equation for $\hat{Q}^{n+1}$:

$$\frac{3\hat{Q}^{n+1} - 4\hat{Q}^n + \hat{Q}^{n-1}}{2\Delta t} - \partial_t \hat{Q}^{n+1} = G_{\mathrm{BDF2}}(\hat{Q}^{n+1}) = 0 \tag{8}$$

where

$$\partial_t \hat{Q}^{n+1} = -\partial_\xi \hat{E}^{n+1} - \partial_\eta \hat{F}^{n+1} + \mathcal{R}e^{-1} \partial_\eta \hat{S}^{n+1} \tag{9}$$

A general ESDIRK scheme of s stages is given by the following:[9]

$$\frac{\hat{Q}^k - \hat{Q}^n}{\Delta t} - \sum_{j=1}^{k} a_{kj} \partial_t(\hat{Q}^j) = G_{\mathrm{ESDIRK4}}(\hat{Q}^k) = 0 \tag{10}$$

$$\hat{Q}^{n+1} = \hat{Q}^n + \Delta t \sum_{j=1}^{s} b_j \partial_t(\hat{Q}^j)$$

0	0	0	0	0	0	0
c_2	a_{21}	a_{22}	0	0	0	0
c_3	a_{31}	a_{32}	a_{33}	0	0	0
c_4	a_{41}	a_{42}	a_{43}	a_{44}	0	0
c_5	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	0
c_6	a_{61}	a_{62}	a_{63}	a_{64}	a_{65}	a_{66}
	b_1	b_2	b_3	b_4	b_5	b_6

Table 1 Sample Butcher table

where $k = 1, \dots s$, a_{ij} and b_j are the coefficients of the scheme, and $\partial_t(\hat{Q}^j)$ is as above. Table 1 shows a Butcher table of the coefficients for a six-stage scheme. c_k indicates the point in the time interval $t + \Delta t$ which the solutions at each stage represent (i.e. the solution at stage k evaluates $Q(t + c_k \Delta t)$). The ESDIRK4 method is a six-stage scheme recommended by Bijl *et al.*,[9] and the coefficients are given in the same paper.

Equations (8) and (10) are nonlinear equations which must be solved at every time step or stage. These equations can be solved in various ways. Here we consider a dual-time stepping approach using approximate factorization and a Newton-Krylov algorithm.

Dual Time Step with Approximate Factorization

The dual time step scheme which we will use is the strategy employed by Venkateswaran and Merkle.[4] A pseudo-time derivative is added to the nonlinear problem arising at each time step. The resulting equation is

$$\frac{\partial \hat{Q}}{\partial \tau} = -G(\hat{Q}^{n+1}) = 0 \tag{11}$$

where the function G can be G_{BDF2} or G_{ESDIRK4}. Applying the implicit Euler method one obtains:

$$\frac{\hat{Q}^{p+1} - \hat{Q}^p}{\Delta \tau} = -G(\hat{Q}^{p+1}) = 0 \tag{12}$$

where $\Delta\tau$ is the pseudo-time step. With local time linearization, one obtains

$$[S + \Delta\tau \partial_\xi \hat{A}^p + \Delta\tau \partial_\eta \hat{B}^p] \Delta \hat{Q}^p = -\Delta\tau G(\hat{Q}^p) \tag{13}$$

where for BDF2

$$S = \left\{ \frac{3}{2\Delta t} + \frac{3\Delta\tau}{2\Delta t} \right\}$$

and for ESDIRK4

$$S = \left\{ \frac{1}{a_{kk}\Delta t} + \frac{\Delta\tau}{a_{kk}\Delta t} \right\}$$

$\hat{A}$ and $\hat{B}$ are the inviscid flux Jacobians, and the viscous terms have been omitted for simplicity. Applying approximate factorization and diagonalization gives

$$\begin{aligned} T_\xi[S + \Delta\tau \partial_\xi \Lambda_\xi] \hat{N}_d [S + \Delta\tau \partial_\eta \Lambda_\eta] T_\eta^{-1} \Delta \hat{Q}^p \\ = -\Delta\tau G(\hat{Q}^p) \end{aligned} \tag{14}$$

where

$$\hat{N}_d = T_\xi^{-1} S^{-1} T_\eta$$

The matrices Λ_ξ and Λ_η are diagonal matrices containing the eigenvalues of the inviscid flux Jacobians. The matrices T_ξ and T_η contain the corresponding eigenvectors. By simplifying using approximate factorization in diagonal form, the problem at each pseudo-time step requires the solution of two scalar pentadiagonal systems of equations.

