47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition
5 - 8 January 2009, Orlando, Florida

AIAA 2009-164

Efficient Implicit Time-Marching Methods Using a Newton-Krylov Algorithm

Mohammad Tabesh* and David W. Zingg †

University of Toronto Institute for Aerospace Studies
4925 Dufferin Street, Toronto, ON, M3H 5T6, Canada

The numerical behavior of two implicit time-marching methods is investigated in solving two-dimensional unsteady compressible flows. The two methods are the second-order multistep backward differencing formula and the fourth-order multistage explicit first stage, single-diagonal coefficient, diagonally implicit Runge-Kutta scheme. A Newton-Krylov method is used to solve the nonlinear problem arising from the implicit temporal discretization. The methods are studied for two test cases: laminar flow over a cylinder and turbulent flow over a NACA0012 airfoil with a blunt trailing edge. Parameter studies show that the subiteration termination criterion plays a major role in the efficiency of time-marching methods. Efficiency studies show that when only modest global accuracy is needed, the second-order method is preferred. The fourth-order method is more efficient when high accuracy is required. The Newton-Krylov method is seen to be an efficient choice for implicit time-accurate computations.

I. Introduction

Researchers are attempting to tackle problems which were considered too ambitious just a few years ago. Multidisciplinary optimization, computational aeroacoustics in the time-domain, and large eddy and direct numerical simulations of turbulence are examples of what is being attempted today. Improvements to the efficiency of these solutions are necessary due to the complexity of such problems. There are several ways to achieve this goal. Exploiting state-of-the-art hardware is an obvious option. There are other examples: grid-adaptation techniques, parallelization of a serial code, and improved algorithms.

Higher-order methods in space and time can improve the efficiency of the algorithms by reducing grid density or time step requirements. For time-marching methods, this permits the use of larger time-steps, hence fewer steps are needed for a specific time interval. Note that this comes at the expense of increased computational cost per time-step. There is a trade-off between these two factors.

Time-marching methods can be categorized into two types: explicit and implicit. Explicit methods are easy to implement, and their computational cost per time step is low. Their stability limitation is a major difficulty, especially for stiff problems. Much bigger time-steps can be used with unconditionally stable implicit schemes, although this is achieved at the expense of increased computational cost per time-step. Higher-order implicit temporal discretization has been shown by many authors to be an effective way of improving efficiency.[1–12]

Implicit multistep methods such as Backward Differencing Formulas (BDF's) can be used to achieve higher orders of accuracy by using more values from previous time steps. Dahlquist[13] showed that the order of an A-stable linear multistep method cannot exceed two (Dahlquist's second barrier). Implicit Runge-Kutta methods offer high-accurate stable solutions to stiff problems. Following the pioneering work of Butcher[14] on implicit and semi-implicit multistage Runge-Kutta methods, Alexander[15] introduced Diagonally Implicit Runge-Kutta (DIRK) methods, and showed the benefits of this type of time-marching method. Kennedy and

*PhD Candidate, AIAA Student Member, mtabesh@oddjob.utias.utoronto.ca

†Professor and Director, Tier I Canada Research Chair in Computational Aerodynamics, Associate Fellow AIAA, http://goldfinger.utias.utoronto.ca/∼dwz/

Copyright © 2009 by D.W. Zingg and M. Tabesh. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

Carpenter[16] investigated the explicit first stage, single-diagonal coefficient, diagonally implicit Runge-Kutta (ESDIRK) schemes of various orders, studying their stability and accuracy.

The other obstacles against achieving higher accuracy in time are additional linearization and factorization errors. These simplifications are both typically second-order accurate. In order to increase the accuracy of the method, we not only need a more accurate approximation for the time derivative, but we also have to find a way to eliminate linearization and factorization errors. The subiteration or dual-time-stepping approach deals with this problem by solving a nonlinear problem at each step, avoiding linearization and factorization errors. Pulliam[1] proposed a subiterative approach to enhance time accuracy. Venkateswaran and Merkle[2] introduced a locally preconditioned dual-time-stepping algorithm for unsteady flows.

Many researchers have used these approaches to model unsteady flows. Rumsey *et al.*[3] studied the numerical efficiency of two types of subiterations. De Rango and Zingg[4] examined the efficiency of the subiterative approach applied to laminar and turbulent flows. Later, Bijl *et al.*[5] applied BDF and ESDIRK schemes of various orders to laminar flows. Jothiprasad *et al.*[6] studied the efficiency of ESDIRK methods using three different nonlinear solvers. Isono and Zingg[7] showed that the combination of higher-order time-marching methods and a Newton-Krylov solver is much more efficient that an approximate factorization solver, and concluded that a Newton-Krylov method can be very effective in modelling unsteady flows. Bijl and Carpenter[8] and Bijl[9] showed that the combination of the standard nonlinear multigrid method and a Newton-Krylov method can lead to considerable speed-up. Yang and Mavriplis[10] investigated the efficiency of several higher-order time marching methods for aeroelastic applications. Rumpfkeil and Zingg[11, 12] used a time-accurate algorithm to drive an optimization method for the optimal control of unsteady flows.

The present research extends the work of De Rango and Zingg[4] and Isono and Zingg[7] to develop an efficient subiterative method using the second-order BDF and fourth-order ESDIRK schemes coupled with a Newton-Krylov method to solve laminar and turbulent flows. The objectives can be summarized as follows:

- to develop efficient time-accurate algorithms suitable for unsteady flow simulations,
- to evaluate and optimize the efficiency of each scheme by extensive parameter studies, and
- to demonstrate the efficiency of the schemes using different test cases.

The paper is organized as follows. In Sections II and III, the governing equations and spatial discretization are reviewed. Section IV explains the different time-marching methods used. Section V focuses on the Newton-Krylov method for solving the nonlinear system. Finally, results and conclusions are presented in Sections VI and VII.

