NASA/CR-2002-211967
ICASE Report No. 2002-44

Higher Order Time Integration Schemes for the Unsteady Navier-Stokes Equations on Unstructured Meshes

Giridhar Jothiprasad
Cornell University, Ithaca, New York

Dimitri J. Mavriplis
ICASE, Hampton, Virginia

David A. Caughey
Cornell University, Ithaca, New York

November 2002

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATIONS. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized data bases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- Email your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Telephone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/CR-2002-211967
ICASE Report No. 2002-44

Higher Order Time Integration Schemes for the Unsteady Navier-Stokes Equations on Unstructured Meshes

Giridhar Jothiprasad
Cornell University, Ithaca, New York

Dimitri J. Mavriplis
ICASE, Hampton, Virginia

David A. Caughey
Cornell University, Ithaca, New York

ICASE, NASA Langley Research Center
Hampton, Virginia
Operated by Universities Space Research Association

Prepared for Langley Research Center
under Contract NAS1-97046

November 2002

Available from the following:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

HIGHER ORDER TIME INTEGRATION SCHEMES FOR THE UNSTEADY NAVIER-STOKES EQUATIONS ON UNSTRUCTURED MESHES

GIRIDHAR JOTHIPRASAD*, DIMITRI J. MAVRIPLIS†, AND DAVID A. CAUGHEY‡

Abstract. The efficiency gains obtained using higher-order implicit Runge-Kutta schemes as compared with the second-order accurate backward difference schemes for the unsteady Navier-Stokes equations are investigated. Three different algorithms for solving the nonlinear system of equations arising at each timestep are presented. The first algorithm (NMG) is a pseudo-time-stepping scheme which employs a non-linear full approximation storage (FAS) agglomeration multigrid method to accelerate convergence. The other two algorithms are based on Inexact Newton's methods. The linear system arising at each Newton step is solved using iterative/Krylov techniques and left preconditioning is used to accelerate convergence of the linear solvers. One of the methods(LMG) uses Richardson's iterative scheme for solving the linear system at each Newton step while the other (PGMRES) uses the Generalized Minimal Residual method. Results demonstrating the relative superiority of these Newton's method based schemes are presented. Efficiency gains as high as 10 are obtained by combining the higher-order time integration schemes with the more efficient nonlinear solvers.

Key words. Jacobian-free Newton, Runge-Kutta methods, Navier-Stokes, multigrid, unstructured grid

Subject classification. Applied and Numerical Mathematics

1. Introduction. The rapid increase in available computational power over the last decade has enabled higher resolution flow simulations and more widespread use of unstructured grid methods for complex geometries. While much of this effort has been focused on steady-state calculations in the aerodynamics community, the need to accurately predict off-design conditions, which may involve substantial amounts of flow separation, points to the need to efficiently simulate unsteady flow fields. Accurate unsteady flow simulations can easily require several orders of magnitude more computational effort than a corresponding steady-state simulation. For this reason, techniques for improving the efficiency of unsteady flow simulations are required if such calculations are to be feasible in the foreseeable future. The purpose of this work is to investigate possible reductions in computer time due to the choice of an efficient time-integration scheme from a series of schemes differing in the order of time-accuracy, and by the use of more efficient techniques to solve the nonlinear equations which arise while using implicit time-integration schemes. This investigation is carried out in the context of a two-dimensional unstructured mesh laminar Navier-Stokes solver.

Implicit in any comparison of efficiency is a precise error tolerance requirement. For stringent accuracy requirements, high-order temporal discretization schemes are well known to be superior to lower order (e.g. second-order) schemes, due to their superior asymptotic properties. However, for engineering calculations, where larger error tolerances ($O(10^{-2}) - O(10^{-3})$) are generally acceptable, second-order accurate time

*Graduate Student, Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853 (email: gj24@cornell.edu). This research was supported in part by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the author was in residence at ICASE, NASA Langley Research Center, Hampton, VA 23681.

†Research Fellow, ICASE, MS 132C, NASA Langley Research Center, Hampton, VA 23681 (email: dimitri@icase.edu). This research was supported in part by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the author was in residence at ICASE, NASA Langley Research Center, Hampton, VA 23681.

‡Professor, Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853 (email: dac5@cornell.edu)

discretizations are currently the method of choice, and higher-order methods are generally avoided due to their increased cost per time step. Recently, the use of higher-order accurate implicit Runge-Kutta schemes has been shown to produce efficiency gains even for relatively coarse error tolerances using a production structured-mesh Navier-Stokes solver [2].

In this paper, we perform a similar investigation within an unstructured mesh setting. Additionally, we investigate the efficiency of various non-linear solution techniques for solving the non-linear problems which arise at each time step for the various time discretizations considered. We consider three solution techniques, namely, a non-linear multigrid method which solves the non-linear problem directly through pseudo-time-stepping, and two variants of an inexact Newton scheme, where the linear system at each Newton iteration is partially solved using a linear multigrid scheme or a multigrid preconditioned GMRES approach. Because high-order time discretizations achieve high temporal accuracy with relatively large time steps, thus increasing the condition number of the non-linear problem, the use of efficient non-linear solvers takes on additional significance in such cases.

Non-linear multigrid methods were originally developed for steady-state fluid flow problems and subsequently adapted to unsteady flow problems [7, 13, 19]. Newton-based methods have often been avoided in this context due to the additional memory overheads incurred by such methods and the difficulties in providing reliable initializations for non-linear convergence. However, Newton-based methods have been shown to offer the potential for higher computational efficiency by avoiding frequent non-linear residual evaluations [12]. Furthermore, the disadvantages of Newton-based methods are less relevant in the context of an unsteady flow solver, where a close initial solution is always available from the previous time step, and where memory considerations are often secondary to cpu-time considerations.

In this paper, we illustrate the potential savings achieved using higher-order time discretizations and more efficient non-linear solvers for unsteady flow simulations on unstructured grids. We investigate the interaction between the time-discretization scheme and the non-linear solution technique as a function of temporal accuracy and show that the beneficial effects are multiplicative, producing up to an order of magnitude savings in computational effort.