Newton-Krylov method

Applying Newton's method to Eqs. (8) and (10) leads to the following linear system, which must be solved at each subiteration:

$$A^p \Delta \hat{Q}^p = -G(\hat{Q}^p) \tag{15}$$

where A is the Jacobian of G given by

$$A = \frac{\partial G}{\partial \hat{Q}}$$

This linear system is solved using the Generalized Minimal Residual method (GMRES), a Krylov-subspace method for nonsymmetric linear systems developed by Saad and Schultz.[15] An incomplete lower-upper factorization with some fill is used to precondition the system. The preconditioner is only updated when the total number of GMRES iterations performed in one Newton iteration increases from one iteration to the next. For a detailed description of the Newton-Krylov algorithm, see the work of Pueyo and Zingg.[12]

Results and Discussion

We are comparing four time-marching algorithms: BDF2 with approximate factorization (BDF-AF), BDF2 with the Newton-Krylov algorithm (BDF-NK), ESDIRK4 with approximate factorization (ESDIRK-AF), and ESDIRK4 with the Newton-Krylov algorithm (ESDIRK-NK). There are two issues to consider in comparing these algorithms. First, a mechanism must be chosen to terminate the subiterations. This can be based on a fixed number of subiterations, on a prescribed convergence tolerance, or some other criterion. Whatever criterion is selected, a parameter is introduced which must be chosen to maximize efficiency. The second issue relates to the definition of efficiency, which can be loosely defined as accuracy per unit of computational effort. More specifically, one can consider the error obtained with fixed computational effort or the computational effort required to achieve a specified level of accuracy. As we have already noted, the relative efficiency of a method can depend on the error level desired. Therefore, a range of error levels must be considered in comparing schemes, and the choice of scheme is problem dependent. Our approach will be to plot the error against the computing time required for a range of different time step values.

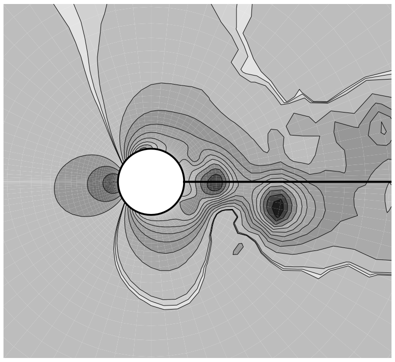

Fig. 1 Pressure contours for flow over a cylinder ($M = 0.3,\ Re = 1200$)

Two physical problems are studied: two dimensional laminar flow about a cylinder and an airfoil. Both involve self-induced periodic unsteadiness resulting from flow separation. All of the methods are applied to the airfoil problem, while only BDF-AF and ESDIRK-AF are applied to the cylinder problem. For both cases, the exact solution to the problem is not known, so a very small time step was used to obtain a reference solution using ESDIRK4. The reference solution is not free of spatial error, but the temporal error is negligible compared to that obtained with significantly larger time step values.

The initial conditions for the test cases were obtained through two stages. First, the flow was simulated from freestream conditions using a reasonably small time step for several shedding cycles until a periodic steady state was reached. The flow was then advanced using a much smaller time step, using ESDIRK4, until a new steady state was reached with the smaller time step. This solution was stored as the initial condition for simulations at larger time steps. For the error assessment, each scheme was run for one period from this initial condition. The time steps used were chosen to be multiples of the reference Δt. The error values were found by integrating the difference from the reference solution for the lift coefficient, C_l.

Test Cases

Cylinder

For the cylinder case, the free stream Mach number is 0.3 with a Reynolds number of 1200. These conditions result in the periodic shedding of vortices. The grid used is a 97×65 O-mesh with an off-wall spacing of 0.001. The boundary is at a distance of 20 diameters from the cylinder. Fig. 1 shows a snapshot of the pressure contours in the flowfield. Bijl et al.[9] also used this test case, with a different grid.

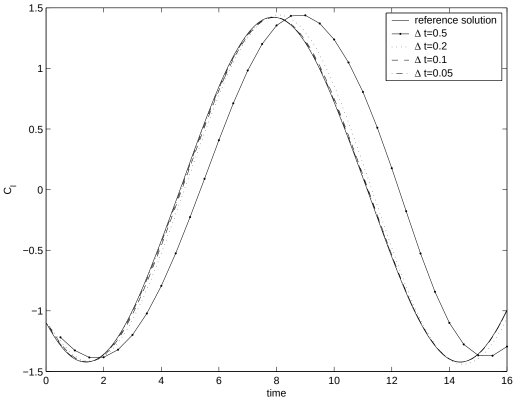

Fig. 2 C_l vs time for flow over a cylinder

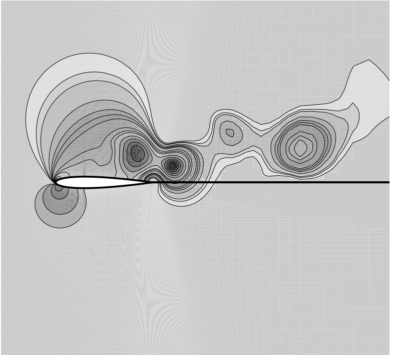

Fig. 3 Pressure contours for flow over the NACA 0012 airfoil ($M = 0.2,\ Re = 800,\ \alpha = 20°$)