II. Governing Equations

The governing equations of interest are the compressible Navier-Stokes equations. Since this research is conducted using structured grids, a curvilinear coordinate transformation is applied to map the nonuniform grid in physical coordinates (x, y) to an evenly-spaced rectangular grid in computational coordinates $(\xi(x, y), \eta(x, y))$.[17] Also, in flow fields with high Reynolds numbers where the flow is attached or mildly separated, the viscous terms associated with derivatives along the body (ξ direction) are negligible. Highly stretched grids are used to resolve the normal gradients of the flow near the surfaces, without resolving the diffusion terms involving derivatives parallel to those surfaces. Dropping all viscous derivatives in the ξ direction in the governing equations, the two-dimensional thin-layer Navier-Stokes equations in curvilinear coordinates can be obtained:

$$\frac{\partial \hat{\mathcal{Q}}}{\partial t} + \frac{\partial \hat{\mathcal{E}}}{\partial \xi} + \frac{\partial \hat{\mathcal{F}}}{\partial \eta} - \mathcal{R}e^{-1} \left(\frac{\partial \hat{\mathcal{S}}}{\partial \eta} \right) = 0 \tag{1}$$

where $\hat{\mathcal{Q}} = J^{-1}[\rho \;\; \rho u \;\; \rho v \;\; e]^T$ are the conservative variables, $\hat{\mathcal{E}}$ and $\hat{\mathcal{F}}$ are the convective flux vectors, $\hat{\mathcal{S}}$ is the viscous flux vector, and $\mathcal{R}e$ is Reynolds number. J is the metric Jacobian of the transformation: $J^{-1} = (x_\xi y_\eta - x_\eta y_\xi)$. The flux vectors are given in Appendix A.

The effect of turbulence can be approximated by means of a turbulence model. This consists of calculating an eddy viscosity term, μ_t, and adding it to the dynamic viscosity μ. In this research we use the one-equation Spalart-Allmaras[18] turbulence model. It has been successfully used in computing a wide range of aerodynamic flows. Note that its accuracy has not been thoroughly validated for unsteady flows.

III. Spatial Discretization

By spatial discretization we convert the governing equations (1) to a semi-discretized form, i.e. from a set of nonlinear partial differential equations to a system of nonlinear ordinary differential equations (ODE's):

$$\frac{d\hat{Q}}{dt} + \hat{R}(\hat{Q}) = 0 \tag{2}$$

OPTIMA is a two-dimensional compressible flow-solver and optimization package developed by the University of Toronto Institute for Aerospace Studies Computational Aerodynamics group.[19, 20] It is capable of handling complex geometries, including multi-element airfoils. OPTIMA uses second-order centered-differencing. Second- and fourth-difference scalar artificial dissipation are used to maintain stability. The algorithm has been well validated for steady[21–24] and unsteady[4, 7] flows and has been the subject of several grid convergence studies.[25, 26]

IV. Time-Marching Methods

Time-marching methods convert the ODE's produced by the spatial discretization to a system of algebraic equations. As mentioned earlier, explicit methods fail to deal with stiffness: the time step selection has to be based on the stability limits rather than physical time scales. Implicit A-stable methods are a much better choice for solving stiff problems. A numerical method is A-stable[a] if it is stable for all ODE's that are inherently stable.[27] An even more attractive property is L-stability[b]. An L-stable method guarantees that large eigenvalues will be damped rapidly. There are two major types of implicit higher-order time-marching method: multistep and multistage. In the next two subsections, we will focus on one method of each type.

A. Multistep Methods

Multistep methods achieve high accuracy by using values from previous time-steps. However, A-stable linear multistep methods are limited to second-order, which is a significant drawback. Also, they are not self-starting, i.e. a secondary scheme is needed to generate the required past values.

The A-stable (Figure 1(a)) second-order backward differencing scheme (BDF2) is the multistep method of choice in this research since it offers L-stability, while depending only on two previous steps. Applying second-order backward differencing to the discretized equations (2) gives the following equation:

$$\mathcal{R}^n(\hat{Q}^n, \hat{Q}^{n-1}, \hat{Q}^{n-2}) = \frac{3\hat{Q}^n - 4\hat{Q}^{n-1} + \hat{Q}^{n-2}}{2\Delta t} + \hat{R}(\hat{Q}^n) = 0 \tag{3}$$

B. Multistage Methods

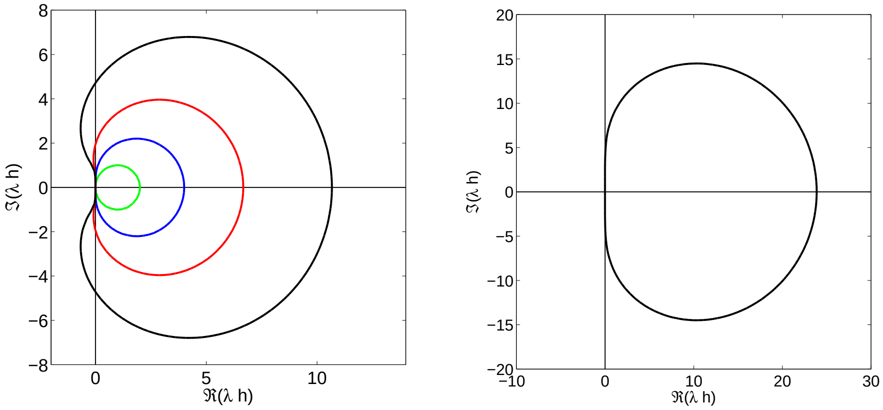
Another way to obtain higher-order accuracy is to perform multiple computations per step. The well-known explicit Runge-Kutta methods are good examples of this family. Implicit Runge-Kutta methods are not as commonly used as BDF methods in the computation of aerodynamic flows. Consider $y' = f(t, y)$; then an implicit Runge-Kutta method with s stages can be written as

$$\begin{aligned} y_{n+1} &= y_n + h\sum_{i=1}^{s} b_i f(t_{n,i}, y_{n,i}) \\ y_{n,i} &= y_n + h\sum_{j=1}^{s} a_{ij} f(t_{n,j}, y_{n,j}) \ , \end{aligned}$$

where $h = \Delta t$, $t_{n,i} = t_n + c_i h$, and $y_{n,i} = y(t_{n,i})$.

[a] The region of stability for A-stable methods contains the left-half-plane.