2. Base Solver.

2.1. Spatial Discretization. For the purpose of comparison, an existing two-dimensional unstructured multigrid steady-state Navier-Stokes solver developed in [11] was modified to simulate transient flows by incorporating various physical time-stepping schemes. The flow equations are discretized using a finite-volume approach. Flow variables are stored at the vertices of the mesh, and control volumes are formed by the median-dual graph of the original mesh, as shown in Figure 2.1. A control-volume flux balance is computed by summing fluxes evaluated along the control volume faces, using the average values of the flow variables on either side of the face in the flux computation. The construction of convective terms corresponds to a second order accurate central difference scheme which requires additional dissipation terms for stability. These may either be constructed explicitly as a blend of a Laplacian and biharmonic operators, or may be obtained by writing the residual of a standard upwind scheme as the sum of a convective and dissipation term:

$$\sum_{k=1}^{\text{neighbors}} \frac{1}{2}\left\{\mathbf{F}\left(\mathbf{w_i}\right)+\mathbf{F}\left(\mathbf{w_k}\right)\right\}.\mathbf{n_{ik}}-\frac{1}{2}\mid \mathsf{A_{ik}} \mid \left(\mathbf{w_L}-\mathbf{w_R}\right) \tag{2.1}$$

where the convective fluxes are denoted by $\mathbf{F}(\mathbf{w})$, $\mathbf{n_{ik}}$ represents the normal vector of the control volume face separating the neighboring vertices i and k, and $\mathsf{A_{ik}}$ is the flux Jacobian evaluated in the direction normal to

this face. The variables $\mathbf{w_L}$ and $\mathbf{w_R}$ represent extrapolated flow properties at the left- and right-hand sides of the control volume face respectively. A matrix-based artificial dissipation scheme is obtained by utilizing the same transformation matrix $\mid \mathsf{A}_{ik} \mid$ as the upwind scheme, but using this to multiply a difference of blended first and second differences rather than a difference of reconstructed states at control-volume boundaries. For the calculations performed in this work, which involve only subsonic flows, the matrix dissipation formed using only second differences has been used exclusively, and the physical viscous terms for the Navier-Stokes equations are discretized to second-order accuracy using a finite-volume approximation.

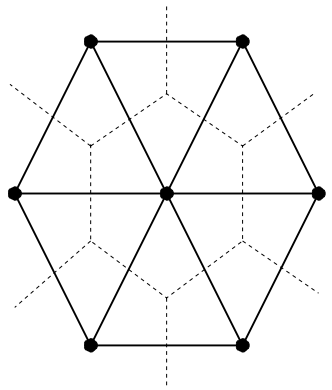

FIG. 2.1. *Median control-volumes for triangular meshes*

2.2. Temporal Discretization. Time is discretized in a fully implicit sense using both multistep Backward Difference Formulas (BDF) and multistage Runge-Kutta (RK) schemes. There are two mathematical properties that are desirable of a numerical integrator. The first is the "A-stability" property which guarantees that all eigenvalues lying in the left half of the complex plane will have an amplification factor no greater than unity, independent of the chosen step size. Hence, the only restriction on the time step with an A-stable scheme is the consideration of solution accuracy. The second is the "L-stable" property which guarantees that eigenvalues approaching $-\infty$ are damped in one time step.

Multi-step BDF formulas, and in particular the second-order accurate BDF scheme (BDF2), are widely used in the computation of large-scale engineering flows. These schemes require only one nonlinear set of equations to be solved at each time step. They suffer, however, from not being self-starting, are difficult to use with variable time steps, and are not A-stable beyond second-order temporal accuracy.

On the other hand, multistage RK schemes are self-starting, are easily implemented in a variable time-stepping mode, and can be designed with A- and L- stability properties for any temporal order. However, these schemes require multiple nonlinear solves at each time step, and hence have often been discounted as non-competitive compared to BDF schemes. One of the objectives of this paper is to investigate the relative efficiencies of BDF and RK schemes in computing time-dependent solutions to a given level of accuracy.

Consider the integration of the system of ordinary differential equations (ODEs) represented by the equation,

$$\frac{d\mathbf{w}}{dt} = \mathbf{S}\left(t, \mathbf{w}\left(t\right)\right) \tag{2.2}$$

where the vector $\mathbf{S}$ results from the spatial discretization of the equations of fluid mechanics. The general

formula for a k-step BDF scheme can be written as:

$$\mathbf{w}^{n+k} = -\sum_{i=0}^{k-1} \alpha_i \mathbf{w}^{n+i} + \Delta t \beta_k \mathbf{S}^{n+k} \tag{2.3}$$

BDF schemes require the storage of k+1 solution levels and the computation of one non-linear solution at each time step. For k=2, the second-order accurate (BDF2) scheme is obtained using the coefficients: $\alpha_0 = -4/3$, $\alpha_1 = 1/3$, $\beta_2 = 2/3$. More details of these standard schemes can be found in [2, 5].

Runge-Kutta methods are multistage schemes and are implemented as:

$$\mathbf{w}^{\{k\}} = \mathbf{w}^n + (\Delta t) \sum_{j=1}^{s} a_{kj} \mathbf{S}\left(\mathbf{w}^{\{j\}}\right), \text{k=1,2,...,s} \tag{2.4}$$

$$\mathbf{w}^{n+1} = \mathbf{w}^n + (\Delta t) \sum_{j=1}^{s} b_j \mathbf{S}\left(\mathbf{w}^{\{j\}}\right) \tag{2.5}$$

where s is the number of stages and a_{ij} and b_j are the Butcher coefficients of the scheme. Following the previous work by Bijl et al. [2] we focus on the ESDIRK class of RK schemes, which stands for Explicit first stage, Single diagonal coefficient, Diagonally Implicit Runge-Kutta. The Butcher table for a six stage ESDIRK scheme is shown in Table 2.1.

TABLE 2.1
Butcher Tableau for the ESDIRK class of RK schemes with number of stages, $s = 6$.

