AIAA 2003-0086

# THE EFFICIENCY OF HIGH ORDER TEMPORAL SCHEMES

Mark H. Carpenter, * Sally A. Viken, † Eric J. Nielsen * ‡

## Abstract

A comparison of four temporal integration techniques is presented in the context of a general purpose aerodynamics solver. The study focuses on the temporal efficiency of high-order schemes, relative to the Backward Differentiation Formulae (BDF2) scheme. The high order algorithms used include the third-order BDF3 scheme, the fourth-order Modified Extended BDF (MEBDF4) scheme, and the fourth-order Explicit, Singly Diagonally Implicit Runge-Kutta (ESDIRK4) scheme.

Design order convergence is observed for all schemes. Specifically, second-, third-, and fourth-order accuracy for the BDF2, BDF3 and MEBDF4 schemes, while the ESDIRK4 scheme converges initially at a fourth-order rate but the order reduces down to third-order at high precisions. Very little advantage is observed with high-order schemes over the popular BDF2 scheme at accuracy tolerances of $10^{-3}$ or less. The MEBDF4 scheme is a possible practical alternative to BDF2 in aerodynamic applications at high precision levels.

## Introduction

The computer resources available in the next decade will provide the opportunity to do routine 3-D computations of $time - dependent$ flows around complex configurations. These computations on large grids are computationally intensive, and require the most efficient time integration techniques. The temporal algorithm currently in vogue in the aerodynamics community is the second-order accurate Backward Differentiation Formulae (BDF2). It is a multi-step scheme that discretely moves the solution forward in time. The BDF2 scheme is linearly stable for arbitrary time-step size (A-stability) and is reasonably robust for nonlinear equations. More sophisticated temporal algorithms are currently available in the ODE literature - most notably algorithms that achieve accuracy levels higher than second-order, while maintaining A-stability.

A primary application for time-dependent methods is in the integration of complex turbulence models. A class of composite models has been developed recently for unsteady flows. These composite models attempt to blend the unsteady capabilities of LES with a method having grid requirements that are more like those conventionally used in Reynolds-averaged Navier-Stokes (RANS) calculations. A composite model typically involves a RANS turbulence model and RANS-type grid in regions near solid surfaces, where the resolution of turbulent eddies would require exceptionally fine grid resolution. A variety of different composite models have been proposed in the last few years. Perhaps the most thoroughly tested has been the Detached Eddy Simulation (DES) model of Spalart[34]. In its most common form, the DES model is implemented in combination with the Spalart-Allmaras (SA) turbulence model, although recently DES results with Mentor's[30] SST model have also been published. All composite models require temporal simulations in three spatial dimensions and are extremely expensive to implement.

An obvious question regarding temporal integrators in the context of time dependent turbulence models (composite or otherwise) is "What is the most efficient integration technique"? Perhaps different optimal schemes exist for different codes or applications. Previous work began the investigation of the utility of high-order temporal techniques. Bijl et al. [4], and later Jothiprasad et al. [19] show that the ESDIRK4 scheme [21] is a factor of two more efficient at achieving engineering accuracy, compared with the BDF2 scheme on several laminar computations. Unfortunately, ESDIRK schemes show order-reduction in the context of turbulent flows [7]. However, the degree to which order reduction affects efficiency is currently unknown for turbulent flows.

The ultimate goal of the current effort work is implementation of an efficient and automated tem-

*Senior Research Scientists, Computational Modeling and Simulation Branch, NASA Langley Research Center, Hampton, VA 23681-2199

†Senior Research Scientist, Flow Physics and Control Branch, NASA Langley Research Center, Hampton, VA 23681-2199

poral integration package for a general purpose 3-D unstructured aerodynamic solver. This package will seamlessly adopt the optimal integration algorithm, and automatically adjust the integration step-size to maintain a user defined accuracy tolerance on high Reynolds number unsteady flows. Automated integration software has existed in various forms for three decades. The first widely available multi step integration library was that developed by Gear,[14] later modified and improved by Hindmarsh,[18] resulting in the LSODE family of codes. Other variants have proliferated over the past two decades to account for the deficiencies of the original approaches (see VODE [5]).

Automated integration control is complex. Embedded logic must control 1) error estimation and step-size prediction, control and rejection, 2) iteration termination strategies, 3) dense output used to predict starting values for the modified Newton process, 4) the timely evaluation of the Jacobian matrix, and 5) re-factorization of the iteration matrix. See, de Swart et al.[12], for a typical application of these principles in the PSIDE software, applied to a four-stage implicit Runge-Kutta method (Radau IIA[17]).

Unfortunately, the control logic used in other general purpose integration software is only partially applicable to our setting. For example, modified Newton methods relying on full Jacobian information, are impractical due to storage limitations on ODE systems of dimensionality $10^6 - 10^8$. Similar difficulties arise in trying to devise optimal convergence strategies with convergence rates in the range $0.1 \leq \alpha_{op} \leq 0.3$[15, 12]. Asymptotic convergence rates in this range are sometimes impossible with current state-of-the-art aerodynamic solver technologies at any $\Delta t$. It is inevitable that portions of the time integration machinery must be adjusted specific to our application.

The first step in the development of an automated integration package is identification of candidate schemes through tests on realistic problems. Four integration schemes are chosen to begin this process. They are the BDF2, BDF3, MEBDF4 and ESDIRK4 schemes. We consider the MEBDF family of schemes as a possible alternative to the ESDIRK schemes. The MEBDF schemes do not suffer from the order reduction and thus should be more suitable for turbulent applications.

A basic control strategy is developed for all four schemes that allows each to run automatically in constant time-step mode. The accuracy, efficiency and convergence characteristics of these schemes are compared on representative flows at low to moderate Reynolds number, and a variety of spatial grid resolutions. The target accuracy for this study is the range $10^{-2} - 10^{-3}$, and shall be refer to as "engineering accuracy". A detailed discussion follows the raw data to help focus future efforts toward the development of an automated integration package.

## Temporal Discretization

Consider the integration of the system of ordinary differential equations (ODEs) represented by the equation [16, 17]

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

In the present case, the vector $\mathbf{S}$ results from the semi-discretization of the turbulent equations of fluid mechanics, containing primarily convection, and diffusion terms. Most typical turbulent aerodynamic flow applications are characterized by:

- Large system dimensionality: $10^5 - 10^8$
- Hyperbolic - parabolic character

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 a range of large eigenvalues. This may give rise to stiffness. Nontrivial near-wall stiffness is common in practical engineering problems due to grid clustering that 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, $||z \rightarrow -\infty||$, be completely damped. Hence, in the paper we consider only L-stable methods and the somewhat less desirable and less stable $L(\alpha)$ methods. Making this choice not only avoids numerical instability but also facilitates convergence of the nonlinear equation solver.

Popular implicit ODE integration methods are defined at both extremes by multistep or multistage methods. Whole classes of schemes, however, combine both approaches in an attempt to correct the flaws of either the multistep or multistage methods.

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 [29]. 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($\alpha$)-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

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

At each time-step the BDF formulae 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 [17]. 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($\alpha$)-stable where $\alpha$ is given by $\{90^o$, $90^o$, $86.03^o$, $73.35^o$, $51.84^o$, $17.84^o\}$. For these same orders, $\beta_k$ is given by {1, 2/3, 6/11, 12/25, 60/137, 60/147}. Smaller values of $\beta_k$ facilitate iterative convergence of the nonlinear algebraic system at each step.

ESDIRK methods [21, 23] are implemented as

$$\begin{aligned}
\mathbf{U}^{k} \;&=\; \mathbf{U}^{n} \;+\; (\Delta t)\sum_{j=1}^{k} a_{kj}\mathbf{S}\left(\mathbf{U}^{j}\right),\; k \;=\; 1, s \\
\mathbf{U}^{n+1} \;&=\; \mathbf{U}^{n} \;+\; (\Delta t)\sum_{j=1}^{s} b_{j}\mathbf{S}\left(\mathbf{U}^{j}\right) \qquad (2)
\end{aligned}$$

$$\hat{\mathbf{U}}^{n+1} \;=\; \mathbf{U}^{n} \;+\; (\Delta t)\sum_{j=1}^{s} \hat{b}_{j}\mathbf{S}\left(\mathbf{U}^{j}\right),$$

where $s$ is the number of stages, $a_{kj}$ are the stage weights, $b_i$ and $\hat{b}_j$ are the main and embedded scheme weights. The vectors **U** and $\hat{\mathbf{U}}$ are the $p^{th}$-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 abscissae 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 §IV.6 in Hairer and Wanner [17]) 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. The Butcher tableau for the ESDIRK4 scheme is included in the Appendix.

To increase the stability of the BDF methods, Cash [8, 9] proposed combining the multistep and multistage ideas and thus developed the Extended Backward (EBDF) and the Modified EBDF (MEBDF) schemes. The MEBDF family of schemes involve three stages to advance the solution one time-step. The first two stages are built from existing $p - 1^{th}$-order BDF formulas, while the last stage combines the two previous BDF results into a $p^{th}$ order solution. Note that the second BDF stage represents a "super-future" point and substantially contributes to the A-Stability of the method. At orders $\{1, 2, 3, 4, 5, 6\}$ the methods are L($\alpha$)-stable where $\alpha$ is given by $\{90^o$, $90^o$, $90^o$, $90^o$, $88.36^o$, $83.07^o\}$. The machinery involved with implementing the MEBDF algorithm is nearly identical to that involved in the BDF formulations. An added advantage of the MEBDF schemes is that very accurate solution data is available on the first and third stages, based on previous information. This information can be used to provide the starting guess for the nonlinear interaction, and to establish time-step error estimates. The second stage typically uses the trivial guess as the starting point for the nonlinear iteration and no error estimate is made.