[b] An A-stable method with additional property $\lim_{\lambda h \to \infty} \sigma(\lambda h) = 0$ is called L-stable[28]



(a) BDF family

(b) ESDIRK4

Figure 1. Stability diagram of different time-marching methods (in each case the exterior region is stable). (a): BDF1 (green), BDF2 (blue), BDF3 (red), and BDF4 (black), (b): ESDIRK4

A Butcher tableau for a general Runge-Kutta scheme is in the form

$$\begin{array}{c|c} c_i & a_{ij} \\ \hline & b_j \end{array}$$

where a_{ij}, b_j and c_i are the coefficients of the scheme.[29]

One way of simplifying implicit Runge-Kutta methods is to use a lower triangular a_{ij} matrix. In this case, each stage is independent of the next stages. These methods are called semi-implicit Runge-Kutta schemes. Consider an s-stage method for solving a system of m equations. A fully implicit Runge-Kutta scheme requires solving an $s \times m$ system. With a semi-implicit scheme, this requirement is reduced to s-stages of m equations each. A Butcher tableau for a four-stage semi-implicit Runge-Kutta scheme is

$$\begin{array}{c|cccc} c_1 & a_{11} & 0 & 0 & 0 \\ c_2 & a_{21} & a_{22} & 0 & 0 \\ c_3 & a_{31} & a_{32} & a_{33} & 0 \\ c_4 & a_{41} & a_{42} & a_{43} & a_{44} \\ \hline & b_1 & b_2 & b_3 & b_4 \end{array}$$

Among this family, high-order L-stable ESDIRK schemes show promising properties. Since the first stage of the ESDIRK scheme is explicit, a method with s stages requires $(s-1)$ nonlinear equation solves per step. A general s-stage ESDIRK scheme is given by the following equation:

$$\mathcal{R}^n_k(\hat{Q}^n_k, \ldots, \hat{Q}^n_1, \hat{Q}^{n-1}) = \frac{\hat{Q}^n_k - \hat{Q}^{n-1}}{\Delta t} + \sum_{j=1}^{k} a_{kj} R(\hat{Q}^n_j) = 0, \qquad k = 1, \ldots, s\,. \tag{4}$$

The Butcher tableau for a four-stage ESDIRK scheme is

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ c_2 & a_{21} & \gamma & 0 & 0 \\ c_3 & a_{31} & a_{32} & \gamma & 0 \\ 1 & a_{41} & a_{42} & a_{43} & \gamma \\ \hline & a_{41} & a_{42} & a_{43} & \gamma \end{array}$$

The c_i coefficients, which represent the stage in the time interval, are not used, since $R(\hat{Q})$ is not time dependent. Also, the b_j coefficients are not needed, since in the ESDIRK schemes $c_s = 1$ and $b_j = a_{sj}$, i.e. the stiffly accurate condition, which also guarantees L-stability for A-stable (see Figure 1(b)) methods of this type.[30] The coefficients of the 6-stage 4th-order (ESDIRK4) scheme are given in Appendix B.

V. Newton-Krylov Method

A. Newton's Method

The fully discretized equations are in the form of $\mathcal{R}^n(\hat{Q}^n, \hat{Q}^{n-1}, \cdots) = 0$. A subiterative method can be used to solve this nonlinear system as a steady-state problem at each time step. We use superscript p as the subiteration index. Since these methods are iterative, a termination criterion is needed to stop the subiterations. The subiterations are terminated when the nonlinear residual is reduced by a specified order of magnitude. We examine the effect of the termination criterion on the efficiency of the subiterative method in the next section.

Applying Newton's method to equations (3) leads to the following linear system, which must be solved at each subiteration:

$$\mathcal{A}^p \Delta \hat{Q}^p = -\mathcal{R}^n(\hat{Q}^p, \hat{Q}^{n-1}, \hat{Q}^{n-2}) \,,$$

where p is the subiteration index, and $\mathcal{A}$ is the Jacobian of $\mathcal{R}$ given by

$$\mathcal{A}^p = \frac{\partial \mathcal{R}^p}{\partial \hat{Q}^p}.$$

From equation (3), the Jacobian of the BDF2 method is:

$$\mathcal{A}^p = \nabla_{\hat{Q}^p} \mathcal{R}^n(\hat{Q}^p, \hat{Q}^{n-1}, \hat{Q}^{n-2}) = \frac{3}{2\Delta t} I + \frac{\partial \hat{R}(\hat{Q}^p)}{\partial \hat{Q}^p} \,. \tag{5}$$

The Jacobian for the ESDIRK4 scheme can be derived in a similar way. Consider the implicit stages

$$\mathcal{R}_k^n(\hat{Q}_k^n, Q_{k-1}^n, \ldots, \hat{Q}_1^n, \hat{Q}^{n-1}) = \frac{\hat{Q}_k^n - \hat{Q}^{n-1}}{a_{kk}\Delta t} + \frac{1}{a_{kk}} \sum_{j=1}^{k} a_{kj} \hat{R}(\hat{Q}_j^n) = 0, \qquad k = 2, .., s \,.$$

For each stage, the Jacobian is:

$$\mathcal{A}_k^p = \nabla_{\hat{Q}_k^p} \mathcal{R}^n(\hat{Q}_k^p, \hat{Q}_{k-1}^n, \ldots, \hat{Q}_1^n, \hat{Q}^{n-1}) = \frac{1}{a_{kk}\Delta t} I + \frac{\partial R(\hat{Q}_k^p)}{\partial \hat{Q}_k^p}, \qquad k = 2, .., s \,. \tag{6}$$

B. Solving the Linear System

The linear system is solved using the generalized minimal residual method (GMRES), a Krylov-subspace method for non-symmetric linear systems developed by Saad and Schultz.[31] A restarted version of GMRES is used, denoted by GMRES(m), where m is the number of search directions. The restart index, m, was set to 40. We found that restarting was not necessary for the cases studied in this research. The residual of the linear system was reduced by one order of magnitude for laminar flows and three orders of magnitude for turbulent flows.

The convergence of the Krylov method strongly depends on the conditioning of the system. Preconditioners can be used to improve the efficiency of the GMRES algorithm by clustering the eigenvalues of the system compared to the original system. In this research, a right-preconditioner in the form of

$$\mathcal{A}^p \mathcal{M}^{-1} \mathcal{M} \Delta Q^p = -\mathcal{R}^p$$

is used, where $\mathcal{M}$ represents the preconditioner matrix.

An incomplete lower-upper factorization with a fill level k (ILU(k)) of an approximate Jacobian matrix is used to form the preconditioner. Factorizations with higher fill levels are better approximations of the approximate Jacobian, but they increase the computational cost and memory usage. Following Pueyo and Zingg,[22] the approximate Jacobian is formed using a combination of second- and fourth-difference dissipation:

$$d_l^{(2)} = d_r^{(2)} + \sigma d_r^{(4)} \,, \tag{7}$$

where $d_l^{(2)}$ is the dissipation coefficient in the approximate Jacobian, and $d_r^{(2)}$ and $d_r^{(4)}$ are dissipation coefficients on the right-hand-side. The parameter σ has an optimal value of 4 to 6. Boundary entries are scaled and pivoted since the boundary entries can be orders of magnitude different from the interior entries. Following Pueyo and Zingg,[22] reverse Cuthill-McKee reordering is used to improve the performance of the preconditioner.