The reference solution was found using a nondimensional time step of 0.01. The initial values were obtained by computing the flow for several cycles with a time step of 0.1 then computing another 20 cycles at a time step of 0.01. Since the grid is quite coarse, the solution is not particularly accurate. However, this does not affect our conclusions, since the spatial and temporal errors are independent. Hence we require that the reference solution be time-step independent, not grid independent. Figure 2 shows the lift history for one shedding cycle computed using the BDF-AF scheme with three subiterations per time step and various values of Δt.

Airfoil

The second test case involves laminar flow over the NACA 0012 airfoil, also with periodic vortex shedding. The free stream Mach number is 0.2 with a Reynolds number of 800, and the angle of attack is 20°. A snapshot of the pressure contours in the flowfield can be seen in Figure 3. This case was also studied by De Rango and Zingg.[8]

In order to examine the effect of grid density on the efficiency of the time-marching algorithms, we consider three different grids. All were generated using a hyperbolic grid generator. The first grid is a 169×49 C-mesh with 50 points on the upper and lower sur-

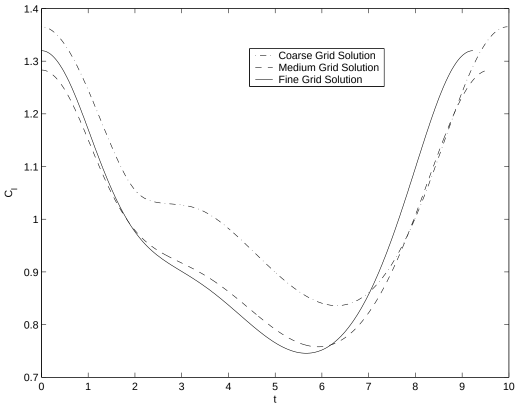

Fig. 4 Comparison of reference solutions from three grids

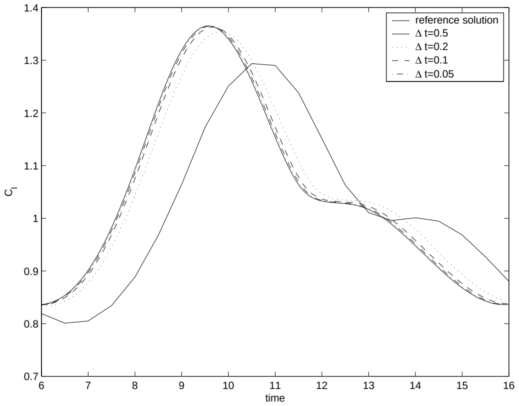

Fig. 5 C_l vs time on coarse grid

faces of the airfoil and 35 points in the wake. The off-wall spacing is 0.01 chords at the surface of the airfoil, and the distance to the outer boundary is 12 chord lengths. The minimum spacing for the streamwise clustering is 0.005 and 0.001 chords at the leading and trailing edges, respectively. The second grid is a 249×65 C-mesh with 75 points on the upper and lower surfaces and 50 points in the wake. The off-wall spacing is 10^{-4} chords at the surface of the airfoil, and the distance to the outer boundary is 12 chord lengths. The minimum spacings for the clustering at the leading and trailing edges are the same as for the coarser grid. The finest grid used is a 499×130 C-mesh with 150 points on the upper and lower surfaces and 100 points in the wake. The minimum spacing for the clustering at the leading and trailing edges is 0.0025 and 0.0005 chords. The distance to the outer boundary is 12 chord lengths with an off-wall spacing of 0.00005 chords at the surface of the airfoil. Figure 4 shows the reference solutions for one period of the vortex shedding cycle computed on the three grids. Figures 5, 6, and 7 show solutions for several values of Δt computed using the BDF-AF time-marching method with three subiterations per time step.