## Numerics

The target for our automated integration package is the 3-D unstructured- aerodynamics solver FUN3D[3] and its replacement solver, the HEFSS code that is currently being upgraded [22, 1]. The HEFSS solver utilizes mixed 3D unstructured elements to solve the turbulent equations of fluid mechanics. A variety of algebraic solvers including multi-grid are used to converge the governing equations. Unfortunately, this platform is not currently available, so we begin our study with the 2D predecessor of FUN3D.

The FUN2D flow solver is a node based, implicit, upwind flow solver used for computing flows around airfoil configurations discretized with an unstructured grid.[2] The governing equations are the time-dependent Reynolds-Averaged Navier-Stokes equations in conservation-law form. The inviscid fluxes are obtained on the faces of each control volume by using the flux-difference-splitting (FDS) technique of Roe.[32] A node-based algorithm is used in which the variables are stored at the vertices of the mesh and the equations are solved on non-overlapping control volumes surrounding each node. The viscous terms are evaluated with a finite-volume formulation that results in a central-difference-type scheme. The Spalart-Allmaras[33] (SA) turbulence model is used in this investigation and all computations assume fully turbulent flow.

A two level iteration is used to achieve convergence of the discrete algebraic equations at each stage/time-step. The outer iteration is an modified Newton method and employs a first-order Van Leer [35] Jacobian (LHS) driving the second-order residual vector (RHS). The inner iteration employs a red-black Gauss Siedel point-implicit algorithm to solve the equations at each step of the outer iteration. Fifteen inner sub-iterations are used in all cases in this study. The SA turbulence model equations are weakly coupled to the hydrodynamic equations via the outer loop iteration.

### Error Estimation

A major component in the development of automated integration software is the development of reliable local error estimators. Error estimation is accomplished by comparing two solutions of different orders ($\mathbf{U}^{n+1}$ and $\hat{\mathbf{U}}^{n+1}$) at the same time-step. For reasons of efficiency, the auxiliary solution $\hat{\mathbf{U}}^{n+1}$ should be available at little additional cost. For example, in ESDIRK schemes (see eqn. 3), as well as MEBDF [9] schemes, both $\mathbf{U}^{n+1}$ and $\hat{\mathbf{U}}^{n+1}$ are constructed from available data. The difference $||\mathbf{U}^{n+1} - \hat{\mathbf{U}}^{n+1}||$ is proportional to the truncation error of the lower order formula $\hat{\mathbf{U}}^{n+1}$. The estimate predicts the magnitude of the error in the solution, and gives insight into its overall quality. Frequently, linear and nonlinear instability can be predicted by the estimator well before the simulation diverges.

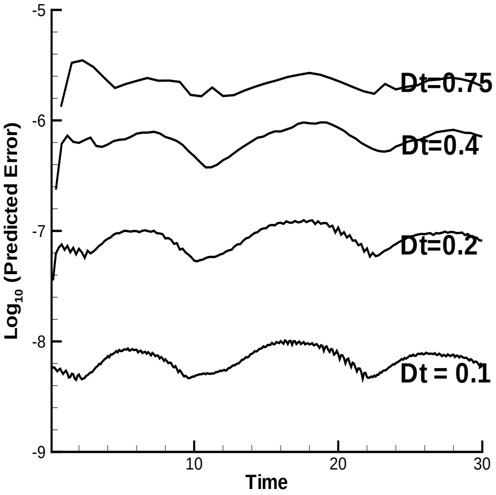

Figure 1. *Cycle variation of predicted temporal error as calculated with MEBDF4.*

Figure (1) shows the error estimate (MEBDF4) for various $\Delta t$. The test problem is for periodic shedding from the turbulent circular cylinder. Although not shown, the estimates are accurate to the correct order based on grid-converged data. The temporal error estimate varies by approximately a factor of three over one shedding cycle. Adjusting the time-step by a factor of 1.5 over the course of the cycle would result in nearly uniform temporal error. Variable time-stepping for this mildly unsteady cylinder application would be relatively simple even with multi-step schemes.

A rudimentary error estimator is developed for the BDF2 and BDF3 schemes. The leading order truncation term for the second-order formula is the third derivative $(\Delta t)^2 \frac{\partial^3 U}{\partial t^3}$, and can be approximated by the difference between second- and third- order temporal formulas. For example, the second- and third-order BDF integration formulas written for variable time-step mode can be expressed as

$$\frac{\partial U}{\partial t} = a_{+1}\, U_{n+1} \; + \; a_{+0}\, U_n \; + \; a_{-1}\, U_{n-1} \; + \; a_{-2}\, U_{n-2}$$

with

$$a^2_{+1} \quad = \quad \frac{(h1 + 2h0)}{(h0(h1 + h0))}$$

$$\begin{aligned} a^2_{+0} &= \frac{-(h1+h0)}{(h0h1)} \\ a^2_{-1} &= \frac{h0}{(h1(h1+h0))} \\ a^2_{-2} &= 0 \end{aligned}$$

for BDF2 and

$$\begin{aligned} a^3_{+1} &= \frac{(h1h2+2h0h2+h1^2+4h0h1+3h0^2)}{(h0(h1+h0)(h2+h1+h0))} \\ a^3_{+0} &= \frac{-(h1h2+h0h2+h1^2+2h0h1+h0^2)}{(h0h1(h2+h1))} \\ a^3_{-1} &= \frac{(h0h2+h0h1+h0^2)}{(h1(h1+h0)h2)} \\ a^3_{-2} &= \frac{-(h0h1+h0^2)}{(h2(h2+h1)(h2+h1+h0))} \end{aligned}$$

for BDF3, respectively. The time-steps $h0$, $h1$ and $h2$ are given by $h0 = t^{n+1} - t^n$, $h1 = t^n - t^{n-1}$, and $h2 = t^{n-1} - t^{n-2}$. Thus, the error estimator for the BDF2 scheme is

$$\epsilon_2 = \|(\frac{\partial U}{\partial t})_3 - (\frac{\partial U}{\partial t})_2\|$$

The error estimate for the BDF3 scheme is taken as $\epsilon_3 = \epsilon_2 h0$. Note that more sophisticated estimators are used in general purpose BDF solvers. The CVODE[11] code uses predictor-corrector differences for example. The present estimator is sufficient for the purpose of this study. In future work, the BDF estimator will be enhanced to be compatible with more sophisticated control strategies.

The MEBDF4 scheme uses as an error estimate the difference between the solutions obtained on the first and third stages respectively. The first stage is a third-order estimate of the solution at the new time. The third stage is the final solution, and is fourth-order accurate in time. The difference is proportional to $\Delta t^4$ and is an extremely good estimate of error. An additional error estimate is available by comparing the solution at the second stage of the previous time-step, with the solution at the end of the first stage. This estimate compares two third-order formulas, and is also very accurate.

The Runge-Kutta scheme uses the difference between the main and the embedded solution for the error estimator. (See equation (3)) for the main and embedded formulas and Kennedy and Carpenter[21] for the coefficient values. In all four temporal integration formulas, the spatial surface area (volume in 3D) of integration is included in the norm. This scales the estimate in proportion to the local volume size.

Termination Strategy

An accurate error estimate can also be used to automate the termination strategy of the nonlinear iteration. Two competing components of temporal error are the *truncation* and *algebraic* errors. Truncation error is related to $\Delta t$ and the order of accuracy $p$, while algebraic error is the residual error generated each time-step by approximately solving the algebraic system. The local temporal error is the *sum* of the two components. To see full design order from the temporal scheme, the algebraic error must be driven below the truncation error at each time-step. This requires an accurate measure of truncation error, and must be provided by the error estimator.

The iteration termination strategy is complicated. An insufficient number of iterations on any time-step can destroy the formal solution accuracy, while excessive iterations decrease the efficiency. Our experience indicates that design-order temporal convergence is achieved by maintaining a tolerance ratio of $10^{-3} \leq \mathcal{T} \leq 10^{-1}$. Here $\mathcal{T}$ is defined as the ratio of nonlinear algebraic error to predicted temporal integration error at each time-step (or stage). Algebraic error for the nonlinear iteration is based on the $L_\infty$ norm of the density residual. The wide range for $\mathcal{T}$ results from the variety of error estimators used in the different algorithms.

Several preliminary tests determine the largest value of the parameter $\mathcal{T}$ that still achieves design order accuracy of each method. All calculations are run in fixed time-step mode and terminated when the tolerance ratio is met. Larger values of $\mathcal{T}$ result in sporadic convergence results as the algebraic error disrupts each method's design order convergence slope. The test case used to train the termination strategy is the turbulent circular cylinder at a Reynolds number of $Re = 10^4$ and a Mach number of $Ma = 0.2$. Table (1) shows the experimentally determined values of the parameter $\mathcal{T}$.

**Table 1**: Upper bounds on the tolerance ratio parameter $\mathcal{T}$.

| Scheme | $\mathcal{T}$ |
|---|---|
| BDF2 | 1/20 |
| BDF3 | 1/500 |
| MEBDF4 | 1/50 |
| ESDIRK4 | 1/50 |

Note that choosing the time-step based on accuracy considerations alone may not be the most efficient strategy for a temporal calculation. Decreasing the time-step could greatly increase the convergence rate of the nonlinear algebraic system, thus increasing efficiency. Gustafsson and Söderlind [15] devised optimal criteria for adjusting $\Delta t$. They assumed that either fixed point iterations, or modified Newton iterations is used for solving the algebraic system. The time-step is adjusted so that the iteration convergence rate approximately equals the optimal value. Typical CFD algebraic solvers fall somewhere between fixed point and modified Newton iterations. Additional work is needed to refine controllers in the context of CFD time-dependent solvers, and is the subject of future work. All calculations performed in this work (except those needed for startup) use a constant time-step.