$c_1 = 0$	0	0	0	0	0	0
c_2	a_{21}	a_{66}	0	0	0	0
c_3	a_{31}	a_{32}	a_{66}	0	0	0
c_4	a_{41}	a_{42}	a_{43}	a_{66}	0	0
c_5	a_{51}	a_{52}	a_{53}	a_{54}	a_{66}	0
$c_6 = 1$	b_1	b_2	b_3	b_4	b_5	a_{66}
$\mathbf{w}^{n+1}$	b_1	b_2	b_3	b_4	b_5	a_{66}

In Table 2.1, c_k denotes the point in the time interval, $[t, t + \Delta t]$. These schemes are characterized by a lower triangular form of the coefficient table, thus resulting in a single implicit solve at each individual stage. The first stage is explicit ($a_{k1} = 0$) and the last stage coefficients take on the form $a_{kj} = b_j$, thus enabling equation (5) to be simplified as

$$\mathbf{w}^{n+1} = \mathbf{w}^{\{k=s\}} \tag{2.6}$$

We use the following notation, RKxy refers to an ESDIRK scheme which has x stages and y^{th} order accuracy. Bijl et al. [2] have compared these schemes and found RK64 to perform well. The numerical values for the coefficients of this scheme are given in the Appendix. More details in general about ESDIRK schemes can be found in [8].

3. Implicit Solution Technique. Both BDF and RKxy schemes require the solution of a nonlinear system of equations. BDF schemes require the solution of one nonlinear equation per time step. In the case of BDF, a nonlinear residual, $\mathbf{R}(\mathbf{w})$, can be defined from equation (2.3) and is given by:

$$\begin{aligned} \mathbf{R}(\mathbf{w}) &\equiv \mathbf{R}\left(\mathbf{w}^{n+k}\right) \\ &\equiv \frac{\mathbf{w}^{n+k}}{\Delta t} - \beta_k \mathbf{S}^{n+k} + \mathbf{SRC}_{\mathbf{BDF}} \end{aligned} \tag{3.1}$$

where the superscript on $\mathbf{w}$ has been dropped for the sake of simplicity. $\mathbf{SRC_{BDF}}$ is the source term independent of $\mathbf{w} \equiv \mathbf{w}^{n+k}$ and is given by,

$$\mathbf{SRC_{BDF}} \equiv \frac{1}{\Delta t}\left[\sum_{i=0}^{k-1} \alpha_i \mathbf{w}^{n+i}\right] \tag{3.2}$$

In the case of RKxy schemes, a nonlinear system arises at each stage of the time-stepping scheme and hence more than one nonlinear solve per time step is required. Again, a nonlinear residual, $\mathbf{R}(\mathbf{w})$ for each stage of the RKxy scheme can be defined using equation (2.4) as follows:

$$\begin{aligned} \mathbf{R}(\mathbf{w}) &\equiv \mathbf{R}\left(\mathbf{w}^{\{k\}}\right) \\ &\equiv \frac{\mathbf{w}^{\{k\}}}{\Delta t} - a_{kk}\mathbf{S}\left(\mathbf{w}^{\{k\}}\right) + \mathbf{SRC_{RK}} \end{aligned} \tag{3.3}$$

where the superscript on $\mathbf{w}$ has again been dropped for the sake of simplicity. Also, $\mathbf{SRC_{RK}}$, is the source term independent of $\mathbf{w} \equiv \mathbf{w}^{\{k\}}$ and is given by,

$$\mathbf{SRC_{RK}} \equiv -\frac{\mathbf{w}^n}{\Delta t} - \sum_{j=1}^{k-1} a_{kj}\mathbf{S}\left(\mathbf{w}^{\{j\}}\right) \tag{3.4}$$

Hence, in both BDF and RKxy we are required to obtain the solution of the nonlinear system of equations,

$$\mathbf{R}(\mathbf{w}) = 0 \tag{3.5}$$

Three different methods are proposed for solving equation (3.5) and their relative performances are studied. The three methods, in this paper, are henceforth referred to as :

1. Nonlinear Multigrid (NMG)
2. Linear Multigrid (LMG)
3. Preconditioned Generalized Minimal Residual (PGMRES)

In NMG, a pseudo-time-stepping scheme is employed to obtain the solution of the nonlinear system of equations, which is accelerated using a non-linear full approximation storage (FAS) agglomeration multigrid method [11, 12]

In the other two approaches, an inexact Newton solution strategy is used to solve the nonlinear system of equations [14, 11, 15]. The resulting linear system of equations is solved using iterative/Krylov techniques. To accelerate convergence, the linear system is left preconditioned using an approximate inverse to the first-order accurate Jacobian which in itself is employed as an approximation to the Jacobian of the second-order accurate discretization. The last two approaches differ only in the methods used to solve the preconditioned linear system of equations. LMG uses the Richardson's iterative method while PGMRES uses the Generalized Minimal Residual method developed by Yaad and Schultz [16].

3.1. NonLinear Multigrid (NMG). In a pseudo-time stepping scheme, the equations are integrated in pseudo-time until Eq. (3.5) is satisfied.

$$\frac{d\mathbf{w}}{dt^*} + \mathbf{R}(\mathbf{w}) = 0 \tag{3.6}$$

where t^* is the pseudo-time. Since we do not require a pseudo-time accurate solution, the equations are preconditioned to accelerate the convergence. Hence, we have,

$$\mathsf{P}^{-1}\frac{d\mathbf{w}}{dt^*} + \mathbf{R}(\mathbf{w}) = 0 \tag{3.7}$$

The equations are integrated in pseudo-time using an explicit preconditioned multi-stage scheme [10], which can be written as :

$$\begin{aligned}
\mathbf{w}^{(0)} &= \mathbf{w}^{[k]} \\
\mathbf{w}^{(1)} &= \mathbf{w}^{(0)} - \Delta t^* \mathsf{P}\mathbf{R}\left(\mathbf{w}^{(0)}\right) \\
&\vdots \\
\mathbf{w}^{(q)} &= \mathbf{w}^{(q-1)} - \Delta t^* \mathsf{P}\mathbf{R}\left(\mathbf{w}^{(q-1)}\right) \\
\mathbf{w}^{[k+1]} &= \mathbf{w}^{(q)}
\end{aligned} \tag{3.8}$$

for a q-stage scheme.

An agglomeration multigrid strategy previously developed as a solver for steady-state problems [10] is used to further accelerate convergence. The agglomeration multigrid may be viewed as a simplified algebraic multigrid strategy. Coarse level grids are constructed by fusing together or agglomerating neighbouring control volumes to form a coarser set of larger but more complex control volumes. In the algebraic interpretation of agglomeration multigrid, the coarse levels are no longer geometric grids, but represent groupings of fine grid equations which are summed together to form the coarse grid equations sets [9, 6] The basic smoother on each grid level is a three-stage explicit preconditioned multi-stage scheme with stage coefficients optimized for high frequency damping properties [1].