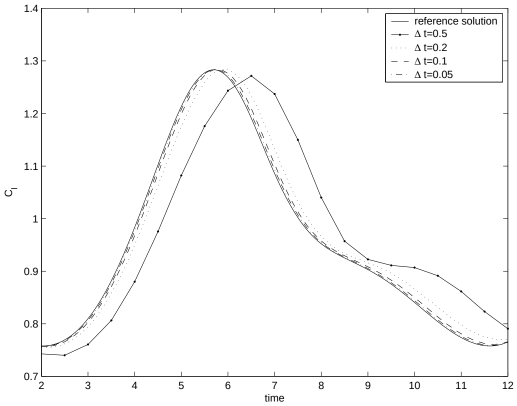

Fig. 6 C_l vs time on medium grid

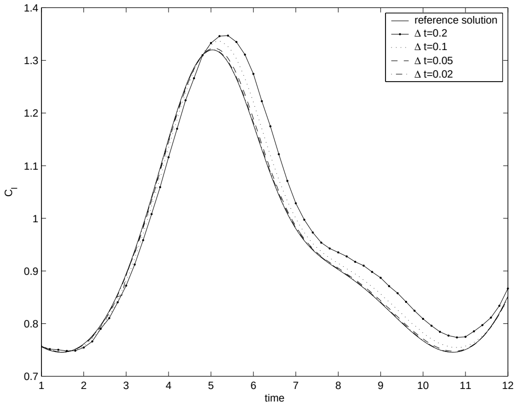

Fig. 7 C_l vs time on fine grid

Efficiency Comparisons

We seek to determine how the temporal error from the four schemes depends on the CPU time for various time step values. The processor used is an Alpha EV68 1000 MHz processor.

We begin by considering the appropriate subiteration termination criterion for each of the four algorithms. Next we compare the efficiency of the approximate factorization and Newton-Krylov algorithms for both time-marching methods, ESDIRK4 and BDF2. Finally we compare ESDIRK4 and BDF2 with optimal choices of iterative method and subiteration termination criterion.

Subiteration Termination Criteria

The efficiency of a dual-time or subiteration scheme depends on the stopping criterion for the subiterations. If the residual is not sufficiently reduced, a significant error is introduced. However, reducing the residual more than necessary does not improve the accuracy of the results and simply slows down the calculation of the solution. The subiterations can be terminated once a predetermined tolerance is reached, or a fixed number of subiterations can be chosen.

BDF-AF: For the BDF-AF time-marching scheme, De Rango and Zingg[8] have shown that using a fixed number of subiterations is effective. Figures 8 and 9

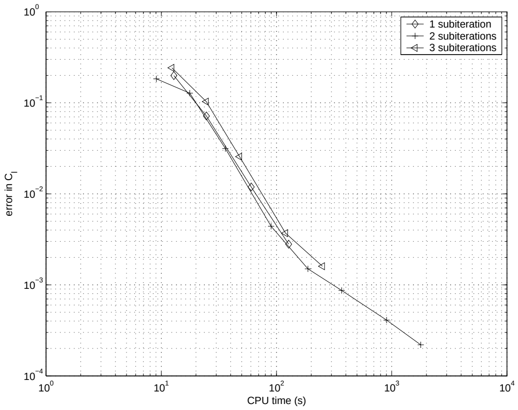

Fig. 8 Effect of number of subiterations on efficiency of BDF-AF (cylinder)

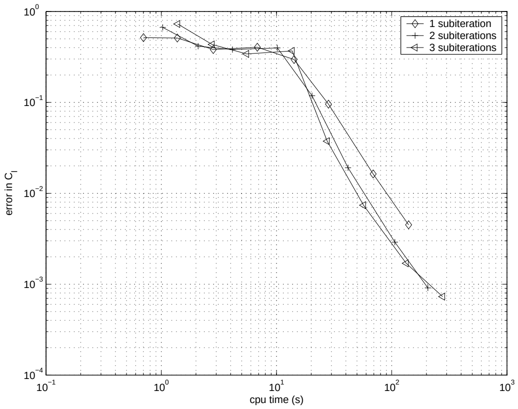

Fig. 9 Effect of number of subiterations on efficiency of BDF-AF (coarse grid)

show the error versus the cpu time required for the cylinder and coarse-grid airfoil cases, respectively, using the BDF-AF scheme with one, two, and three subiterations. Analysis of the runs on the finer grids gives similar results. Two to three subiterations are sufficient. All further results shown for the BDF-AF scheme were obtained using two subiterations per time step.

ESDIRK-AF: The ESDIRK4 scheme requires a much larger reduction in the residual than the BDF2 scheme. Consequently, we found a fixed tolerance to be more effective with the ESDIRK-AF time-marching scheme. Figures 10 and 11 show the error versus the cpu time required for the cylinder and coarse grids, respectively, with various subiteration tolerances. The results are inconclusive. In both cases, a different tolerance is optimal depending on the degree of accuracy required. For larger time steps, the residual does not need to be reduced as much. This is quite intuitive and suggests that a more sophisticated criterion could be effective. We have somewhat arbitrarily chosen a subiteration tolerance of 10^{-8} for all remaining ESDIRK-AF results. These figures also show that the ESDIRK-AF scheme is much more efficient when low errors are required. A large error reduction can be achieved with a relatively small increase in computing expense.