Since GMRES only needs matrix-vector products in the form of $\mathcal{A}^p\mathbf{v}$, where $\mathbf{v}$ is a vector, there is no need to form $\mathcal{A}^p$, and the product can be approximated by a forward-difference approximation. For example for the BDF2 scheme the following approximation can be used:

$$\mathcal{A}^p\mathbf{v} \approx \frac{\mathcal{R}^n(\hat{Q}^p + \epsilon\mathbf{v}, \hat{Q}^{n-1}, \hat{Q}^{n-2}) - \mathcal{R}^n(\hat{Q}^p, \hat{Q}^{n-1}, \hat{Q}^{n-2})}{\epsilon}, \tag{8}$$

where $\epsilon = \sqrt{\epsilon_m} \,/\, ||\mathbf{v}||_2$, and $\epsilon_m = 2.2 \times 10^{-16}$ is the approximate value of machine zero. For more details on the Jacobian-free approach see Knoll and Keyes.[32]

In steady-state simulations, a globalization phase is needed to provide a better initial iterate for Newton's method.[32] However, in unsteady flow calculations, since the initial guess is already close to the solution, globalization is not needed (at least for the cases studied here).

For more details on the Newton-Krylov method used see Pueyo and Zingg[22] and Chisholm.[33]

VI. Results and Discussion

Two unsteady cases are studied: laminar flow over a cylinder, and turbulent flow over a NACA0012 airfoil with a blunt trailing edge. In each case, the flow is unsteady and periodic. We recognize that the thin-layer approximation introduces some error for both flows studied here, but this does not affect our conclusions with respect to temporal accuracy and efficiency. All cases were run on a single Itanium 2 processor with a clock speed of 1500 MHz.

In order to find the initial condition, the flow was simulated from freestream conditions using a relatively large time step for several shedding cycles. The flow was then advanced using a very small time step until a periodic state was reached. This solution was then stored as the initial condition for the study of accuracy and efficiency of the time-marching methods. Note that since BDF2 scheme is not self-starting, the ESDIRK4 scheme was used to advance the first step in all cases.

To obtain a reference solution, a very small time step with the ESDIRK4 scheme is used. The reference solution still contains spatial error due to spatial discretization, but the temporal error is negligible since the time step is sufficiently small. This solution can be used as a reference to compute the temporal error of different time-marching methods. The error in C_l is defined as

$$C_{l\text{error}} = \sqrt{\frac{\displaystyle\sum_{i=1}^{N}(C_{l_i} - C_{l_i,\text{ref}})^2}{N}} \tag{9}$$

where N is the number of time-steps.

We define efficiency as accuracy per unit of computational effort. The subiteration termination criterion has a significant impact on the efficiency of the algorithm. The subiterations have to be converged sufficiently; otherwise the error at each step will be significant. On the other hand, reducing the residual too much will lead to oversolving the system. There are two ways to control the subiterations: a fixed number of subiterations, or a convergence tolerance. Here we use a relative convergence tolerance, which dictates how many orders of magnitude the algorithm reduces the residual at each step/stage.

A. Laminar flow over a cylinder

Two-dimensional laminar vertex shedding from a circular cylinder is studied. The Mach number and Reynolds number are 0.2 and 100 respectively. Under these conditions, the flow is unsteady and two-dimensional.[34] Grids with different densities were used to understand the dependencies of parameters on grid size. The grids are O-type structured meshes. Table 1 provides the grid specifications for the three different grids. Figures 2 and 3 show a snapshot of pressure contours and the variation of lift and drag coefficients for the reference solution.

Accuracy verification: Numerical experiments were conducted to verify the order of accuracy of each time-marching method. $C_{l\,\text{error}}$, defined by equation (9), was computed for different time-step sizes. Figure 4

Table 1. Grid data for cylinder case

grid name	grid size	grid dimensions	off-wall spacing	far-field distance
coarse	6363	101 × 63	4×10^{-5}	20 diameters
medium	25125	201 × 125	2×10^{-5}	20 diameters
fine	99849	401 × 249	1×10^{-5}	20 diameters

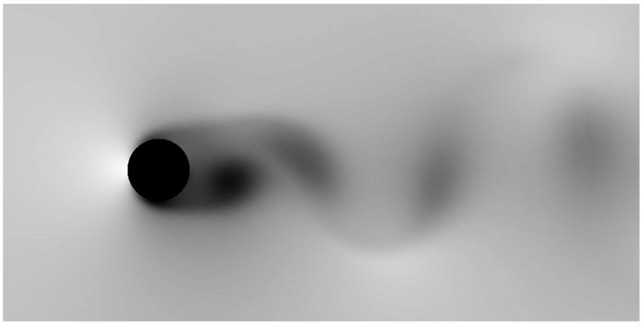


Figure 2. Snapshot of pressure contours, flow over a cylinder

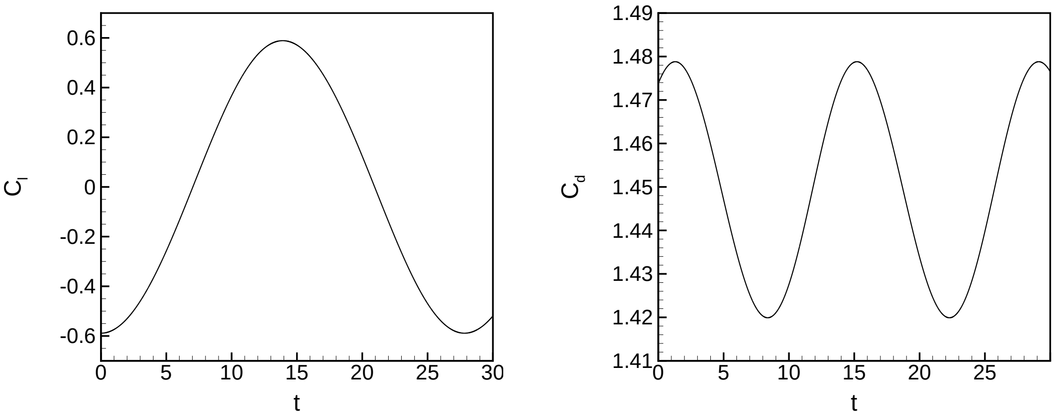


Figure 3. Variation of lift and drag coefficients, flow over a cylinder

confirms the order of accuracy of the two methods. Note that a tight tolerance was used to achieve these orders of accuracy. Using looser tolerances is more efficient and will be discussed later.