## Results

### ODEs

Cash [10] updates the considerable progress made through the mid 90's for method of lines algorithms applied to stiff oscillatory problems. His study focuses on the BDF, MEBDF, and Runge-Kutta classes of schemes. The model problem Cash uses to compare the efficiencies of the different schemes is the stiff, damped oscillatory problem of Enright et al.[13]. See Cash [10] for a problem description, but note that oscillatory in this context refers to complex eigenvalues. This problem highlights the inadequacies of the BDF class of schemes. It is designed with complex eigenvalues that lie near the imaginary axis in the unstable lobes of the third and higher order BDF schemes. To maintain stability, all BDF schemes of accuracy greater than two must be run with an explicit time-step. Thus, the BDF family of schemes is not efficient unless extremely coarse tolerances are desired, and BDF2 can be used.

Unfortunately, Cash's study is only partially applicable to the problems in fluid mechanics. The Navier-Stokes equations govern the time variation of viscous fluids. These flows are usually oscillatory and exhibit considerable stiffness. In addition, the eigenvalues of the linearized equations often are near the imaginary axis. A notable difference, however, is the precise location of the complex eigenvalues. Frequently the BDF3 scheme is stable on realistic problems with no reduction in time-step, making it an extremely efficient algorithm for engineering calculations.

We now compare the behavior of the candidate schemes on a model problem that is simple enough to be thoroughly tested. Rather than use Enright's problem, we choose van der Pol's equation as our test vehicle, because it does not significantly penalize the BDF3 scheme. Van der Pol's equation is used in the work of Kennedy and Carpenter[21] to determine the relative merits of integration schemes, because of its challenging character. See Hairer[17] pages 402-403 for the exact problem specification and initial conditions. Fixed time-step mode is used in all cases to allow direct comparison with the results in Hairer.

Van der Pol's equation is a two variable system that separates into a non-stiff and stiff variable, with the non-stiff differential variable being the variable of interest. Figure (2) shows work-precision plots for the differential (non-stiff) and algebraic (stiff) variables in van der Pol's equation. Work is defined by the total number of function evaluations. All plots show the solution accuracy for a stiffness of $\epsilon = 10^{-3}$. This value of the stiffness parameter $\epsilon$ is chosen from the range $10^{-8} \leq \epsilon \leq 1$, and is representative of the entire range. The nonlinear system is solved at each time-step using either Newton's method or Jacobi iteration and is driven to machine precision. Newton's method requires between 5 and 11 iterations to reach convergence, making work nearly proportional to time-step. Jacobi iteration required as few as 5 iterations to converge, but varied noticeably with stiffness $\epsilon$ and time-step. Runge-Kutta would not converge for $\epsilon = 10^{-8}$ with Jacobi iteration.

The theoretical convergence rate of each scheme is presented in Table (2). The theoretical rates are consistent with the numerically determined rates shown in figure (2). (The differential and algebraic convergence rates shown for MEBDF4 are speculative, based on the similar analysis presented in Hairer[17].) The first two plots show the convergence of the non-stiff and stiff variables, respectively as a function of work. Newton's method is used to drive the algebraic equations to zero. The convergence rate of BDF2, BDF3 and MEBDF4 is independent of the stiffness parameter $\epsilon$ for both the non-stiff and stiff variables. The ESDIRK4 scheme order reduces due to the stiffness in the problem. The non-stiff differential variable converges initially at a fourth-order rate, but abruptly changes to a lesser slope at fine tolerances. The stiff algebraic variable order reduces abruptly at an error of $-6$ for the ESDIRK4 scheme. The ESDIRK4 scheme is the most efficient of the four candidate schemes in van der Pol's equation, despite the dramatic order reduction present in the algebraic variable.

**Table 2**: Theoretical convergence rates of high-order schemes.

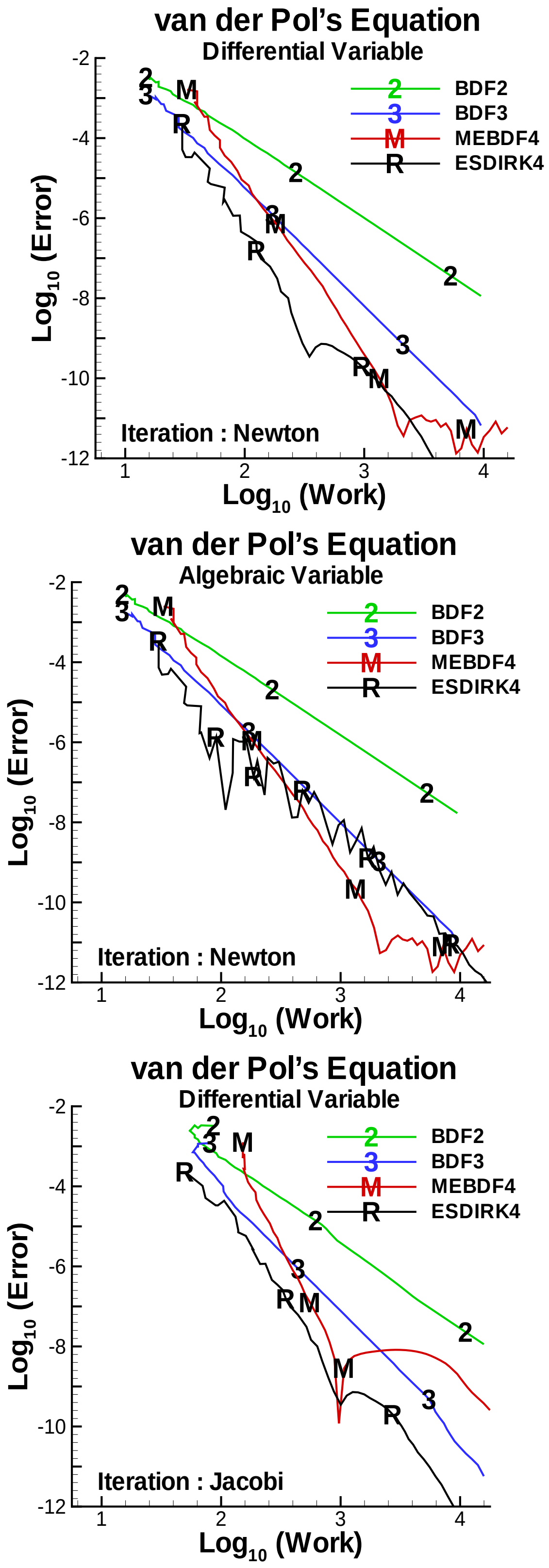

Figure 2. *Work-precision plots for the BDF2, BDF3, MEBDF4, and ESDIRK4 schemes. Top and middle plots show differential and algebraic errors converged with Newton iteration. The bottom plot shows the differential error as converged with Jacobi iteration.*

| scheme | Differential | Algebraic |
|---|---|---|
| BDF2 | $h^2$ | $h^2$ |
| BDF3 | $h^3$ | $h^3$ |
| MEBDF4 | $h^4$ | $h^4 + \epsilon h^3$ |
| ESDIRK4 | $h^4 + \epsilon h^3$ | $h^4 + \epsilon h^2$ |

The first two work-precision plots of figure (2) are idealistic comparisons of the general behavior of the four candidate schemes. Implicit in the comparison is similar convergence of the algebraic system, independent of method, time-step and stiffness. The last plot, however, shows the work-precision behavior obtained using Jacobi iteration for the algebraic system. Inadequate convergence characteristics of the algebraic solver can have disastrous effects on any temporal algorithm. The Jacobi iteration is far less effective in driving the residuals to zero, and diverges in some cases (ESDIRK4 at $\epsilon < 1.0e-8$). Jacobi iteration shifts all curves towards larger values of work. More instructive, however, is the anomaly that occurs with the MEBDF4 scheme at high precision. The slow iterative convergence dramatically increases the work.

This preliminary study of van der Pol's equation is used to carefully study the four candidate schemes in an ideal setting. Unfortunately, it only begins to unravel the complex work-precision interactions encountered in FUN2D. In closing, efficiency is strongly influenced by

- Convergence rate of iterative solver
- Stages necessary to implement scheme
- Accuracy per step

A scheme that has desirable accuracy properties may not be an efficient solver. This is particularly true for high-order solvers where huge time-steps are needed to utilize their high accuracy per step.

## PDEs

The van der Pol's equation study compares the nuances of the four candidate schemes in an extremely clean setting. The present study attempts to establish on realistic applications, the stability, convergence, and efficiency of the four schemes. The study begins by establishing the sensitivities to grid resolution as well as Reynolds number effects. A circular cylinder is used to quantify the sensitivity of temporal results on varying spatial resolution. Reynolds number effects are studied next, on a circular cylinder and a NACA0012 airfoil.

All test cases involve systematic refinement studies at each flow condition. Each study begins by advancing the solution at a large relative error (MEBDF4 with $\Delta t = 1$ yields an error of $10^{-4}$/step

) for approximately 100 shedding cycles. At this point the solution is periodic. One hundred steps are then run at a time-step of $\Delta t = 0.01$ with the MEBDF4 algorithm. The three time-plains of startup data that are written at the end of each run correspond to $\Delta t = 0.01$. The stored start-up data is fourth-order accurate and is obtained on the smallest time-step used in the simulation, and is used by all the multi-step algorithms (ESDIRK4 does not need it) in variable time-stepping mode without loss of formal accuracy. Finally, the "exact solution" is obtained by running the BDF3 algorithm at $\Delta t = 0.01$ for 1 1/2 shedding cycles.

Data from each run is processed using the exact solution data. Error estimates are obtained using $L_2$ and $L_\infty$ norms over the entire solution field. All the norms reported herein are un-weighted norms and do not involve geometric information. A comparison of area-weighted and un-weighted reveals that both yield similar results, distinguishable only by a scaling factor.

Grid dependence

A spatial refinement study of the circular cylinder is performed to provide insight into the influence of space resolution on the performance of temporal algorithms. An additional objective is determination of the iteration termination procedure for all schemes. The termination cutoffs for all schemes are determined numerically by experimentation.