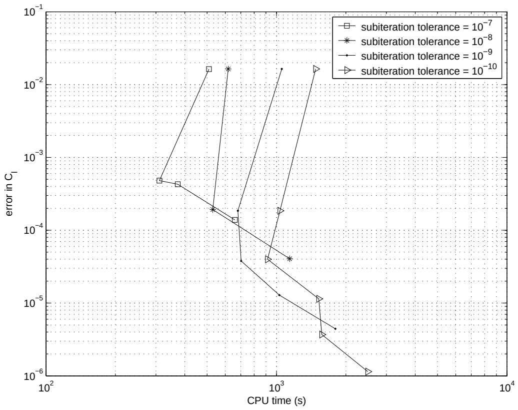

Fig. 10 Effect of subiteration tolerance on efficiency of ESDIRK-AF (cylinder)

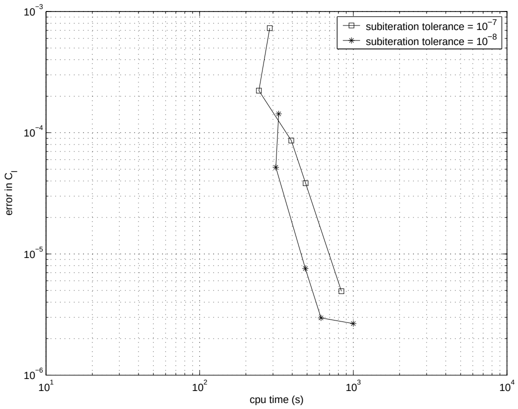

Fig. 11 Effect of subiteration tolerance on efficiency of ESDIRK-AF (coarse grid)

BDF-NK: The Newton-Krylov algorithm includes both outer Newton iterations for the nonlinear problem and inner GMRES iterations for the linear problem solved at each Newton iteration. In all cases the GMRES iterations are terminated after a reduction of the residual of the linear problem of one order of magnitude. Here we consider the termination criterion for the Newton iterations. The effect of the number of subiterations on the efficiency of the BDF-NK algorithm is shown in Figs. 12 and 13, which display the results obtained on the coarse and medium airfoil grids. With one subiteration the scheme proved to be unstable. Two subiterations are optimal.

ESDIRK-NK: Results for ESDIRK-NK can be seen in Figs. 14 and 15. As in the case of the ESDIRK-AF scheme, there is no clear choice, and we have selected four subiterations for further study. Note that this is a good choice for low error levels, but a smaller number of subiterations is optimal for higher error levels.

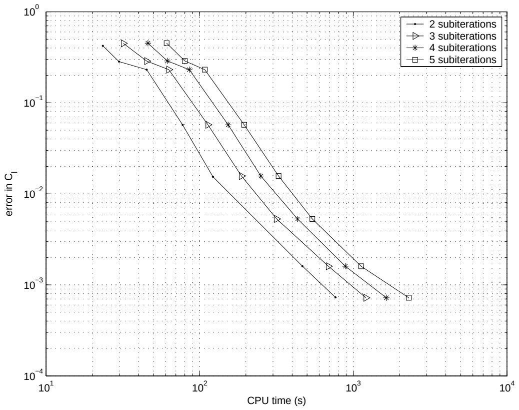

Fig. 12 Effect of number of subiterations on efficiency of BDF-NK (coarse grid)

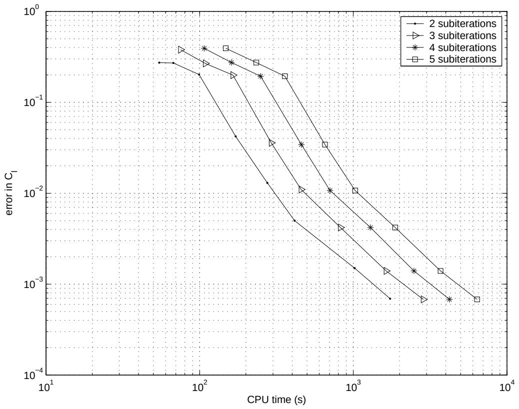

Fig. 13 Effect of number of subiterations on efficiency of BDF-NK (medium grid)

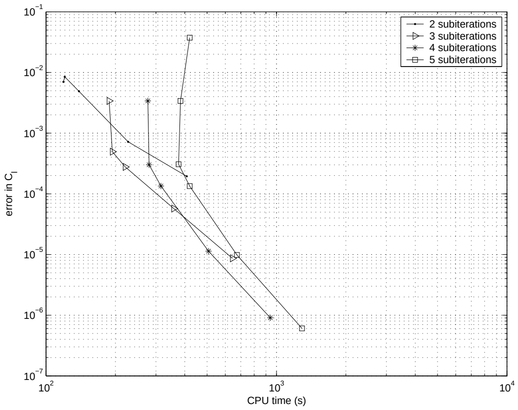

Fig. 14 Effect of number of subiterations on efficiency of ESDIRK-NK (coarse grid)