Efficiency of the time-marching methods: Plotting the $C_{l\,\mathrm{error}}$ against the required computational time for a complete cycle for different time-step sizes will help us to quantify the efficiency of the methods used, since it represents the relationship between accuracy and cost. Figures 5 through 9 show this relationship for a range of different time-step values. For each curve, the numbers of time steps per period used for the BDF2 scheme are 10, 20, 50, 100 and 500, respectively. The numbers of time-steps per period used for the ESDIRK scheme are 5, 10, 20, 50 and 100, respectively. As the number of time steps per period is increased, the error is reduced and the cost per period increased.

Figure 5 shows the effect of the subiteration termination tolerance on the efficiency of the schemes. Note that this tolerance is relative; in other words, the subiterations are terminated when the residual of the nonlinear problem is reduced by the factor specified. Numerical experiments show that the optimum relative tolerance is 4 orders of residual reduction for the BDF2 scheme and 6 orders of magnitude for the ESDIRK4

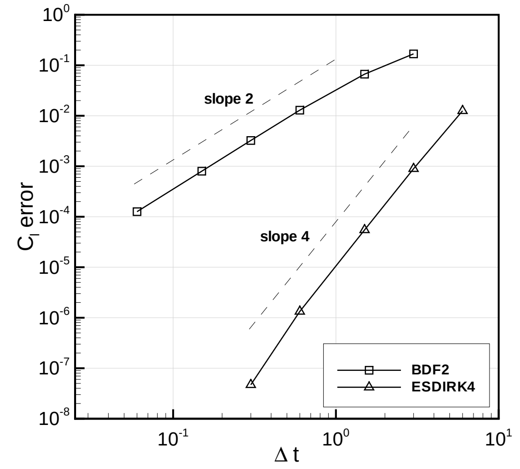
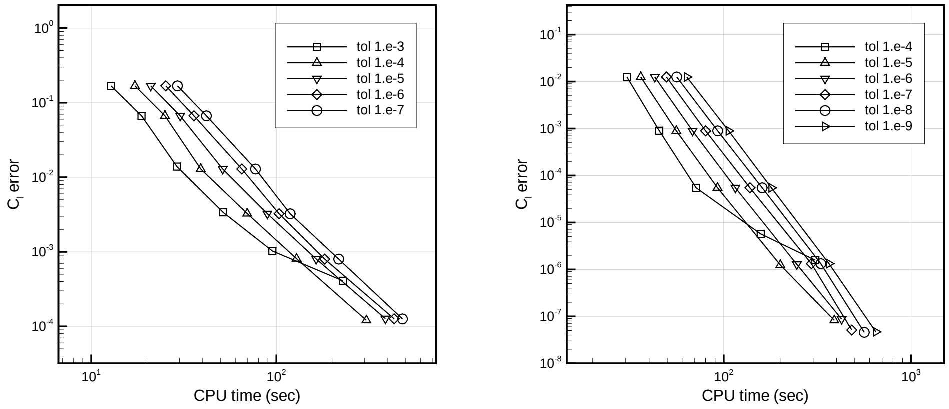


Figure 4. Accuracy diagram for BDF2 and ESDIRK4 time-marching methods, flow over a cylinder



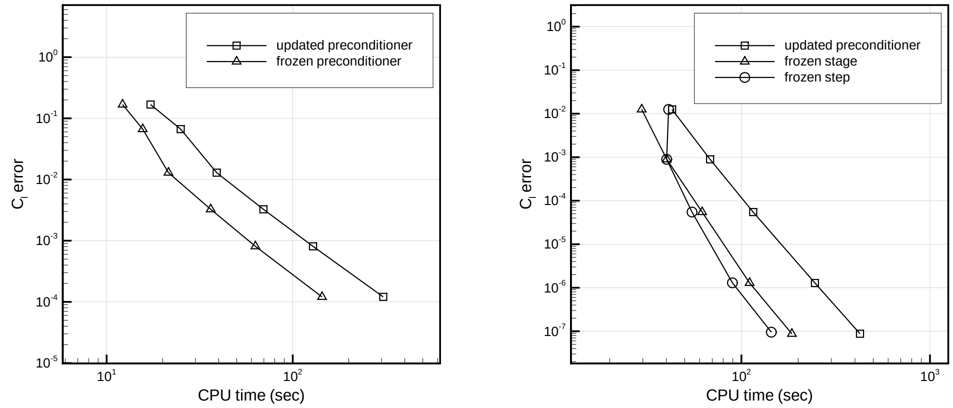
(a) BDF2

(b) ESDIRK4

Figure 5. Effect of subiteration termination tolerance, flow over a cylinder

scheme. Values presented in Figure 5 are computed on the coarse grid. The results on the medium and fine grids confirm the same optimum tolerances.

Since the initial guess at the beginning of each nonlinear step is close to the final converged solution, the preconditioner can be updated at the beginning of each subiteration and kept frozen until the converged solution is obtained. For the BDF2 scheme, this means updating the preconditioner at each time step. For the ESDIRK4 scheme, however, there are two ways of doing this. One method is to update the preconditioner at the beginning of each stage (every nonlinear problem). The other way to treat the ESDIRK4 scheme is to keep the preconditioner for the entire time step (frozen for all stages). Note that the first term in the ESDIRK4 Jacobian equation (6), $I/(a_{kk}\Delta t)$, does not change for different stages ($a_{kk} = \gamma$ is constant for all stages). For small time steps, this term is large, and the variation of the second term is small. Therefore the preconditioner can be stored and used for subsequent stages. However, for large time steps, the constant term is small and the variation of $\partial R(\hat{Q}_k^p)/\partial \hat{Q}_k^p$ is significant, so it is beneficial to update the preconditioner for each stage. Figure 6 shows the effect of freezing the preconditioner on the coarse grid for both schemes. Figure 6(b) also compares the effect of freezing the preconditioner for the entire step compared to updating it for each stage. Updating the preconditioner at each stage leads to a small increase in CPU time, but is a more robust strategy. Figure 7 compares the effect of freezing the preconditioner on both schemes. Freezing the preconditioner is beneficial for both time-marching methods.