Three spatial discretizations are chosen for the grid dependence study. Table (3) presents the grid nodes, surface nodes and minimum normal wall spacing for of each of the cylinder grids. All grids are tailored for the $Re = 10^4$ turbulent flow regime. Wall normal spacing is determined based on turbulent boundary layer growth on a flat plate.

**Table 3**: Grid information for spatial refinement study.

| Grid | Grid Nodes | Surf. Nodes | $\Delta x_{min}$ |
|---|---|---|---|
| Coarse | 8716 | 100 | 1.80e-3 |
| Medium | 30838 | 200 | 0.90e-3 |
| Fine | 115498 | 400 | 0.45e-3 |

Figure (3) shows a comparison of the Strouhal numbers on a coarse, medium and fine grids for the case Reynolds, and Mach numbers equal $10^4$, and 0.20, respectively. The twenty cycles shown are run from the periodic initial data stored previously using a time-step $\Delta t = 0.01$. All simulations are run with with MEBDF4 at a time-step of $\Delta t = 1/2$. The initial transient in each curve reflects the discrepancy between the intrinsic temporal error in the stored initial data, and the temporal error of the present calculations. Strouhal number as a function of cycle is determined by measuring the time needed to traverse one cycle in the lift variable. Lift equals zero in each cycle is used as the datum to reference the beginning and end of a cycle. Cubic interpolation is used to approximate the time for which lift is zero in each case.

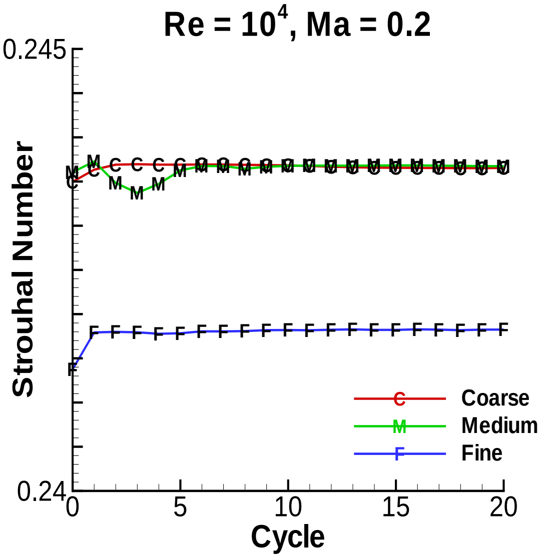

Figure 3. *Comparison of Strouhal number on the coarse, medium and fine grid for a turbulent circular cylinder at Reynolds and Mach numbers of* $10^4$*, and* 0.20*, respectively.*

The Strouhal number variation between all three grids is approximately one percent. The results obtain on the medium grid represent engineering accuracy of approximately 2 significant digits, and are a reasonable compromise between accuracy and efficiency. The general guidelines used to establish this medium grid are used in subsequent cylinder comparisons, and those involving the NACA0012 airfoil.

We now turn to temporal accuracy. Figure (4), shows the nature of convergence for the BDF2, BDF3, MEBDF4 and ESDIRK4 schemes on the coarse, medium and fine grids, respectively. Plotted is the $L_2$ convergence of the density field. The time interval is 30 time units, or about one and one half shedding cycles. The exact solution is determined from a BDF3 run at a time-step of $\Delta t = 0.01$, and is accurate to approximately 7 significant digits throughout the density field. Plots of the $L_\infty$ convergence of the density field show the same qualitative behavior, with the absolute level of error shifted up by about two orders of magnitude. Note that for large time steps ($\Delta t \rightarrow 10$), all curves asymptote to the datum Logarithm (Error) $\approx -2$. This datum is adjustable, and results from the specific choices made in determining the norm. (For example, is grid volume included in the norm). It is important to recognize that absolute $L_2$ error is the difference between the plotted result and the asymptotic datum $-2$. For example, results with accuracy of two significant digits occur at approximately Logarithm (Error) $\approx -4$. The time-step needed to achieve this accuracy is approximately $\Delta t = 0.1, 0.4, 0.75, 2.0$ for BDF2, BDF3, MEBDF4, and ESDIRK4, respectively.

Figure (4) demonstrates the design nature of the convergence of all four schemes. The BDF2, BDF3 schemes converge at rates that are extremely close to 2 and 3, respectively, and are nearly the same in absolute magnitude. The MEBDF4 scheme converges at a rate approaching four. The ESDIRK4 scheme initially converges at a rate approaching four. Order-reduction changes the convergence with smaller time-steps to approximately third-order temporal accuracy.

Verwer[36] showed that diagonally implicit Runge-Kutta (DIRK) schemes order reduce when applied to method of lines ODEs. His test case includes a spatial operator that is discontinuous on the scale of the spatial grid, and produces a stage-order plus one convergence rate for the DIRK scheme. Note the coarse to fine convergence behavior of the ESDIRK4 scheme shown in figure (4). Nearly smooth convergence profiles are obtained for the ESDIRK4 scheme on the fine grid, while sporadic convergence is experienced on the coarse grid. We speculate that the smoothness of the spatial operator causes this variation. The coefficients on the fine grid are nearly smooth, while those on the coarse grid are somewhat discontinuous.

The design order nature convergence in the previous resolution study, indicates that the temporal algorithms appear to be implemented correctly. In addition, the iteration termination strategy used at each stage or time-step appears to be adequate.

Figure (5) presents a work-precision study comparing all four schemes. Each case covers an accuracy range of approximately five order of magnitude. Perhaps the most striking feature about the results from this study is the close proximity of **ALL** curves. The largest variations occur at intermediate precision on the fine grid. The largest difference between any of the schemes is less than a factor of three. All curves vary by less than a factor of 1.5 on the coarse grid. Results on the medium grid fall between the coarse and fine grid results.

More subtle distinctions include the following. The BDF2 scheme is the most efficient of the four schemes at precisions less than two significant digits (-4 on the plots). The MEBDF4 scheme becomes the most efficient at higher precisions. The ESDIRK4 and BDF3 schemes never are as efficient at any precision. At high spatial resolution, the the BDF2 scheme is most efficient up to four significant digits (-6 on the plot), before being replaced by MEBDF4.

The work-precision behavior of the BDF2 and BDF3 schemes is counterintuitive. Both are single step schemes and require the same number of steps to traverse the time interval. The BDF3 scheme is more accurate, thus implying that it will be more efficient. The perplexing BDF2/BDF3 results in figure (5) stem from the constant time-step controller. Algebraic error is always driven below truncation error in this study. The initial portion of each iteration converges rapidly, while the asymptotic portion of the iteration is very slow. The BDF2 scheme benefits from the rapid initial convergence of the iteration, while the BDF3 suffers from the slow asymptotic convergence needed to achieve the necessary tolerance ratio $\mathcal{T}$. The BDF3 results in this study would improve with a different controller. For example, implementing the BDF3 scheme with the control strategy used in the BDF2 scheme should produce similar work-precision plots, although both would be second-order accurate.

The total work is relatively insensitive to grid resolution. (Note that work is defined as the number of total residual evaluations, not CPU time). The fine resolution cases converge in about twice as many iterations as do the coarse grid cases.

Combining the results from figures (4) and (5), leads to the general conclusion that spatial resolution at this set of flow conditions does not strongly influence the temporal algorithmic trends. Thus, future studies of other flow conditions or geometries, can focus extensively on only one grid resolution instead of three.

Reynolds Number effects

A second objective of the study is to determine the sensitivity of temporal algorithms to the effects of Reynolds number. The resolution requirements in the near-wall region increase with increasing Reynolds number. Fine grids increase the stiffness stiffness in the problem. The effects of Reynolds number on the convergence, and efficiency of temporal schemes is now presented.

The study begins with the Reynolds number effects on the circular cylinder. The Reynolds number is varied over the range $Re = 10^3, 10^4, 10^5$. The Mach number is assumed constant at $Ma = 0.2$. Table (4) shows the relevant features of the grids used in this study. The general guidelines used to estab-

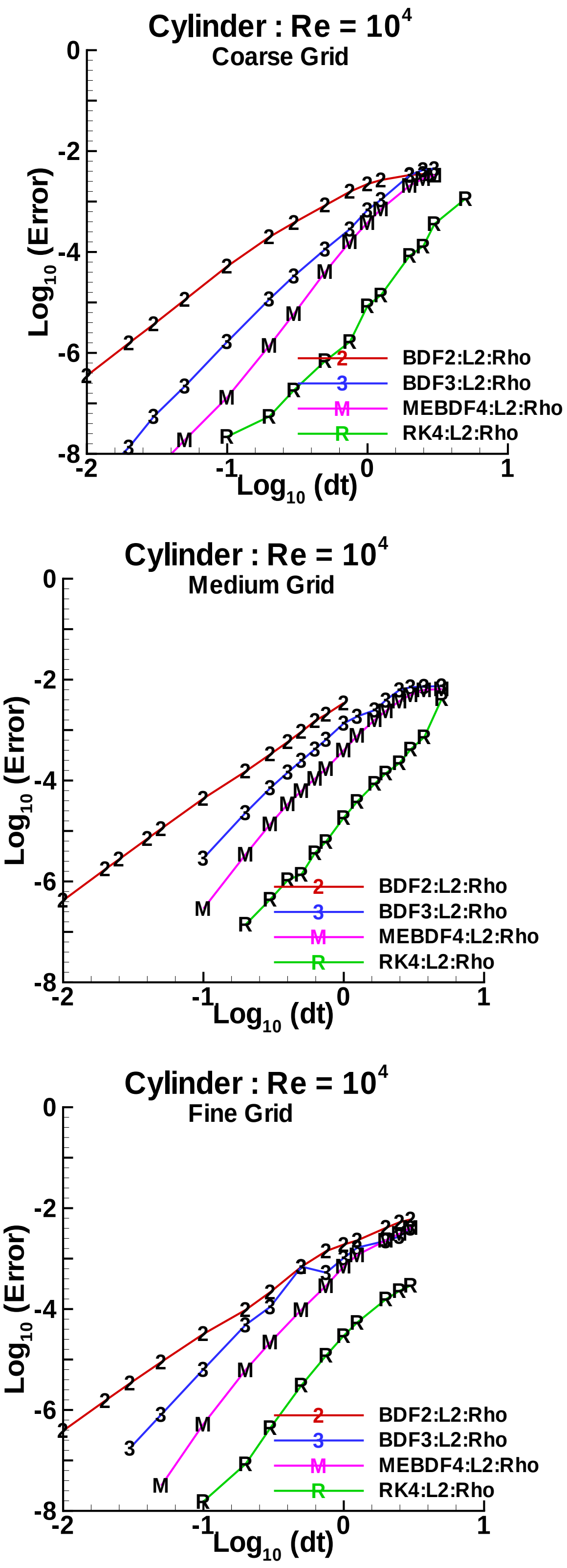

Figure 4. *Temporal convergence of BDF2, BDF3, MEBDF4, and ESDIRK4 on a coarse, medium, and fine grid circular cylinder. Test conditions are* $Re = 10^4$*, and* $Ma = 0.20$.