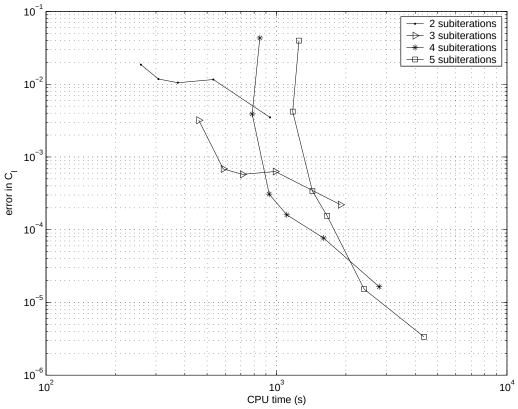

Fig. 15 Effect of number of subiterations on efficiency of ESDIRK-NK (medium grid)

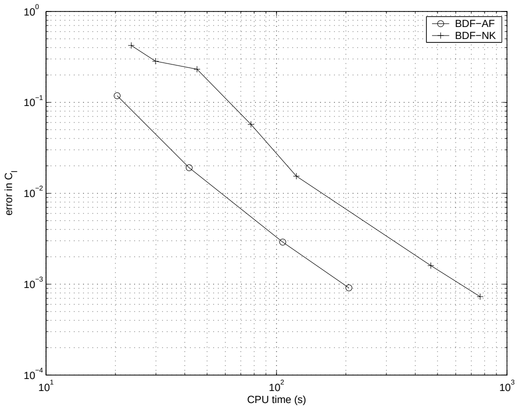

Fig. 16 Comparison of subiteration algorithms for the BDF2 time-marching method (coarse grid)

Choice of Iterative Scheme

BDF-AF vs. BDF-NK: Figures 16, 17, and 18 compare the efficiency of the BDF-AF and BDF-NK schemes for the coarse, medium, and fine grids respectively. The results are quite interesting. While BDF-AF is clearly superior on the coarse grid, BDF-NK is equally clearly superior on the fine grid. The approximate factorization algorithm does not converge as well as the Newton-Krylov algorithm on the medium and fine grids, which are more representative of practical problems. One point to note is that the approximate factorization algorithm uses a local pseudo time step that is based solely on inviscid, i.e. Courant number, considerations, which is appropriate for high Reynolds number flows. It is possible that better convergence could be obtained at these modest Reynolds numbers by incorprating the von Neumann number in the local pseudo time step definition.

ESDIRK-AF vs. ESDIRK-NK: Figures 19 and 20 compare the efficiency of the ESDIRK-AF and ESDIRK-NK schemes. On the coarse grid, the two iterative methods are equally efficient. However, on the medium grid the approximate factorization subiterations take far too long to converge, and the ESDIRK-NK is much more efficient. On the fine grid, the problem with the approximate factorization subiterations was even greater, and no reliable solutions could be obtained. Therefore, for the ESDIRK4 scheme, the Newton-Krylov algorithm is a much better choice for solving the nonlinear problem arising at each time step.

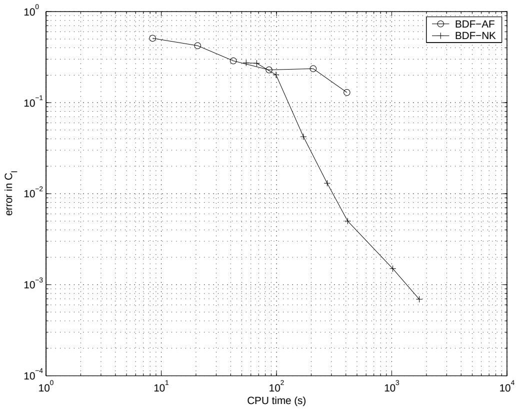

Fig. 17 Comparison of subiteration algorithms for the BDF2 time-marching method (medium grid)

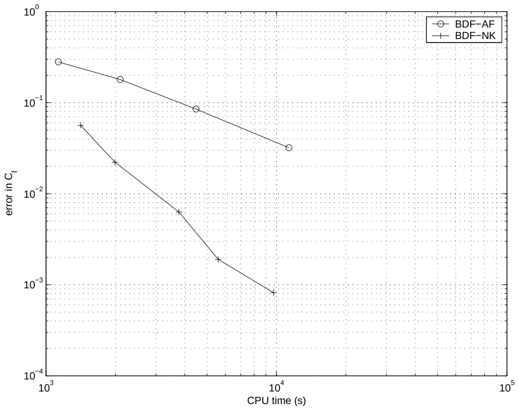

Fig. 18 Comparison of subiteration algorithms for the BDF2 time-marching method (fine grid)