The preconditioner, P is chosen to be the block diagonal of the Jacobian of the residual, $\mathbf{R}(\mathbf{w})$,

$$\mathsf{P}_{ii}^{-1} = \mathsf{D}_{ii} = \left[\frac{\partial \mathbf{R_i}}{\partial \mathbf{w_i}}\right]_{\mathbf{w}^{(0)}} \tag{3.9}$$

This is referred to as Jacobi preconditioning and provides substantial increases in efficiency when upwind or matrix dissipation discretizations are used [18]. It should be noted that the use of Jacobi preconditioning involves inverting a 4×4 matrix for each vertex at each stage.

3.2. Inexact Newton's Methods. To solve the non-linear system of equations $\mathbf{R}(\mathbf{w}) = 0$, Newton's method requires the solution of a series of linear systems of the form,

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{w}}\right]_{\mathbf{w}^{[k]}} \delta \mathbf{w}^{[k]} = -\mathbf{R}\left(\mathbf{w}^{[k]}\right) \tag{3.10}$$

where,

$$\mathbf{w}^{[k+1]} = \mathbf{w}^{[k]} + \delta \mathbf{w}^{[k]} \tag{3.11}$$

Let,

$$\mathbf{x} \equiv \delta \mathbf{w}^{[k]}; \qquad \mathbf{r} \equiv \mathbf{R}\left(\mathbf{w}^{[k]}\right); \qquad \mathsf{A} \equiv \left[\frac{\partial \mathbf{R}}{\partial \mathbf{w}}\right]_{\mathbf{w}^{[k]}} \tag{3.12}$$

Hence, equation (3.10) now becomes,

$$\mathsf{A}\mathbf{x} = -\mathbf{r} \tag{3.13}$$

Traditionally, there have been two main obstacles to the use of Newton's method for large-scale multi-physics applications :

1. *An initial guess inside of the radius of convergence is required for Newton's method to converge.* However, for unsteady problems, a good initial guess is provided by the solution at the previous time step. If the Newton's method does not converge, then by lowering the time step one can get the initial guess as close as necessary to the solution at the next time level. In the calculations presented in this paper, no difficulties were encountered in the convergence of the Newton iterations for any of the time steps used.
2. *Construction and storage of the Jacobian matrix,* A *becomes prohibitive.* This problem is particularly exacerbated in 3D for higher-order spatial discretizations which are not confined to the nearest neighbor stencils. This problem is overcome by the use of Jacobian-free methods to solve the linear system of equations. However, additional memory is still required to store the first-order Jacobian for the preconditioning operation, and the various Krylov vectors for the GMRES scheme. On the other hand, the use of additional memory can be rationalized if this produces substantial gains in CPU time, particularly for unsteady flow simulations where cpu time is the dominant concern.

Additionally, in order to improve the computational efficiency of these methods, we use an inexact Newton's method[14, 3] where the arising system of equations are not solved exactly. In this paper, we employ a very simplistic method where the number of iterations carried out by the underlying iterative linear solver is held fixed. In it's exact form, Newton's method provides quadratic convergence. However due to the various approximations employed by the solution method in this paper, this rate of convergence is not achieved.

3.3. Preconditioned Inexact Newton's Method. In order to achieve rapid convergence of the linear problem at each Newton iteration, preconditioning methods are used to cluster the eigenvalues of the system. We adopt the approach of left preconditioning in order to achieve this desirable distribution of eigenvalues. The preconditioned system can be written as:

$$\mathcal{P}_{\mathcal{N}}\mathsf{A}\mathbf{x} = -\mathcal{P}_{\mathcal{N}}\mathbf{r} \tag{3.14}$$

We make the following comments on the preconditioning:

- The preconditioner, $\mathcal{P}_{\mathcal{N}}$, is looked upon here as an operator as opposed to a matrix. Hence, $\mathcal{P}_{\mathcal{N}}$, may or may not be able to be written as a matrix.
- The preconditioner must be chosen as close as possible to A^{-1} so that $\mathcal{P}_{\mathcal{N}}\mathsf{A} \approx \mathcal{I}$, where $\mathcal{I}$ is the identity operator.
- Since each step of the Newton's method, equation (3.10), moves $\mathbf{w}^{[k]}$, towards the solution, $\mathbf{w}^*$, of $\mathbf{R}(\mathbf{w}) = 0$, any operator which produces a correction $\delta\mathbf{w}^{[k]} = \mathbf{x}$ to advance $\mathbf{w}^{[k]}$ towards $\mathbf{w}^*$ would serve as a reasonable preconditioner.
- Based on the above fact, single or multiple cycles of the nonlinear multigrid method discussed in the earlier subsection could be used as a preconditioner. However, this is computationally inefficient as it does not recognize the fact that we are now solving a system of linear equations.

Keeping in mind the above observations we now propose a better preconditioner, $\mathcal{P}_{\mathcal{N}}$. We first choose $\mathcal{P}_{\mathcal{N}} = \widetilde{\mathsf{A}}^{-1}$, where $\widetilde{\mathsf{A}}$ is a matrix approximation to the Jacobian, A. Considerations governing the choice of $\widetilde{\mathsf{A}}$ would be,

1. *Storage requirements for* $\widetilde{\mathsf{A}}$ *must not be prohibitive.* Storage for $\widetilde{\mathsf{A}}$ must use less space than that needed for A or else there would be no space gain in using Jacobian-free methods.
2. *The inverse of* $\widetilde{\mathsf{A}}$ *must be simple to calculate or approximate.* If one is able to compute an approximate inverse to $\widetilde{\mathsf{A}}$ fairly easily using iterative methods, this new operator, $\mathcal{P}_{\mathcal{N}} = \widetilde{\widetilde{\mathcal{A}}}^{-1}$ would serve as

an appropriate preconditioner. The two tildes are used to symbolize the fact that there are two approximations involved in the definition of the $\mathcal{P}_{\mathcal{N}}$;

(a) $\widetilde{\mathsf{A}}$ which is an approximation to the Jacobian, A

(b) An approximation in computing the inverse of $\widetilde{\mathsf{A}}$

In this paper, we choose $\widetilde{\mathsf{A}}$ to be the Jacobian of the first-order accurate, nearest neighbor discretization of the nonlinear set of equations. Hence, the storage of $\widetilde{\mathsf{A}}$ requires substantially less memory than that of the full Jacobian A of the second-order accurate scheme.