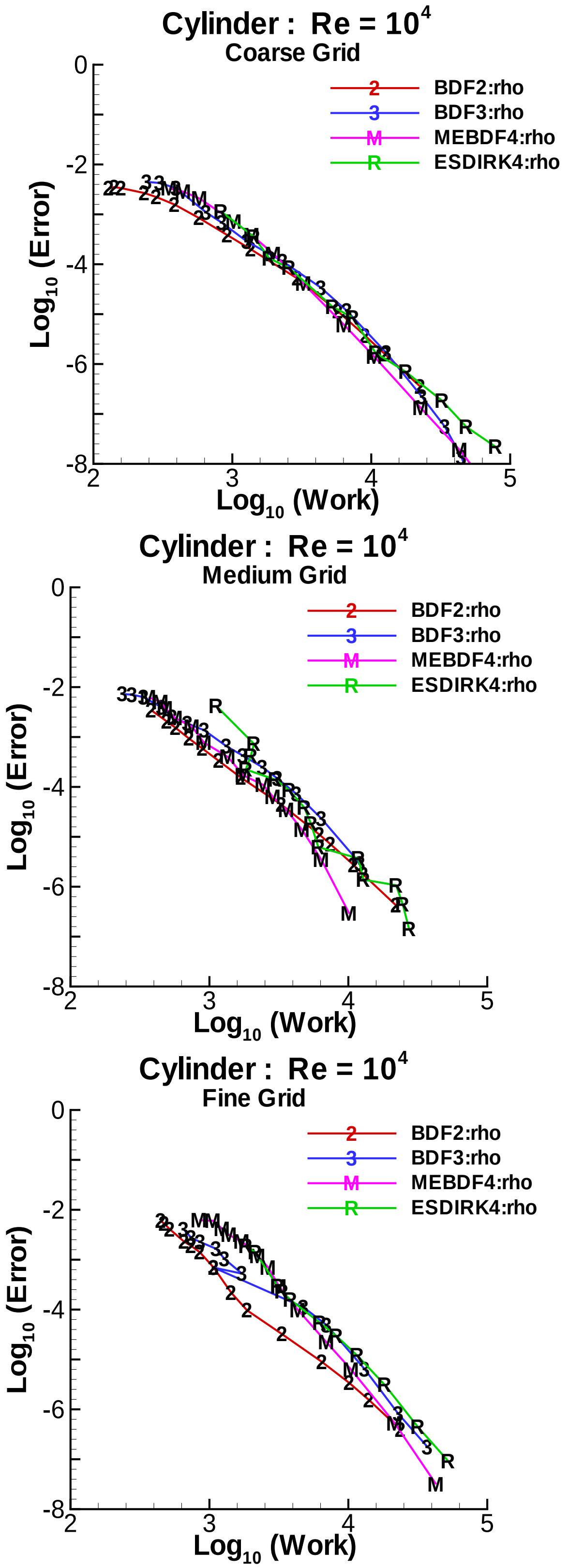

Figure 5. *Work-precision plots for the BDF2, BDF3, MEBDF4, and ESDIRK4 on a coarse, medium, and fine grid circular cylinder. Test conditions are* $Re = 10^4$*, and* $Ma = 0.20$.

lish the coarse and fine grids are identical to those used in developing the medium grid in the previous circular cylinder resolution study.

**Table 4**: Grid information for cylinder study of Reynolds number.

| Reynolds | Grid Nodes | Surf. Nodes | $\Delta x_{min}$ |
|---|---|---|---|
| 1000 | 28501 | 200 | 5.00e-3 |
| 10000 | 30838 | 200 | 0.90e-3 |
| 100000 | 61933 | 200 | 0.10e-3 |

Figures (6) and (7) show the temporal $L_2$ density convergence and work-precision plots obtained on the circular cylinder at the three previously described flow conditions. Design order convergence is achieved on the lower two Reynolds number cases. The absolute level of error increases with Reynolds number on these two cases. The BDF2 and MEBDF4 schemes are the two most efficient schemes, with BDF2 winning at low precision, and MEBDF4 winning at high-precision. The cross over point occurs at approximately an accuracy of two significant digits. (This is at about $-4$ based on the asymptotic zero datum of $-2$). Little variation in efficiency is evident (perhaps a factor of two) in figure (7) on the two lower Reynolds numbers.

Significant iteration convergence obstacles plagued the Reynolds number $Re = 10^5$ circular cylinder test case. The solution of the algebraic equations resulting from each discretization, could not be driven to an arbitrary level of convergence, regardless of the number of sub-iterations. This "hanging residual" made it impossible to reach the tolerance ratio $\mathcal{T}$ necessary to achieve design order accuracy. The adverse effects of inadequate convergence are shown in the third frame of figure (6). The convergence problems are so severe for the $Re = 10^5$ case that no work-precision plot is available for this Reynolds number.

The exact cause of the convergence problems is unknown for the $Re = 10^5$ test case. The turbulence field indefinitely flips between two approximate states after an initial convergence of two to three orders. Perhaps the cause of instability is the drag crisis on the cylinder at approximately this Reynolds number, combined with the weak coupling of the turbulence model. The original intent of the Reynolds number study was to include $Re = 10^6$. Poor algebraic convergence at high Reynolds numbers made it impossible to continue studying the cylinder.

A second study of Reynolds number effects is presented and provides more conclusive trends for the temporal integrators study. A NACA0012 airfoil at $30^o$ angle of attack is used in this study. The Reynolds number is varied over the range $Re = 10^3,\ 10^4,\ 10^5$. The Mach number is assumed con-

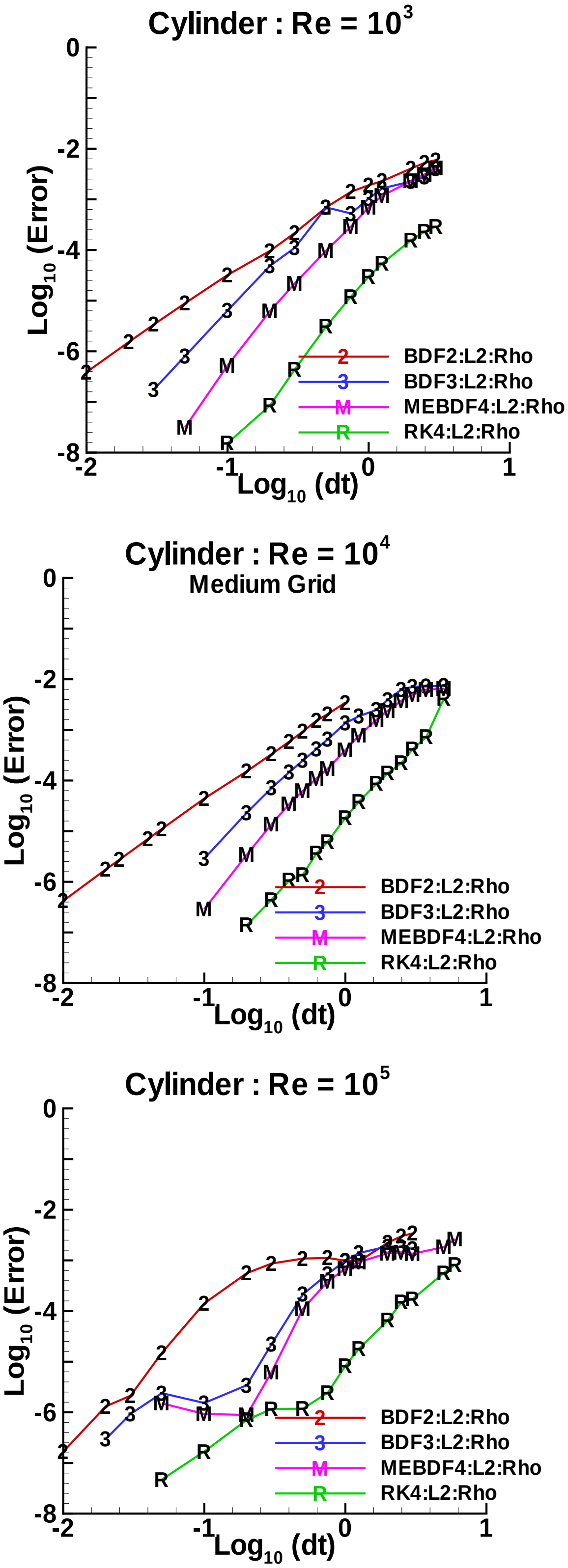

Figure 6. *Temporal convergence for the BDF2, BDF3, MEBDF4, and ESDIRK4 schemes on a circular cylinder at* $Re = 10^3,\ 10^4,\ 10^5$. *The Mach number is* $Ma = 0.2$.