(a) BDF2

(b) ESDIRK4

Figure 6. Effect of freezing the preconditioner, flow over a cylinder

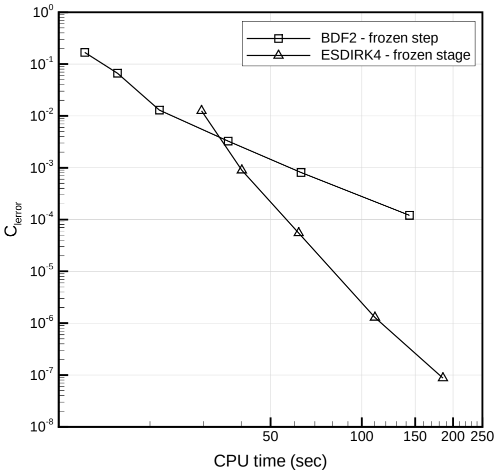
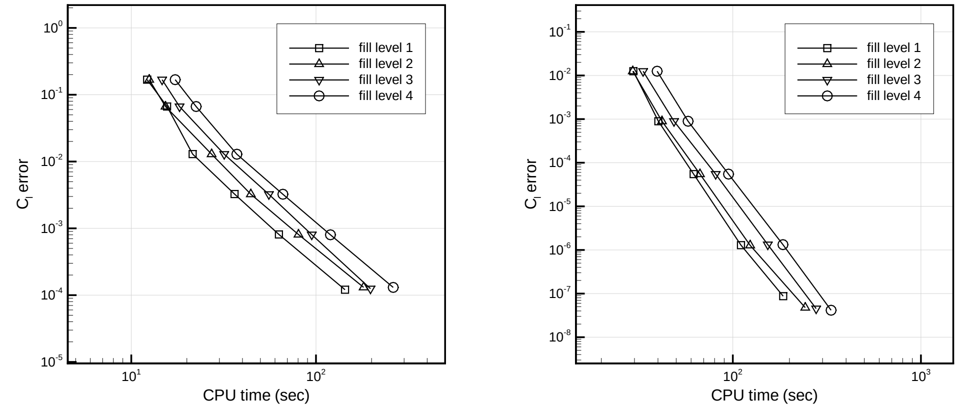


Figure 7. Comparison between BDF2 and ESDIRK4 schemes with frozen preconditioner, flow over a cylinder

Figure 8 shows the effect of the preconditioner fill level. Smaller fill levels are beneficial in terms of memory and efficiency. Fill level 1 seems to be the best choice for both schemes.

Figures 9 shows the results for all three grids using optimal parameters. The time-marching schemes show similar relative performance on all three grids. In each case, there is a point where both methods have the same efficiency, i.e. the same error level and the same cost. At this point, ESDIRK4 uses a much larger time-step. If the acceptable $C_{l\,\text{error}}$ is higher than this level, then the BDF2 scheme is more efficient. When a lower $C_{l\,\text{error}}$ is needed, the ESDIRK4 scheme is the better choice.

Newton-Krylov performance: Tables 2 and 3 demonstrate the performance of the Newton-Krylov method used with the BDF2 and ESDIRK4 schemes on the coarse grid. The data show that the average number of subiterations per time step for BDF2 and per stage for ESDIRK4 decreases as the time step is reduced. This occurs because the initial guess is improved. Moreover, the number of GMRES iterations per subiteration decreases as the time step is reduced. This occurs because the diagonal term in the Jacobian associated with the time step has a favorable effect on the conditioning of the linear system. As a result of these two effects, the cost per time step decreases substantially as the time step is reduced. Overall, the Newton-Krylov approach is seen to be a very efficient approach to such unsteady problems, as the cost of solving the associated nonlinear problems is much lower than the cost of solving a steady-state problem.



(a) BDF2 (b) ESDIRK4

Figure 8. Effect of preconditioner fill level, flow over a cylinder

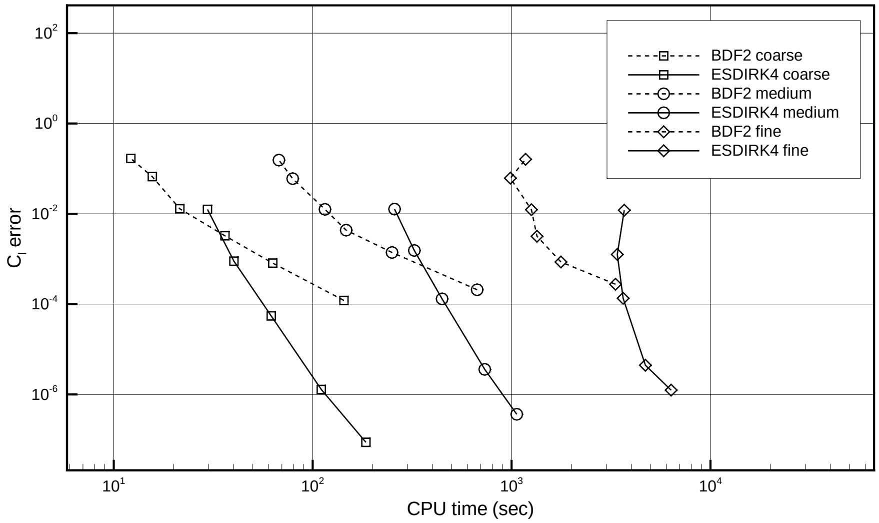


Figure 9. Comparison between time-marching methods on three different grids, flow over a cylinder

Table 2. Newton-Krylov performance data for BDF2, flow over a cylinder

Number of time steps per period	Total number of subiterations per period	Total number of GMRES iterations per period	Average number of subiterations per time step	Average number of GMRES iterations per time step	Average number of GMRES iterations per subiteration
10	82	714	8.20	71.40	8.71
20	139	867	6.95	43.35	6.24
50	274	1022	5.48	20.44	3.73
100	523	1608	5.23	16.08	3.07
200	1020	2584	5.10	12.92	2.53
500	2520	5537	5.04	11.07	2.20

Table 3. **Newton-Krylov performance data for ESDIRK4, flow over a cylinder**

Number of time steps per period	Total number of subiterations per period	Total number of GMRES iterations per period	Average number of subiterations per time step	Average number of GMRES iterations per time step	Average number of GMRES iterations per subiteration	Average number of subiterations per stage	Average number of GMRES iterations per stage
5	192	1874	38.40	374.80	9.76	7.68	74.96
10	359	2427	35.90	242.70	6.76	7.18	48.54
20	703	3535	35.15	176.75	5.03	7.03	35.35
50	1753	5608	35.06	112.16	3.20	7.01	22.43
100	3223	8689	32.23	86.89	2.70	6.45	17.38

Table 4. **Grid data for NACA0012 case**

Block Number	$jmax$ (ξ-direction)	$kmax$ (η-direction)	Number of Nodes
1	97	73	7081
2	201	73	14673
3	97	73	7081
4	97	73	7081
total	-	-	35916