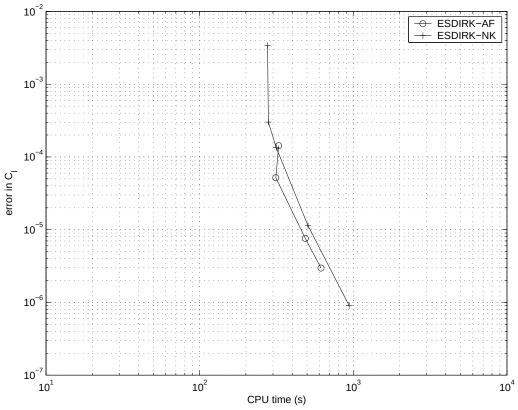

Fig. 19 Comparison of subiteration algorithms for the ESDIRK4 time-marching method (coarse grid)

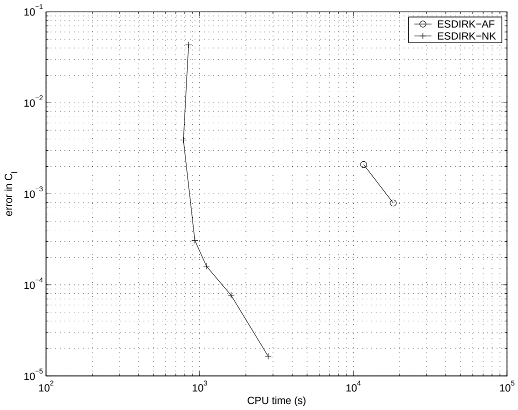

Fig. 20 Comparison of subiteration algorithms for the ESDIRK4 time-marching method (medium grid)

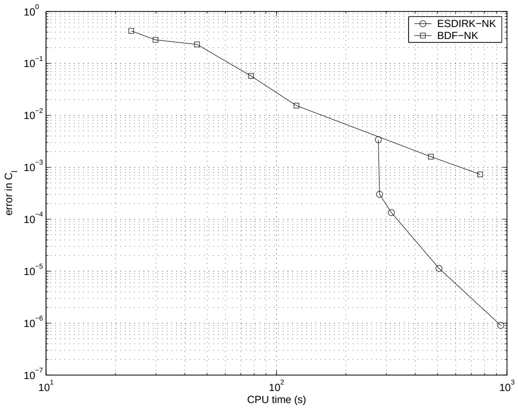

Fig. 21 Efficiency comparison of ESDIRK-NK and BDF-NK (coarse grid)

Comparison of ESDIRK4 and BDF2

Figures 21, 22, and 23 compare the efficiency of the BDF-NK and ESDIRK-NK schemes for the coarse, medium, and fine grids respectively. At higher error levels, the BDF-NK time-marching method is more efficient. For error levels below about 10^{-3}, however, the ESDIRK-NK time-marching method performs better. Therefore, for studies requiring very high accuracy in time, the ESDIRK time-marching method with Newton-Krylov subiterations is a very efficient alternative. Note that the point at which the ESDIRK-NK scheme becomes more efficient is dependent on the subiteration termination criterion.

Conclusions

The accuracy and efficiency of two time-marching methods, the second-order BDF method and the fourth-order ESDIRK method, were examined for the unsteady Navier-Stokes equations. Furthermore, two different algorithms were used to solve the nonlinear problem arising at each time step, dual time stepping combined with approximate factorization, and a Newton-Krylov algorithm.

The following conclusions can be drawn from the

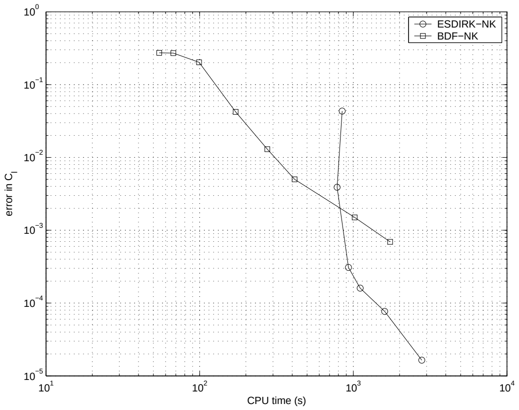

Fig. 22 Efficiency comparison of ESDIRK-NK and BDF-NK (medium grid)

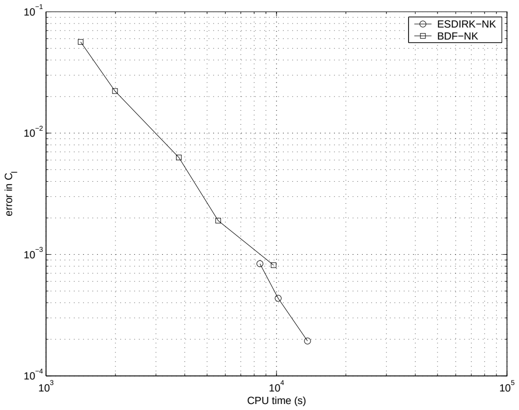

Fig. 23 Efficiency comparison of ESDIRK-NK and BDF-NK (fine grid)

results obtained:

- With BDF2, the Newton-Krylov algorithm is the more efficient on finer grids, while the approximate factorization algorithm is more efficient on coarser grids.