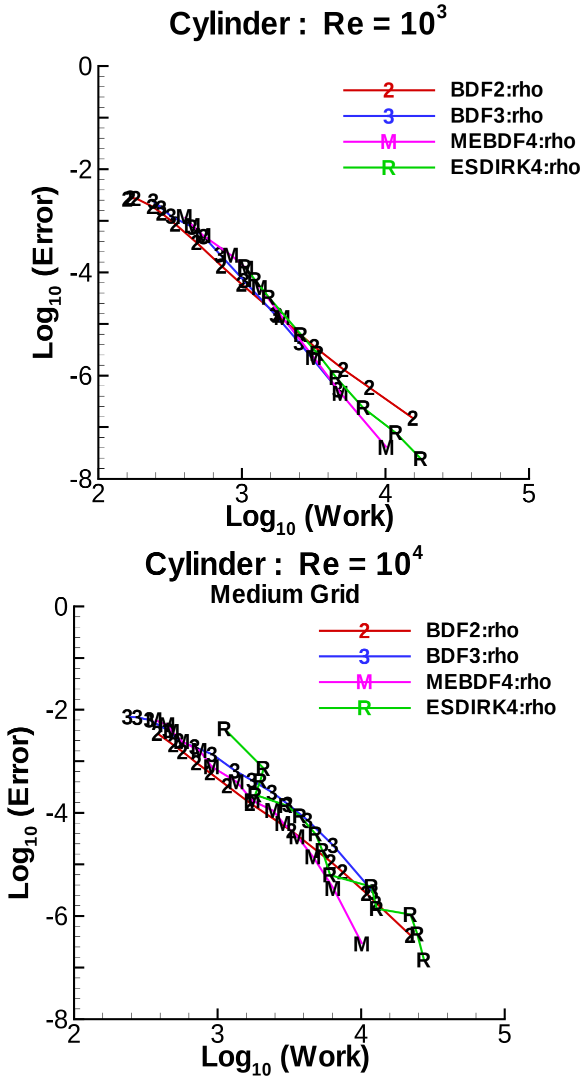

Figure 7. *Work-precision plots for the BDF2, BDF3, MEBDF4, and ESDIRK4 schemes on a circular cylinder at $Re = 10^3$, $10^4$. The Mach number is $Ma = 0.2$.*

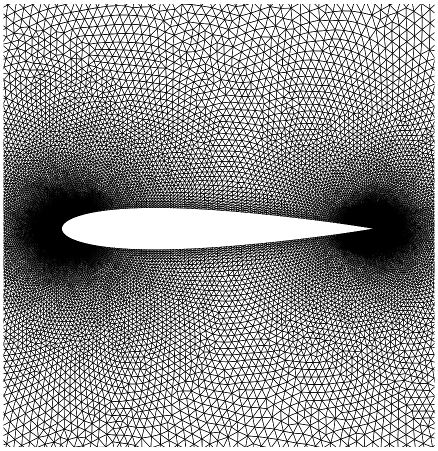

Figure 8. *Unstructured grid for the NACA0012 test case at a Reynolds number of $Re = 10^3$*

stant at $Ma = 0.3$. Figure (8) shows the grid for the $Re = 10^3$ case. Table (5) shows the important features of the grids used in this study. The general guidelines used to establish this grid was similar to those used in developing the medium grid in the previous circular cylinder resolution study.

**Table 5**: Grid information for NACA0012 study of Reynolds number.

| Reynolds | Grid Nodes | Surf. Nodes | $\Delta x_{min}$ |
|---|---|---|---|
| 1000 | 52997 | 564 | 5.00e-3 |
| 10000 | 55090 | 564 | 0.90e-3 |
| 100000 | 61933 | 564 | 0.10e-3 |

Figures (9) and (10) show the temporal convergence and work-precision plots obtained on the NACA0012 at the three previously described flow conditions. The convergence plot is based on the $L_2$ error in the density equation, although $L_\infty$ yields nearly identical results. (Error increases uniformly by approximately two orders of magnitude). In general terms, the absolute accuracy of the density fields in the three test cases decreases with increasing Reynolds number. The zero accuracy datum is again approached asymptotically at a value of approximately $-2$. Solution accuracies with two significant digits are achieved at an error level of approximately $-4$.

The NACA0012 temporal convergence and work-precision plots are qualitatively similar to those presented in the cylinder spatial refinement study, and Reynolds number study. Specifically, the design nature of the BDF2, BDF3, MEBDF4, and ESDIRK4 schemes are evident in figure (9). Figure (10) shows that the variation in the efficiency between schemes is less than a factor of three. The BDF2 scheme is slightly more efficient at low precision, while the MEBDF4 scheme is more efficient at high precision. The BDF3 and ESDIRK4 schemes are never the most efficient schemes. The ESDIRK4 scheme deviates from the other schemes most significantly in the low Reynolds number and at low levels of precision.

Subtle differences can be observed in figures (9) and (10) , that were not present in figures (6) and (7). The nature of temporal convergence in the NACA0012 study is more sporadic compared with the previous grid resolution study. The asymptotic convergence rates are not precisely second-, third-, fourth- and fourth-order for the BDF2, BDF3, MEBDF4 and ESDIRK4 schemes, respectively.

A significant (yet not insurmountable) iteration obstacle plagued the Reynolds number $Re = 10^5$ NACA0012 test case. The solution of the algebraic equations resulting from each discretization, could not be driven to an arbitrary level of convergence, regardless of the number of sub-iterations. This "hanging residual" apparently did not adversely affect the convergence plots shown in figure (9). The work-precision plots, however, are directly affected by a "hanging residual". Table (6) shows the test cases that experienced convergence difficulties. The algebraic system failed to converge in the indicated cases for at least one step or stage during the integration. This data was disregarded in completing the work-precision plot, and was replaced with the number of iterations required for the previous algebraic iteration.

**Table 6**: Test cases that failed to converge for NACA0012 at $Re = 10^5$.

| $\Delta t$ | BDF2 | BDF3 | MEBDF4 | ESDIRK4 |
|---|---|---|---|---|
| 0.1 | | X | X | |
| 0.3 | | | | X |
| 0.5 | | | | X |
| 0.75 | | | | X |
| 1.0 | | | X | |
| 2.5 | | | X | X |

A deeper understanding of the efficiency of all four schemes can be gained by reviewing the number of iterations necessary to converge the nonlinear equations at each step. Figure (11) is an iteration-time-step plot at each of the three Reynolds numbers. The average number of iterations required to reach the termination point at each stage (ESDIRK4 and MEBDF4) or time-step (BDF2, BDF3) is plotted as a function of time-step. (Note that data was averaged in the $Re = 10^5$ case to account for non-converging iterations).

Figure (11) is the key to the efficiency of the different schemes. The BDF2 scheme has a very well behaved iteration behavior. The number of iterations monotonically increases with time-step in a smooth fashion. The MEBDF4 scheme is nearly as well behaved, but occasionally experiences convergence problems on the second stage of the method. (This is the stage for which the trivial guess is used to start the algebraic iteration).

The BDF3 and ESDIRK4 schemes both experience convergence problems. They have difficulty converging at both large time-steps and at small time-steps, and their efficiency suffers accordingly. Several factors influence the convergence rate for large time steps. The diagonal dominance of the algebraic system decreases with increasing time-step, making convergence slower. In addition, the trivial guess used to start the nonlinear iteration becomes progressively worse as time-step is increased. Failure to converge for small time-steps is due to inadequate convergence of the turbulence model for the high precision cases. The weak coupling of the turbulence model becomes an issue as residuals are driven toward machine precision. The residual initially converges approximately three to four orders. It then begins to oscillate between two states and fails to converge any further. The BDF2 scheme terminations its iteration before convergence problems are encountered.

## Discussion

Design order convergence is achieved for all candidate schemes in the FUN2D code. The relative accuracy of all methods (at a given time-step) is similar to that predicted in van der Pol's equation. In practical terms, the BDF2 scheme achieves engineering accuracy for a time-step of $\Delta t = 0.1$, which corresponds of approximately 100 time steps per period. The fourth-order ESDIRK formulation attains a similar accuracy (in spite of order reduction) using 10 time steps per period, with five stages per step, yielding approximately 50 time-samples/period. The high order schemes are able to achieve solutions with engineering accuracy using huge time steps. We speculate that new state-of-the-art general linear methods (GLM) methods[6] could lower this estimate to 30 time-samples/period, an improvement of approximately $O(10^{1/2})$ over BDF2 temporal accuracy.