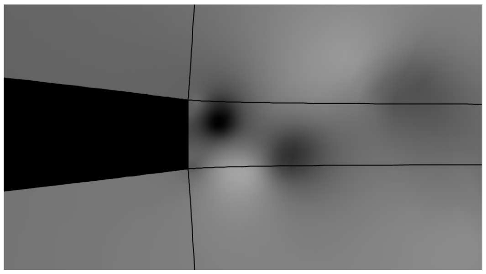


Figure 10. **Snapshot of pressure contours, flow over a NACA0012 airfoil with a blunt trailing edge**

B. Turbulent flow over a NACA0012 airfoil with a blunt trailing edge

Two-dimensional subsonic turbulent flow about a NACA0012 airfoil with a blunt trailing edge is studied. The angle of attack is 0°, and the Mach number and Reynolds number are 0.2 and 2 million respectively. The bluntness at trailing edge is 3% of the chord length. A C-type grid is used with an extra block for the blunt trailing edge. The off-wall spacing is 2.5×10^{-6} chords. Table 4 is a summary of the number of grid nodes in each block and the entire mesh. Figures 10 and 11 show a snapshot of pressure contours and the variation of lift and drag coefficients for the reference solution respectively.

Accuracy verification: Figure 12 shows the accuracy diagram for this case. Again, both methods show good agreement with the expected order of accuracy.

Efficiency of the time-marching methods: Figure 13 displays the efficiency of the time-marching methods for this case. In both cases, a frozen preconditioner was used (at each stage for ESDIRK4). No deterioration in the linear convergence was observed. This is consistent with the findings for laminar flow. Also, a fill level of one was sufficient for both schemes.

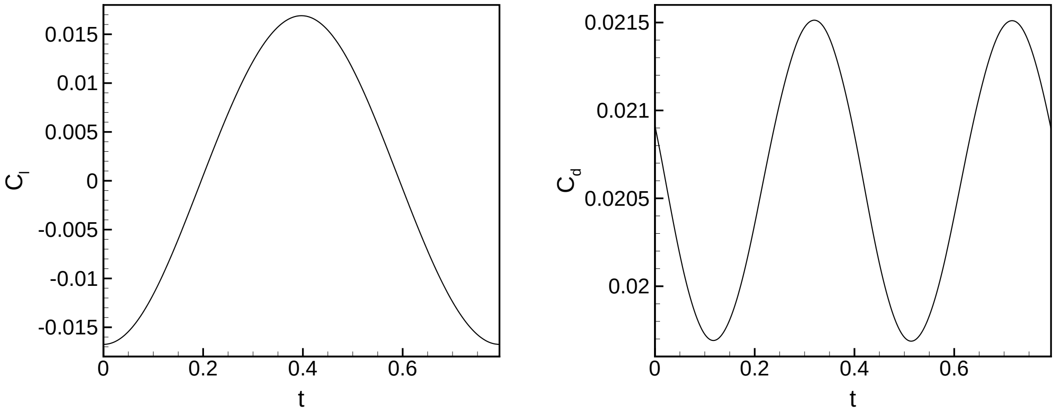


Figure 11. Variation of lift and drag coefficients, flow over a NACA0012 airfoil with a blunt trailing edge

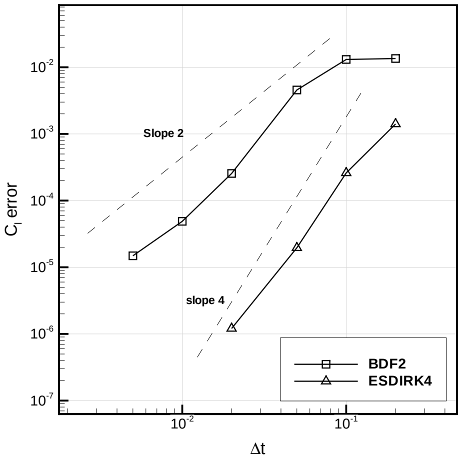


Figure 12. Accuracy diagram for BDF2 and ESDIRK4, flow over a NACA0012 airfoil with a blunt trailing edge

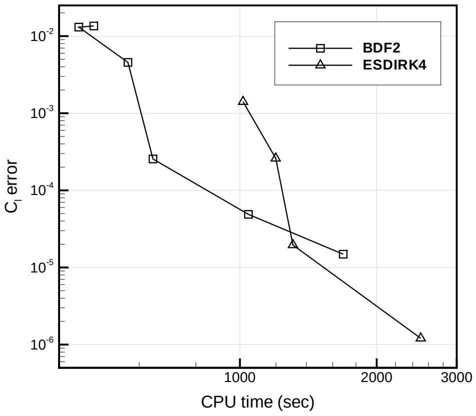


Figure 13. Comparison between time-marching methods, flow over a NACA0012 airfoil with a blunt trailing edge

VII. Conclusions

The accuracy and efficiency of two time-marching methods, BDF2 and ESDIRK4, solved using a Newton-Krylov algorithm, were investigated. Results show that BDF2 is more efficient for modest error levels; ESDIRK4 is more efficient for very accurate computations. In addition, our results show that the Newton-Krylov method is a very efficient algorithm for unsteady flow computations.

We also draw the following specific conclusions:

- The termination tolerance has a large impact on the efficiency of the scheme. Four and six order residual reduction is optimal for the BDF2 and ESDIRK4 schemes respectively.
- In unsteady computations, the initial guess at each time step is close to the converged solution, so the Newton start-up phase is not needed.
- Freezing the preconditioner is beneficial for both time-marching methods, and is recommended for unsteady computations.
- The preconditioner fill level of one is optimal for both methods in most cases.

Appendix

A. Inviscid and Viscous Fluxes

The inviscid flux vectors are:

$$\hat{\mathcal{E}} = J^{-1}\begin{bmatrix} \rho U \\ \rho U u + \xi_x p \\ \rho U v + \xi_y p \\ (e+p)U \end{bmatrix}, \quad \hat{\mathcal{F}} = J^{-1}\begin{bmatrix} \rho V \\ \rho V u + \eta_x p \\ \rho V v + \eta_y p \\ (e+p)V \end{bmatrix}$$

where u and v are x and y velocity components, ξ_x, ξ_y, η_x, and η_y are grid metrics, J is the grid Jacobian, and U and V are the contravariant velocities:

$$U = \xi_x u + \xi_y v\,, \qquad V = \eta_x u + \eta_y v\,.$$

For the thin-layer Navier-Stokes equations, the viscous flux vector is:

$$\hat{\mathcal{S}} = J^{-1}\begin{bmatrix} 0 \\ \eta_x m_1 + \eta_y m_2 \\ \eta_x m_2 + \eta_y m_3 \\ \eta_x(u m_1 + v m_2 + m_4) + \eta_y(u m_2 + v m_3 + m_5) \end{bmatrix}$$

with

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

where a is speed of sound, γ is the specific heat ratios, and $\mathcal{P}r$ and $\mathcal{P}r_t$ are laminar and turbulent Prandtl numbers.