$\widetilde{\mathsf{A}}^{-1}$ is computed approximately using a fixed number of W-cycles of a linear multigrid method. This particular multigrid method can be viewed as the linear analogue of the non-linear FAS agglomeration multigrid scheme described in the NMG method. This approach has been previously described in detail in reference [12]. In this particular approach the coarse level approximations to the Jacobian are obtained taking the Jacobian of the Galerkin projection of the (frozen) fine grid operator as:

$$\mathsf{A}_{\mathsf{H}} = \frac{\partial}{\partial w_H}\left(\mathsf{I}_{\mathsf{h}}^{\mathsf{H}}\mathbf{R}_{\mathbf{h}}\mathsf{I}_{\mathsf{H}}^{\mathsf{h}}\right) \tag{3.15}$$

as opposed to the more traditional linear multigrid Galerkin projection:

$$\mathsf{A}_{\mathsf{H}} = \mathsf{I}_{\mathsf{h}}^{\mathsf{H}}\frac{\partial \mathbf{R}_{\mathbf{h}}}{\partial w_H}\mathsf{I}_{\mathsf{H}}^{\mathsf{h}} = \mathsf{I}_{\mathsf{h}}^{\mathsf{H}}\mathsf{A}_{\mathsf{h}}\mathsf{I}_{\mathsf{H}}^{\mathsf{h}} \tag{3.16}$$

where $\mathsf{I}_{\mathsf{h}}^{\mathsf{H}}$ is the restriction operator and $\mathsf{I}_{\mathsf{H}}^{\mathsf{h}}$ is the prolongation operator. This approach was chosen purely for convenience, as the terms in equation (3.15) are readily available. The smoother on each grid was taken as a block diagonal Jacobi solver.

3.4. Linear Multigrid (LMG). In the method referred to as LMG, the linear multigrid preconditioned system arising at each Newton iteration is solved using a single Richardson iteration. In order to solve the system,

$$\mathcal{P}_{\mathcal{N}}\mathsf{A}\mathbf{x} = -\mathcal{P}_{\mathcal{N}}\mathbf{r} \tag{3.17}$$

we define the splitting [4]

$$\mathcal{P}_{\mathcal{N}}\mathsf{A} = \mathcal{I} + \mathcal{N} \tag{3.18}$$

The resulting iterative scheme is defined as,

$$\begin{aligned} \mathcal{I}\mathbf{x}^{(m+1)} &= -\mathcal{P}_{\mathcal{N}}\mathbf{r} - \mathcal{N}\mathbf{x}^{(m)} \\ \mathcal{I}\delta\mathbf{x}^{(m)} &= -\mathcal{P}_{\mathcal{N}}\mathbf{r} - \mathcal{P}_{\mathcal{N}}\mathsf{A}\mathbf{x}^{(m)} \end{aligned} \tag{3.19}$$

As indicated earlier, since we are required to solve equation (3.17) only approximately, we carry out only a single Richardson's iteration. Assuming $\mathbf{x}^{(1)} = \mathbf{0}$, equation (3.19) reduces to,

$$\begin{aligned} \delta\mathbf{x}^{(1)} &= -\mathcal{P}_{\mathcal{N}}\mathbf{r} \\ \mathbf{x}^{(2)} &= \mathbf{x}^{(1)} + \delta\mathbf{x}^{(1)} = \delta\mathbf{x}^{(1)} = -\mathcal{P}_{\mathcal{N}}\mathbf{r} \end{aligned} \tag{3.20}$$

Hence, we have

$$\delta\mathbf{w}^{[k]} = \mathbf{x}^{(2)} = -\mathcal{P}_{\mathcal{N}}\mathbf{r} \tag{3.21}$$

Equation (3.21) illustrates the correspondence of this scheme to a Newton's method in which the first-order accurate Jacobian is used along with a second-order accurate residual.

3.5. Preconditioned Generalized Minimal Residual (PGMRES). Having presented the LMG scheme as a preconditioned Richardson iteration, the PGMRES scheme can similarly be described as the equivalent scheme obtained when the single Richardson iteration is replaced by a GMRES Krylov subspace iterative approach [17]. In this method, we use GMRES to solve equation (3.14) in a matrix-free Newton-Krylov fashion [20], making use of the same preconditioner, $\mathcal{P}_{\mathcal{N}}$, which is used in the linear multigrid method (LMG). The matrix-free implementation of PGMRES requires the computation of the product, $\mathcal{P}_{\mathcal{N}}\mathsf{A}\mathbf{x}$, which is approximated using a first-order Taylor series expansion as,

$$\begin{aligned} \mathcal{P}_{\mathcal{N}}\mathsf{A}\mathbf{x} &= \frac{\mathcal{P}_{\mathcal{N}}\mathbf{R}\left(\mathbf{w}^{[k]}+\epsilon\mathbf{x}\right)-\mathcal{P}_{\mathcal{N}}\mathbf{R}\left(\mathbf{w}^{[k]}\right)}{\epsilon} \\ &= \frac{\mathcal{P}_{\mathcal{N}}\mathbf{R}\left(\mathbf{w}^{[k]}+\epsilon\mathbf{x}\right)-\mathcal{P}_{\mathcal{N}}\mathbf{r}}{\epsilon} \end{aligned} \tag{3.22}$$

where $\mathbf{x}$ is some unit vector and ϵ is a number chosen close to machine round-off. We use the restarted form of GMRES with a fixed number of search directions. While increasing the number of Krylov vectors accelerates convergence, storage and cpu time increase with the number of search directions. The optimal number of search directions is therefore determined experimentally.