High-order formulations suffer in the FUN2D code from poor efficiency. The BDF2 scheme is equally efficient in achieving the target accuracy levels used

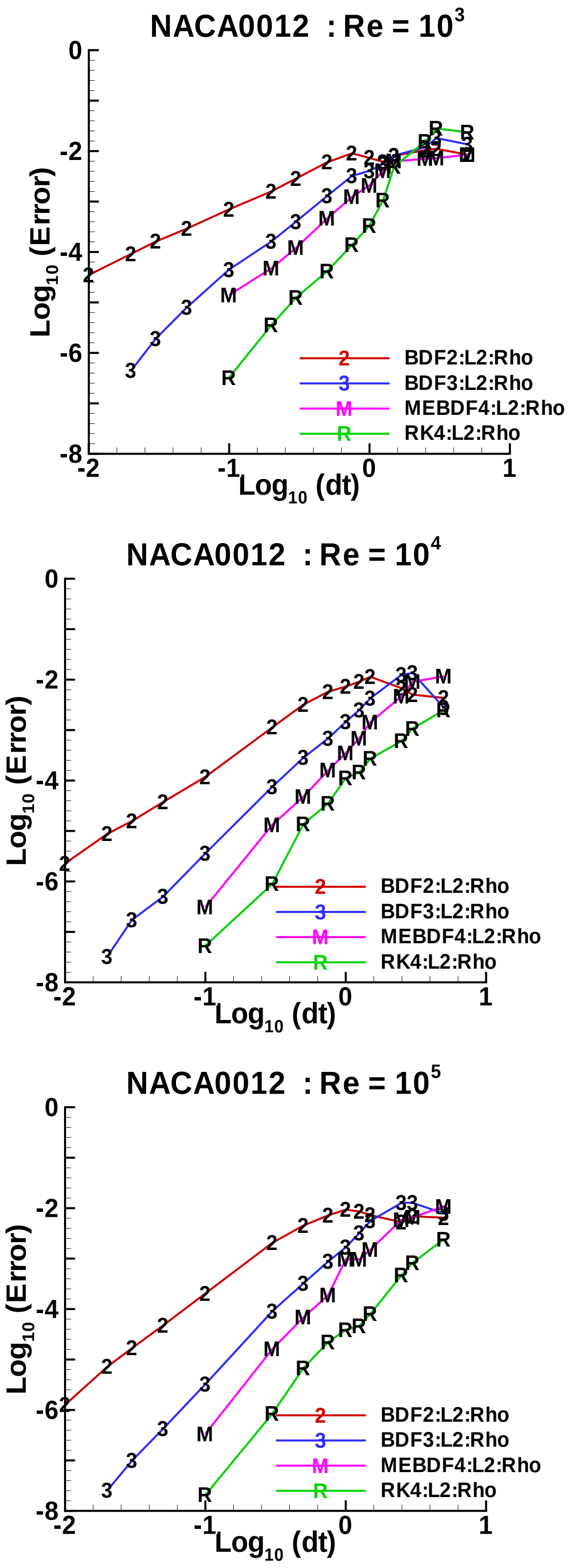

Figure 9. *Temporal convergence for the BDF2, BDF3, MEBDF4, and ESDIRK4 schemes on a NACA0012 airfoil at* $Re = 10^3,\ 10^4,\ 10^5$. *The Mach number is* $Ma = 0.3$.

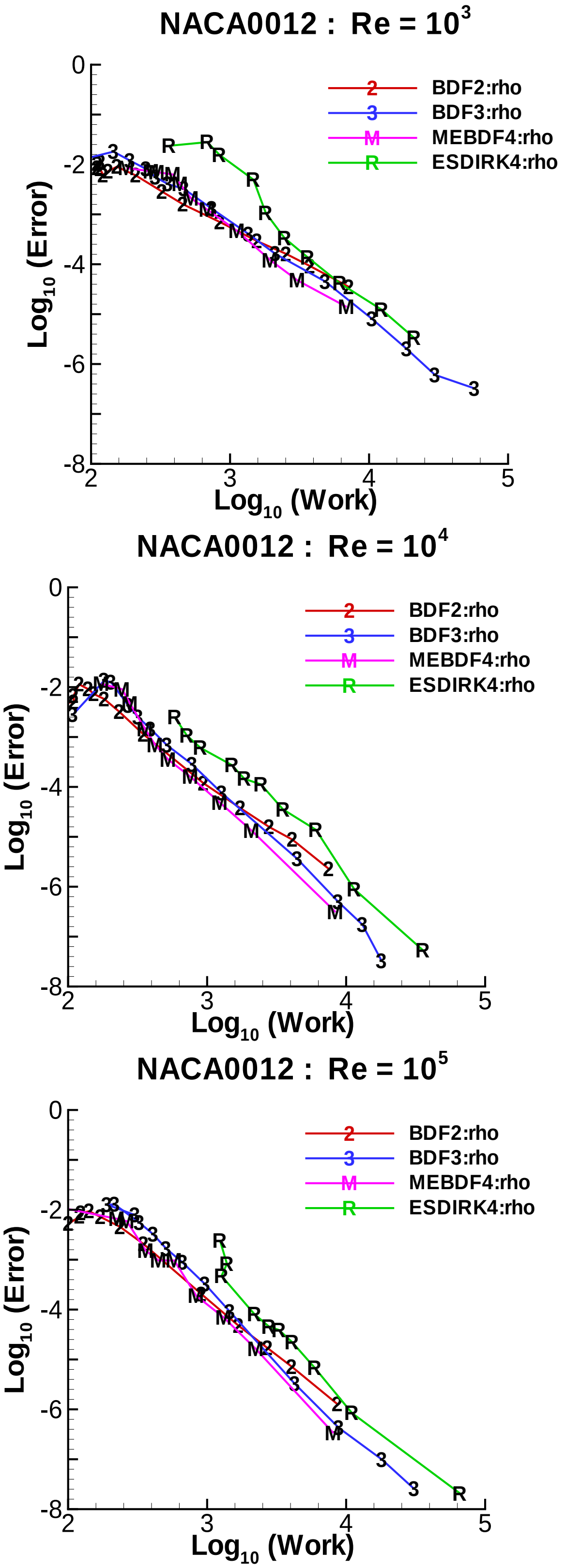

Figure 10. *Work-precision plots for the BDF2, BDF3, MEBDF4, and ESDIRK4 schemes on a NACA0012 airfoil at* $Re = 10^3,\ 10^4,\ 10^5$. *The Mach number is* $Ma = 0.3$.

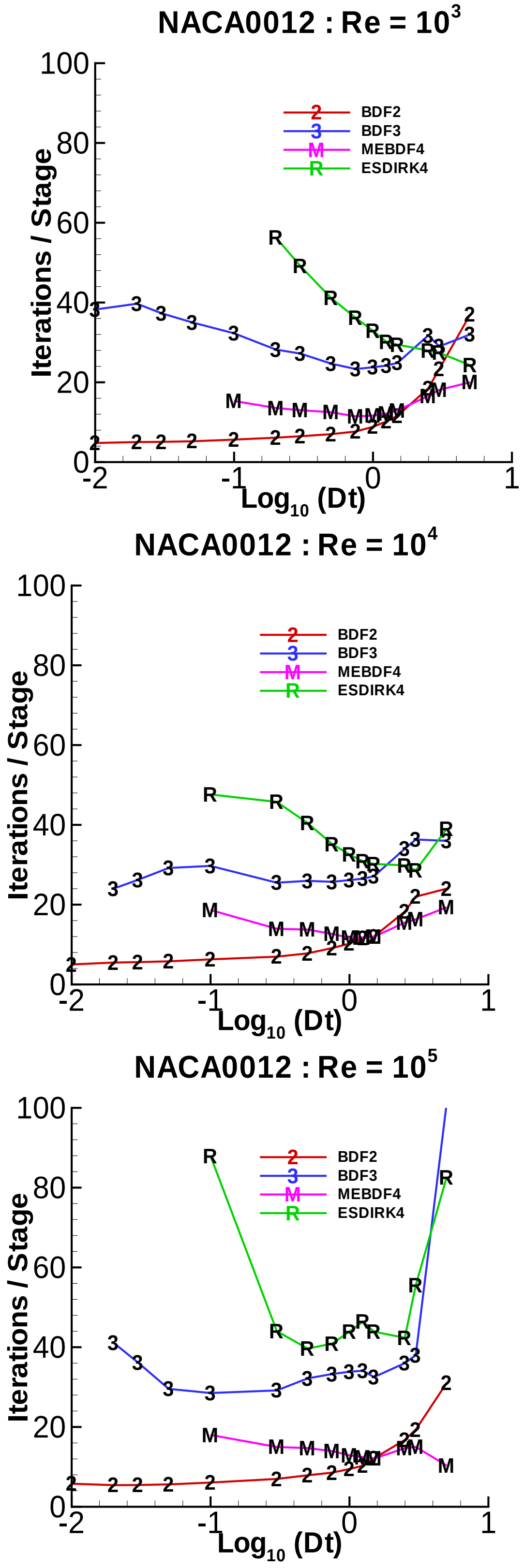

Figure 11. *Variations of iterations needed to reach termination tolerance for the BDF2, BDF3, MEBDF4, and ESDIRK4 schemes, as a function of time-step.*

in this study ($10^{-2} - 10^{-3}$). We speculate that other GLM methods will exhibit trends similar to the MEBDF4 and RK4 schemes if implemented into the FUN2D code. It is clear that in the constant time-stepping mode, significant gains in algebraic solver efficiency are needed if high-order solvers are to displace BDF2 in the FUN2D aerodynamics code.

The relative efficiency of each temporal algorithm is difficult to generalize to other computer codes. The overall efficiency of each scheme is strongly dependent on the proficiency and idiosyncrasies of the algebraic solver! Time-step independent algebraic convergence rates are unlikely on realistic problems in the near future. Convergences rates vary dramatically with solver technology. We speculate that strong solvers will favor high-order formulations, while weak solvers will favor existing low order formulations (BDF2). The optimal scheme for the development of an automated integration CFD package remains undetermined at present.

A fundamental obstacle in the FUN2D formulation is the unpredictable nature of the algebraic convergence observed with all high-order formulations. High-order schemes are plagued with poor convergence behavior at large time-steps and at low error tolerances. (Perhaps BDF2 would experience similar difficulties if high accuracy tolerances are sought.)

The red-black Gauss-Siedel inner-iteration strategy used in the FUN2D code is weak. Present developments in the HEFSS code include a multigrid algorithm, with a line implicit wall normal smoother[31]. Published results of Mavriplis [24, 25, 26, 27, 28], indicate that these algorithms are a notable improvement over the current FUN2D/FUN3D algebraic solver. Additional improvements to HEFSS include a tight coupling of the turbulence and fluid equations in the solution stage of the algorithm[31].