B. ESDIRK4 Coefficients

Butcher table for the six-stage fourth-order ESDIRK4 scheme:

$$
\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 \\
\frac{83}{250} & \frac{8611}{62500} & \frac{-1743}{31250} & \frac{1}{4} & 0 & 0 & 0 \\
\frac{31}{50} & \frac{5012029}{34652500} & \frac{-654441}{2922500} & \frac{174375}{388108} & \frac{1}{4} & 0 & 0 \\
\frac{17}{20} & \frac{15267082809}{155376265600} & \frac{-71443401}{120774400} & \frac{730878875}{902184768} & \frac{2285395}{8070912} & \frac{1}{4} & 0 \\
1 & \frac{82889}{524892} & 0 & \frac{15625}{83664} & \frac{69875}{102672} & \frac{-2260}{8211} & \frac{1}{4} \\
\hline
 & \frac{82889}{524892} & 0 & \frac{15625}{83664} & \frac{69875}{102672} & \frac{-2260}{8211} & \frac{1}{4}
\end{array}
$$

References

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

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

[3] 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, 1995.

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

[5] Bijl, H., Carpenter, M. H., Vasta, 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.

[6] Jothiprasad, G., Mavriplis, D. J., and Caughey, D. A., "Higher-Order Time Integration Schemes for the Unsteady Navier-Stokes Equations on Unstructured Meshes," *Journal of Computational Physics*, Vol. 191, 2003, pp. 542–566.

[7] Isono, S. and Zingg, D. W., "A Runge-Kutta-Newton-Krylov Algorithm for Fourth-Order Implicit Time Marching Applied to Unsteady Flows," AIAA Paper 2004-0433.

[8] Bijl, H. and Carpenter, M. H., "Iterative Solution Techniques for Unsteady Flow Computations Using Higher Order Time Integration Schemes," *International Journal of Numerical Methods in Fluids*, Vol. 47, 2005, pp. 857–862.

[9] Bijl, H., "Iterative Methods for Unsteady Flow Computations Using Implicit Runge-Kutta Integration Schemes," AIAA Paper 2006-1278.

[10] Yang, Z. and Mavriplis, D. J., "Higher-Order Time-Integration Schemes for Aeroelastic Applications on Unstructured Meshes," *AIAA Journal*, Vol. 45, No. 1, 2007, pp. 138–150.

[11] Rumpfkeil, M. P. and Zingg, D. W., "A General Framework for the Optimal Control of Unsteady Flows with Applications," AIAA Paper 2007-1128.

[12] Rumpfkeil, M. P. and Zingg, D. W., "The Optimal Control of Unsteady Flows with a Discrete Adjoint Method," *Optimization and Engineering*, doi:10.1007/s11081-008-9035-5, 2008.

[13] Dahlquist, G. G., "A Special Stability Problem for Linear Multistep Methods," *BIT*, Vol. 3, 1963, pp. 27–43.

[14] Butcher, J. C., "Implicit Runge-Kutta Processes," *Mathematics of Computation*, Vol. 18, No. 85, 1964, pp. 50–64.

[15] Alexander, R., "Diagonally Implicit Runge-Kutta Methods for Stiff O.D.E.'s," *SIAM Journal on Numerical Analysis*, Vol. 14, No. 6, 1977, pp. 1006–1021.

[16] Kennedy, C. A. and Carpenter, M. H., "Addative Runge-Kutta Schemes for Convection-Diffusion-Reaction Equations," *Applied Numerical Mathematics*, Vol. 44, 2003, pp. 139–181.

[17] Pulliam, T. H., *Efficient Solution Methods for the Navier-Stokes Equations*, Lecture Notes for the Von Karman Institute For Fluid Dynamics Lecture Series, 1986.

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

[19] Nemec, M. and Zingg, D., "A Newton-Krylov Algorithm for Aerodynamic Design Using the Navier-Stokes Equations," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1146–1154.

[20] Nemec, M. and Zingg, D., "Multipoint and Multi-Objective Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 42, No. 6, 2004, pp. 1057–1065.

[21] Frew, K. and Zingg, D. W., "On Artificial Dissipation Models for Viscous Airfoil Computations," AIAA Paper 96-1970, 1996.

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

[23] Nemec, M. and Zingg, D. W., "Aerodynamic Computations Using the Convective-Upstream Split-Pressure Scheme with Local Preconditioning," *AIAA Journal*, Vol. 38, No. 3, 2000, pp. 402–410.

[24] De Rango, S. and Zingg, D. W., "Higher-Order Spatial Discretization for Turbulent Aerodynamic Computations," *AIAA Journal*, Vol. 39, No. 7, 2001, pp. 1296–1304.

[25] Zingg, D. W., "Grid Studies for Thin-Layer Navier-Stokes Computations of Airfoil Flowfields," *AIAA Journal*, Vol. 30, No. 10, 1992, pp. 2561–2564.

[26] Zingg, D. W., De Rango, S., Nemec, M., and Pulliam, T. H., "Comparison of Several Spatial Discretizations for the Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 160, 2000, pp. 683–704.

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

[28] Hairer, E. and Wanner, G., *Solving Ordinary Differential Equations. II: Stiff and Differential-Algebraic Problems*, Springer-Verlag, 1996.

[29] Butcher, J. C., *Numerical Methods for Ordinary Differential Equations*, John Wiley & Sons, 2003.

[30] Prothero, A. and Robinson, A., "On the Stability and Accuracy of One-Step Methods for Solving Stiff System of Ordinary Differential Equations," *Mathematics of Computation*, Vol. 28, No. 125, pp. 145–162.

[31] Saad, Y. and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solvong Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869.

[32] Knoll, D. A. and Keyes, D. E., "Jacobian-Free Newton-Krylov Method: A Survay of Approaches and Applications," *Journal of Computational Physics*, Vol. 193, 2004, pp. 357–397.

[33] Chisholm, T. T., *A Fully Coupled Newton-Krylov Solver with a One-Equation Turbulence Model*, Ph.D. thesis, University of Toronto, 2006.

[34] Williamson, C. H. K., "Vortex Dynamics in the Cylinder Wake," *Annual Review of Fluid Mechanics*, Vol. 28, 1996, pp. 477–539.