4. Validation of the Temporal Scheme. Numerical experiments have been performed to determine the observed order of accuracy of the various time-integration schemes. The test problem chosen for this purpose consists of the unsteady laminar flow around a two-dimensional circular cylinder at a Reynolds number of 1200 and a Mach number of 0.2. The initial flow is symmetric with zero lift. As the wake behind the cylinder starts to grow, it becomes unstable and begins to shed vortices from alternate sides of the cylinder. The computational grid is shown in Figure 4.1. The far-field boundary is a circle concentric with the cylinder and diameter given by $40D$ where D is the diameter of the cylinder. A close up of the mesh around the cylinder is shown in Figure 4.2

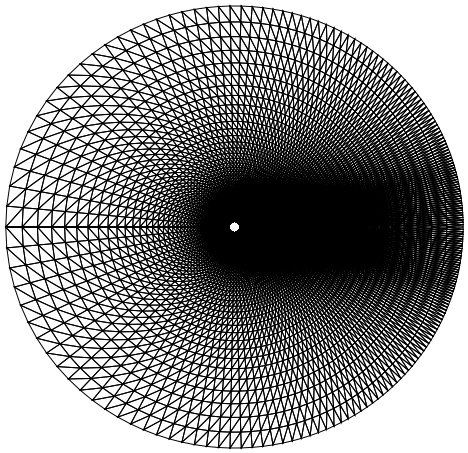

FIG. 4.1. *Computational mesh for circular cylinder*

Time is non dimensionalized as Ut/D where U is the free stream velocity and D is the diameter of the cylinder. The initial condition for the various studies was obtained by simulating the limit cycle behavior

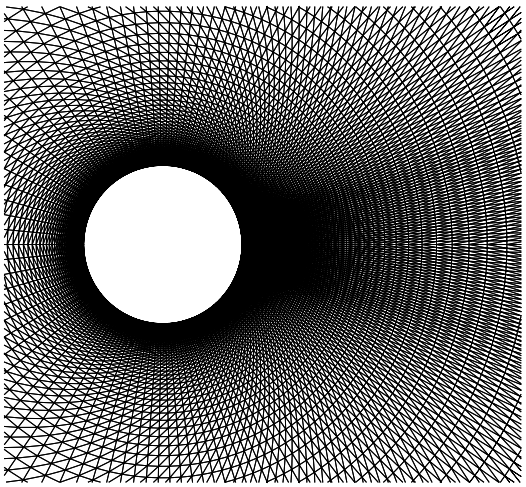

FIG. 4.2. *Computational mesh in the region around the cylinder*

for about 10 shedding cycles using a relatively small time step. Using $\Delta t = 0.025$, the Strouhal number was calculated to be 0.2469. The variation of C_L with time is shown in Figure 4.3. A plot of the density contours at an intermediate time is shown in Figure 4.4.

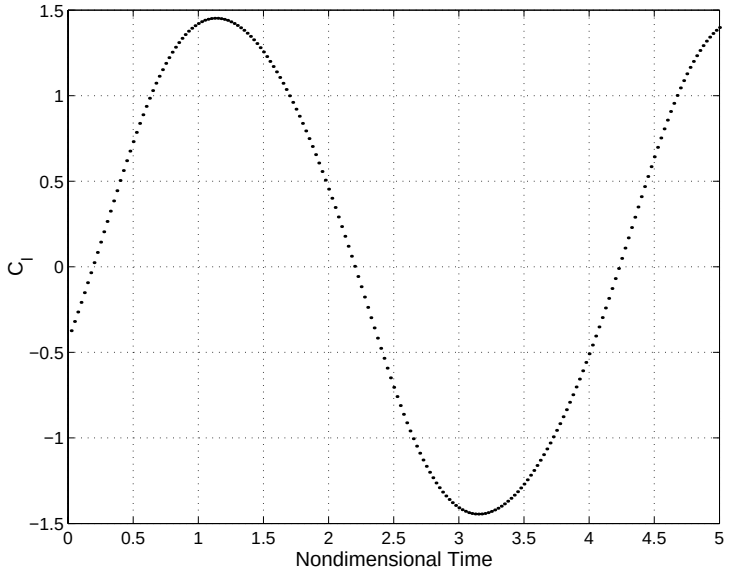

FIG. 4.3. *Variation of C_L with non dimensional time*

In order to determine the temporal order of accuracy, the test problem was solved using the same initial condition but with different time steps. The time interval of the study was approximately $1\frac{1}{4}$ shedding cycles. The solution at the end of the time interval is assumed to have accumulated the temporal error. Integral measures such as lift on the body, drag due to pressure forces and pitching moment of the body were then compared as follows to determine the order of accuracy. Let $G_{\Delta t}$ denote the integral measure being compared using a time step Δt, while G_{exact} denotes the exact solution. We do not know G_{exact} but based on the order of accuracy n of the scheme, we expect the following behavior:

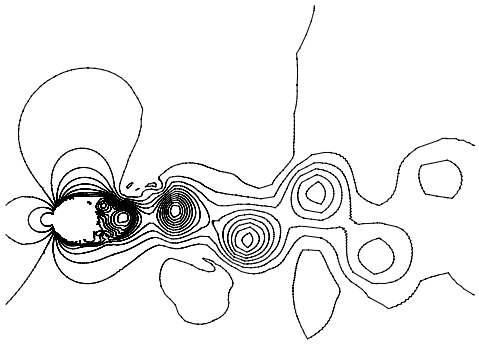

FIG. 4.4. *Density Contours calculated using a time step of* 0.025

$$G_{\Delta t} = G_{\text{exact}} + C_1 \left(\Delta t\right)^n + \text{Higher Order Terms} \tag{4.1}$$

where C_1 is a constant. By subtracting from equation (4.1) a similar expression for $G_{\frac{\Delta t}{2}}$ and neglecting the higher-order terms, we can obtain the following relation,

$$\begin{aligned} G_{\Delta t} - G_{\frac{\Delta t}{2}} &= C_1 \left(1 + \frac{1}{2^n}\right) \left(\Delta t\right)^n \\ &= C_2 \left(\Delta t\right)^n \end{aligned} \tag{4.2}$$

where C_2 is another constant. Equation (4.2) can be used to determine the order of accuracy of the scheme which can then be compared with the expected order of accuracy based on theory.

The order of accuracy was verified for two ESDIRK schemes (RK64 and RK43) and the second-order BDF (BDF2) scheme. The nonlinear systems which arose were converged until the maximum density correction, $\mid \Delta\rho \mid_{\max} \; < \; 10^{-10}$. This ensures that the "iteration error" is negligibly small relative to the discretization error. Figures 4.5, 4.6 and 4.7 show the detailed refinement study for the RK64, RK43 and BDF2 schemes respectively. It can be seen that all the integral measures yield nearly the same quantitative conclusions. The anomalous behavior for large time steps in Figure 4.7 is likely due to the timesteps being outside the asymptotic range. It is also seen that the computed order of accuracy is close to the expected order of accuracy. For example, Figure 4.5 shows that the computed order of accuracy for the RK64 scheme is 3.8938 while the expected order of accuracy is 4.