Future work on the HEFSS code will focus on achieving an iteration strategy that seldom fails on problems similar in type to those used in this study. Under-relaxation of large time-steps will be needed to improve convergence for high-order methods. Additional improvements for high-order schemes will come from better use of available data. High-order formulations must use reliable stage value predictors to provide accurate starting guesses for every iteration. For example, the MEBDF4 scheme has excellent stage value predictors for the first and third stages. The work required for stages one and three is usually less than that required for stage two in the FUN2D implementation of MEBDF4. The ESDIRK4 scheme currently does not have good stage value predictors. We conjecture that the ESDIRK4 scheme could improve in efficiency by a factor of two by developing good stage value predictors for use as the starting guess in each iteration. Good starting value guesses become even more crucial as larger time-steps are used. Improvements in iteration convergence rates as well as improvements in stage value predictors, will allow the HEFSS code to utilize the full potential of high-order schemes.

The ultimate objective for the HEFSS solver is automated control, designed around a variable time-stepping algorithm. A broad-based controller capable of detecting divergence or slow algebraic convergence will automatically adjust the time-step to minimize the effects of solver difficulties. The time-step will decrease if poor convergence is encountered to ensure that increasing time-steps never increase the work.

## Conclusions

A comparison of four temporal integration techniques is presented in the context of a general purpose aerodynamics solver. The study focuses on the temporal efficiency of high-order schemes, relative to the Backward Differentiation Formulae (BDF2) scheme. The high order algorithms used include the third-order BDF3 scheme, the fourth-order Modified Extended BDF (MEBDF4) scheme, and the fourth-order Explicit, Singly Diagonally Implicit Runge-Kutta (ESDIRK4) scheme.

Design order convergence is observed for all schemes. Specifically, second-, third-, and fourth-order accuracy for the BDF2, BDF3 and MEBDF4 schemes, while the ESDIRK4 scheme converged initially at a fourth-order rate but order reduced down to third-order at high precisions. Tests included simulation of turbulent flow around a circular cylinder and a NACA0012 airfoil at Reynolds numbers of $Re = 10^3, 10^4, 10^5$. Little sensitivity of the temporal algorithms to spatial grid resolution and Reynolds number is observed.

The work required to achieve engineering accuracies of two significant digits was surprisingly similar for all four temporal algorithms. With regards to work, very little advantage is observed in going to high-order schemes over the popular BDF2 scheme at accuracy tolerances of $10^{-3}$. At high-precision levels, for reasons of robustness and error estimation, the MEBDF4 scheme is a possible alternative to BDF2 in aerodynamic applications. Several reasons contributed to the poor showing of the the high-order schemes. The principle reason was poor convergence characteristic of the algebraic solver when using large time-steps.

Future work will focus on improving solver robustness, and implementing extrapolation/interpolation procedures to obtain better starting guesses for the algebraic solves.

## Acknowledgment

The authors would like to sincerely thank Dr. Chris A. Kennedy at Sandia National Laboratory, and Professor H. Bijl at Technical University of Delft for their helpful comments and insight.

## References

1. N. Alexandrov, S. Alter, H. Atkins, K. Bey, K. Bibb, R. Biedron, M. H. Carpenter, F. Cheatwood, P. Drummond, P. Gnoffo, W. Jones, W. Kleb, E. Lee-Rausch, R. Merski, R. Mineck, E. Nielsen, M. Park, S. Pirzadeh, T. Roberts, J. Samareb, C. Swanson, J. Thomas, V. Vatsa, J. Weilmuenster, J. White, W. Wood, and L. Yip, “Opportunities for Break-throughs in Large-Scale Computational Simulation and Design,” NASATM-2002-211747, April 2002.

2. W. K. Anderson, and D. L. Bonhaus, “An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids,” *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1-21.

3. W. K. Anderson, R. D. Rausch, and D. L. Bonhaus, “Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids,” *J. of Comp. Phys.*, Vol. 128, 1996, pp. 391-408.

4. H. Bijl, M. H. Carpenter and V. N. Vatsa, C. A. Kennedy, “Time Integration Schemes for the Unsteady Navier-Stokes Equations,” *J. of Comp. Phys.*, Vol. 179, 2002, pp. 313-329.

5. P. N. Brown, B. D. Byrne and A. C. Hindmarsh, “VODE: A Variable Coefficient ODE Solver,” *SIAM J. Sci. Stat. Comput.*, Vol. 10, 1989, 1038-1051.

6. J. C. Butcher, General Linear Methods for Stiff Differential Equations, *BIT*, Vol. 41, No. 2, 2001, pp. 240-264.

7. M. H. Carpenter, C. A. Kennedy, V. N. Vatsa, H. Bijl, “Implicit Runge-Kutta Schemes for Aerodynamic Applications: Order Reduction with Turbulence,” Still in preparation (2003).

8. J. R. Cash, “On the Integration of Stiff Systems of O.D.E.’s Using Extended Backward Differentiation Formulae,” *Numer. Math.*, Vol, 34, pp. 235-246.

9. J. R. Cash, "The Integration of Stiff Systems of O.D.E.'s Using Modified Extended Backward Differentiation Formulae," *Comp. and Maths. with Appls.*, Vol 9, No. 5, pp.645-657.

10. J. R. Cash, "A Comparison of Some Codes for the Stiff Oscillatory Problem", *Computers Math. Applic.*, Vol. 36, No. 1, pp. 51-57, 1998.

11. S. D. Cohen and A. C. Hindmarsh, "CVODE Users Guide, LLNL Report UCRL-MA-118618, Sept. 1994.

12. J. J. B. de Swart, W. M Lioen, W. A. van der Veen, "Specification of PSIDE," at http://www.cwi.nl/cwi/projects/PSIDE/.

13. W. H. Enright, T. E. Hull, and B. Lindberg, "Comparing Numerical methods for Stiff Systems of ODEs," *BIT*, Vol. 15, pp 10-48, (1975).

14. C. W. Gear, "Algorithm 407-DIFSUB for the Solution of Ordinary Differential Equations," *Commun. ACM*, Vol. 14, No. 3, 1971, pp. 185-190.

15. K. Gustafsson, and G. Söderlind, "Control Strategies for the Iterative Solution of Nonlinear Equations in ODE Solvers," *SIAM J. Sci. Comput.*, Vol. 18, No. 1, 1997, pp. 23-40.

16. E. Hairer S.P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, 2nd Ed., Springer-Verlag, Berlin (1993).

17. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, 2nd Ed., Springer-Verlag, Berlin (1996).

18. A. C. Hindmarsh, "LSODE and LSODI: Two new initial value ordinary differential equation solvers," *SIGNUM News*, Vol. 15, No. 10, 1980.

19. G. Jothiprasad, D. Mavriplis, D. Caughey, "Higher Order Time Integration Schemes for the Unsteady Navier-Stokes Equations on Unstructured Meshes," AIAA Paper 2002-2734.

20. C. A. Kennedy, M. H. Carpenter, and R. M. Lewis, "Low-Storage, explicit Runge-Kutta Schemes for the compressible Navier-Stokes equations," *Appl. Num. Math.*, 35(3), 177-219 (2000).

21. C. A. Kennedy and M. H. Carpenter, "Additive Runge-Kutta schemes for convection-diffusion-reaction equations," *Appl. Num. Math.*, 44(1-2): pp. 139-181, (2002).

22. W. L. Kleb, E. J. Nielsen, and P. A. Gnoffo, "Collaborative Software Development in Support of Fast Adaptive AeroSpace Tools (FAAST)," Submitted to 16th AIAA Computational Fluid Dynamics Conference.

23. 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).

24. D. J. Mavriplis, V. Venkatakrishnan, "A Unified Multigrid Solver for the Navier-Stokes Equations on mixed element meshes," *Int. J. Comput. Fluid Dyn.*, Vol. 8 1997, pp. 247-263.

25. D. J. Mavriplis, "Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes," *J. Comput. Phys.*, 145 1998, pp. 141-165.

26. D. J. Mavriplis, "On Convergence Acceleration Techniques for Unstructured Meshes," AIAA Paper 98-2966, 1998.

27. D. J. Mavriplis, "Directional Agglomeration Multigrid Techniques for High-Reynolds Number Viscous Flows," *AIAA J.*, 37 1999, pp. 1222-1230.

28. D. J. Mavriplis and S. Pirzadeh, "Large-Scale Parallel Unstructured Mesh Computations for 3D High-Lift Analysis," *AIAA J. Aircr*, 36 1999, pp. 987-998.

29. N. D. Melson, M. D. Sanetrik, H. L. Atkins, "Time-accurate Navier Stokes calculations with multigrid acceleration," *Proceedings of the Sixth Copper Mountain conference on multigrid methods, Copper Mountain, Colorado, April, 1993*, edited by Melson, Manteuffel, and McCormick, NASA conference publication 3224, Washington D.C., part 2, pp. 423-437.

30. F. Menter, "Improved Two-Equation k-w Turbulence Models for Aerodynamic Flows," 1992, NASA TM, 103975.

31. E. J. Nielsen, J. Lu, M. A. Park, D. L. Darmofal, "An Exact Dual Adjoint solution Method for Turbulent Flows on Unstructured Grids," AIAA Paper 2003-0272.

32. P. Roe, "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *J. of Comp. Phys.*, Vol. 43, 1981, pp. 357-372.

33. P. R. Spalart and S. R. Allmaras, "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-439 (1992).

34. P. R. Spalart, "Strategies for Turbulence Modeling and Simulations," 4th International Symp. on Engineering Turbulence Modeling and Measurements, 1999, Editor W. Rodi and D. Laurence, Elsevier, Corsica, France.

35. B. Van Leer, "Flux Vector Splitting for the Euler Equations," *Lecture Notes in Physics*, Vol. 170, 1982, pp. 501-512

36. J. G. Verwer, "Convergence and order reduction of diagonally implicit Runge-Kutta schemes in the method of lines," '*Numerical Analysis*, Eds. D. F., Griffiths and G. A. Watson, Res. Notes in Math. Ser. 140. Burnt Mill, Harlow, Essex, England: Longman Sci. and Tech.; NY (1986).

## Appendix

### ESDIRK4

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