- With ESDIRK4, the Newton-Krylov algorithm is much more efficient than the approximate factorization algorithm.

- BDF2 is more efficient than ESDIRK4 when relatively large errors can be accepted. When lower errors are required ESDIRK4 becomes more efficient.

The main result of this paper is to show that the combination of the fourth-order implicit Runge-Kutta method with the Newton-Krylov algorithm is a very efficient option for high-fidelity time-accurate simulations. Future work can concentrate on developing subiteration termination criteria which produce more uniform efficiency for the ESDIRK-NK scheme. Another area of interest is to get a better understanding of the relative efficiency of the ESDIRK4 and BDF2 schemes for problems of current practical interest, such as aeroacoustics and large eddy simulations of turbulent flows. Although the present study shows that the ESDIRK4 scheme becomes significantly more efficient than the BDF2 scheme when very low error levels are required, it is difficult to relate these error levels to those obtained in other problems. Finally, it would be useful to compare the Newton-Krylov algorithm used here with the multigrid algorithm used by Bijl *et al.*

References

[1] Lomax, H., Pulliam, T.H., and Zingg, D.W., *Fundamentals of Computational Fluid Dynamics*, Springer, Germany, 2001.

[2] Zingg, D.W., "Comparison of High-Accuracy Finite-Difference Schemes for Linear Wave Propagation," *SIAM J. Sci. Comput.*, Vol. 22, No. 2, 2000, pp. 476-502.

[3] Pulliam, T.H., "Time Accuracy and the Use of Implicit Methods," AIAA Paper 93-3360, Orlando, 1993.

[4] Venkateswaran, S., and Merkle, C.L., "Dual Time Stepping and Preconditioning for Unsteady Computations," AIAA Paper 95-0078, Reno, Jan. 1995.

[5] Alonso, J., Martinelli, L., and Jameson, A., "Multigrid Unsteady Navier-Stokes Calculations with Aeroelastic Applications," AIAA Paper 95-0048, Jan. 1995.

[6] Arnone, A., Liou, M.-S., and Povinelli, L.A., "Integration of Navier-Stokes Equations Using Dual Time Stepping and a Multigrid Method," *AIAA J.*, Vol. 33, No. 6, 1995, pp. 985-990.

[7] Rumsey, C.L., Sanetrik, M.D., Biedron, R.T., Melson, N.D., and Parlette, E.B., "Efficiency and Accuracy of Time-Accurate Turbulent Navier-Stokes Computations," AIAA Paper 95-1835, San Diego, June 1995.

[8] De Rango, S. D., and Zingg, D. W., "Improvements to a Dual-Time-Stepping Method for Computing Unsteady Flows," *AIAA Journal*, Vol. 35, No. 9, 1997, pp. 1548-1550.

[9] Bijl, H., Carpenter, M. H., and Vatsa, V. N. "Time Integration Schemes for the Unsteady Navier-Stokes Equations," AIAA Paper 2001-2612, Anaheim, June 2001.

[10] Bijl, H., Carpenter, M. H., Vatsa, V. N., and Kennedy, C. A., "Implicit Time Integration Schemes for the Unsteady Compressible Navier-Stokes Equations: Laminar Flow," *Journal of Computational Physics*, Vol. 179, 2002, pp. 313-329.

[11] Carpenter, M. H., Viken, S. A., and Nielsen, E. J. "The Efficiency of High Order Temporal Schemes," AIAA Paper 2003-0086 Reno, January 2003.

[12] Pueyo, A., and Zingg, D.W., "Efficient Newton-Krylov Solver for Aerodynamic Computations," *AIAA J.*, Vol. 36, No. 11, 1998, pp. 1991-1997.

[13] Pulliam, T.H., "Efficient Solution Methods for the Navier-Stokes Equations," Lecture Notes for von Karman Inst. for Fluid Dynamics Lecture Series: Numerical Techniques for Viscous Flow Computation in Turbomachinery Bladings, von Karman Inst., Brussels, Jan. 1986.

[14] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping," AIAA paper 81-1259, Palo Alto, June 1981.

[15] Saad, Y., and Schultz, M. H. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, 1986, pp. 856-869.