5. Parameter Selection in the Linear Multigrid method (LMG) . The inexact Newton methods contain various parameters which must be chosen judiciously in order to optimize the overall run-time of these methods. For the LMG scheme, the parameters to be chosen include:

1. Number of linear multigrid cycles carried out in $\mathcal{P}_{\mathcal{N}}$
2. Number of smoothing iterations carried out on each grid of the multigrid.

Increasing either of these parameters would make $\mathcal{P}_{\mathcal{N}}$ a better approximation to $\widetilde{\mathcal{A}}^{-1}$. Figure 5.1 illustrates the effect of increasing the number of linear multigrid cycles used in $\mathcal{P}_{\mathcal{N}}$. All the results shown in this section and the next are carried out using the BDF2 physical time stepping scheme and a timestep of $\Delta t = 0.05$.

We observe the following :

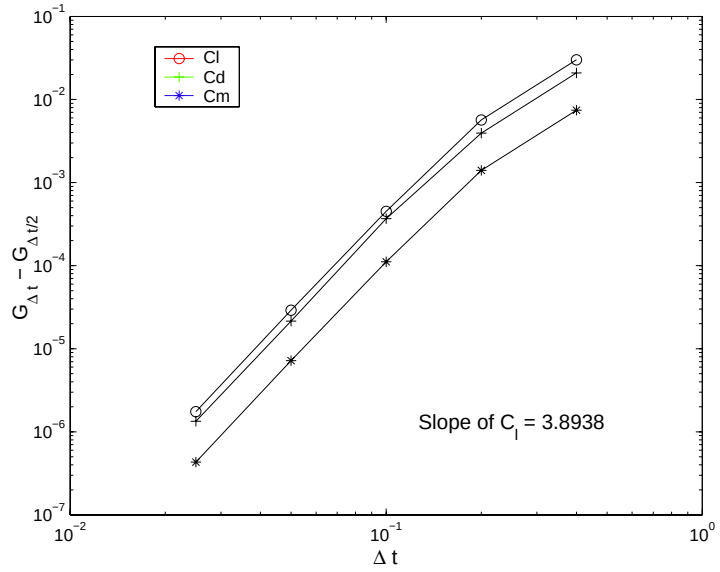

FIG. 4.5. *Verifying order of accuracy of RK*64

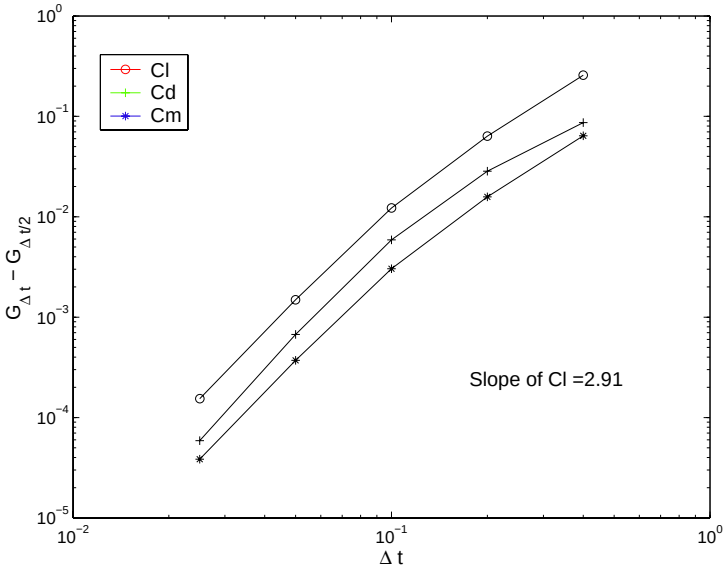

FIG. 4.6. *Verifying order of accuracy of RK*43

1. The rate of convergence slows down as the non-linear residual decreases. This is due to the inexact solution of the linear system described by equation (3.14) at each Newton iteration, as we use only a single Richardson's iteration to solve the equation. Hence, the quadratic convergence expected of exact Newton's methods is not achieved.
2. Increasing the number of linear multigrid cycles in $\mathcal{P}_{\mathcal{N}}$ can increase the non-linear convergence by only a finite amount, and eventually asymptotes to a maximum rate. This is due to the fact the preconditioner $\mathcal{P}_{\mathcal{N}}$ is based on a first-order Jacobian, which even if inverted exactly does not correspond to the inverse of the exact Jacobian, A

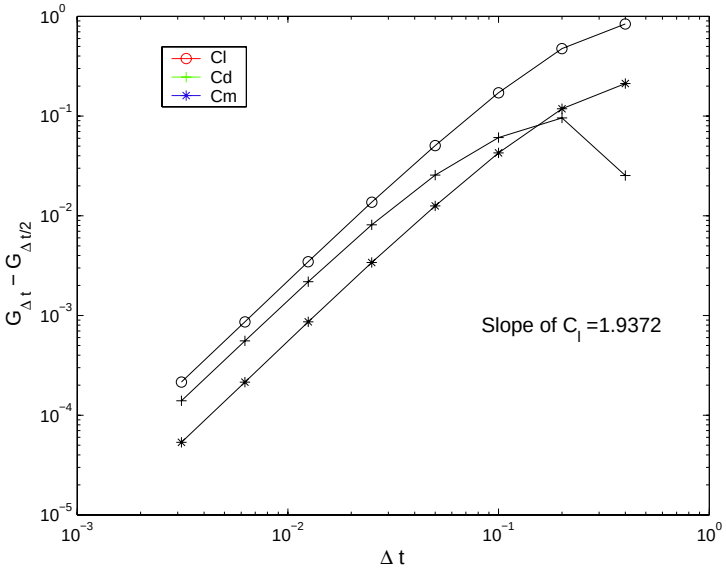

FIG. 4.7. *Verifying order of accuracy of BDF2*

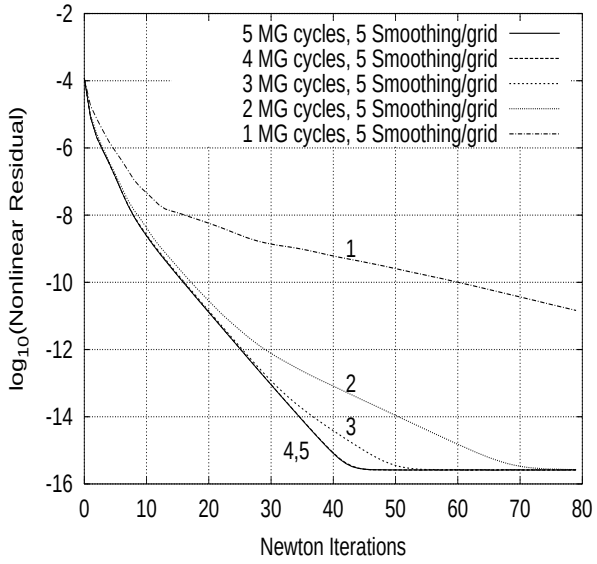

FIG. 5.1. *Effect of changing the number of linear multigrid cycles in $\mathcal{P}_{\mathcal{N}}$ of LMG*

Since, the ultimate goal is to minimize the runtime required to converge the equations, $\mathbf{R}(\mathbf{w}) = \mathbf{0}$, to a given tolerance level, we plot the reduction of the residual against runtime for different choices of the number of linear multigrid cycles. It can be seen from Figure 5.2 that the use of 2 linear multigrid cycles in $\mathcal{P}_{\mathcal{N}}$ is computationally most efficient for the range of error tolerances to which the nonlinear equations are converged.

Figures 5.3 and 5.4 show the effect of performing different number of smoothing iterations on each grid of the multigrid. As expected, Figure 5.3 shows that the number of Newton iterations required for a given level of residual reduction decreases with increase in the number of smoothing iterations.

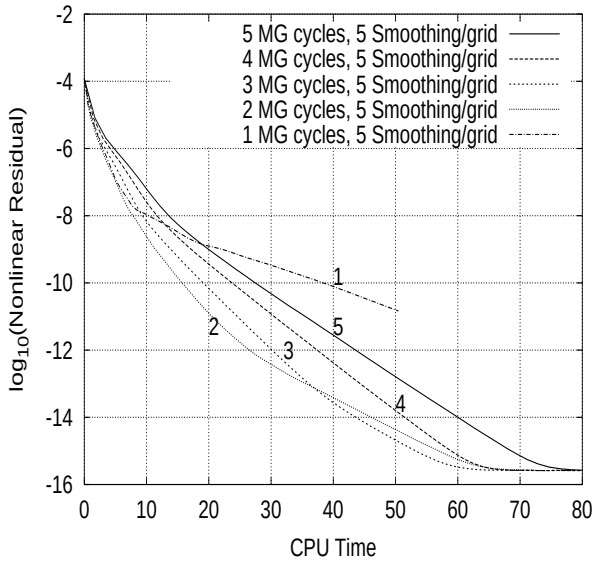

FIG. 5.2. *Effect of changing the number of linear multigrid cycles in $\mathcal{P}_{\mathcal{N}}$ of LMG on the runtime*

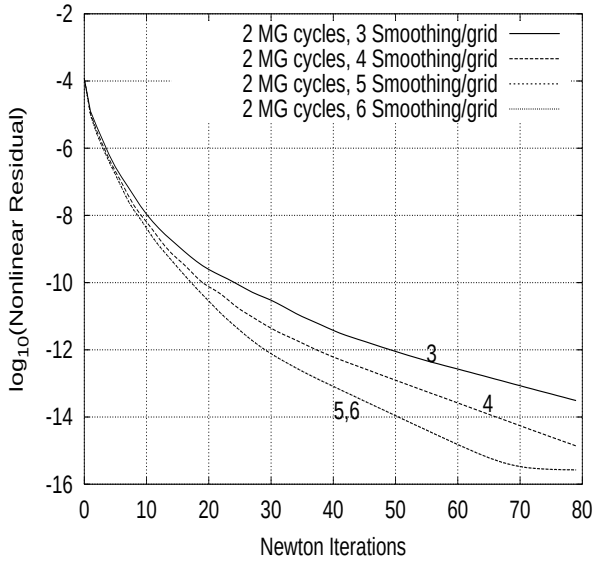

FIG. 5.3. *Effect of changing the number of smoothing iterations carried out on each grid of the multigrid*

The study indicates that the use of 5 smoothing cycles on each grid, while using two linear multigrid cycles, results in a computationally optimal preconditioner, $\mathcal{P}_{\mathcal{N}}$, for the problem under consideration.

6. Parameter Selection in PGMRES. Having selected all required parameters for the linear multigrid preconditioner, $\mathcal{P}_{\mathcal{N}}$, the remaining parameter to be determined in the PGMRES method is the number of search directions. Increasing the number of search directions provides a more accurate solution to the linear system, which arises at every step of the Newton's method. Thus, the Inexact Newton's method approaches the Exact Newton's method as the number of search directions is increased.

Figure 6.1 shows the effect of increasing the number of search directions on the number of Newton iterations required to converge to the solution of $\mathbf{R}(\mathbf{w}) = \mathbf{0}$. It can seen that the effect of increasing the

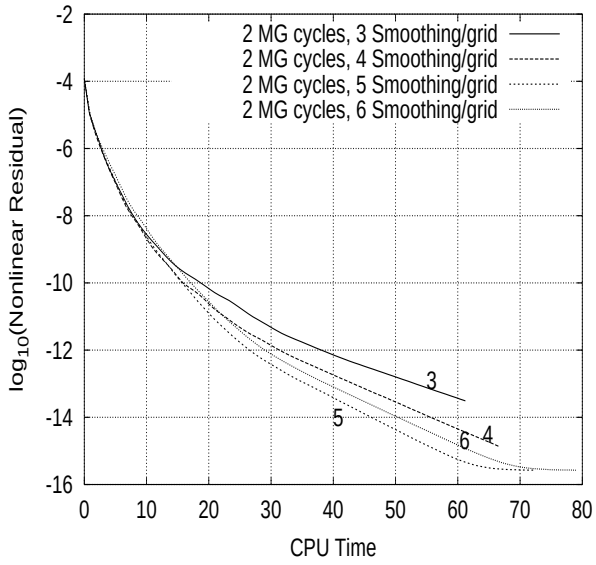

FIG. 5.4. *Effect of changing the number of smoothing iterations carried out on each grid of the multigrid on the runtime*

search directions is more pronounced as the nonlinear residual becomes smaller.

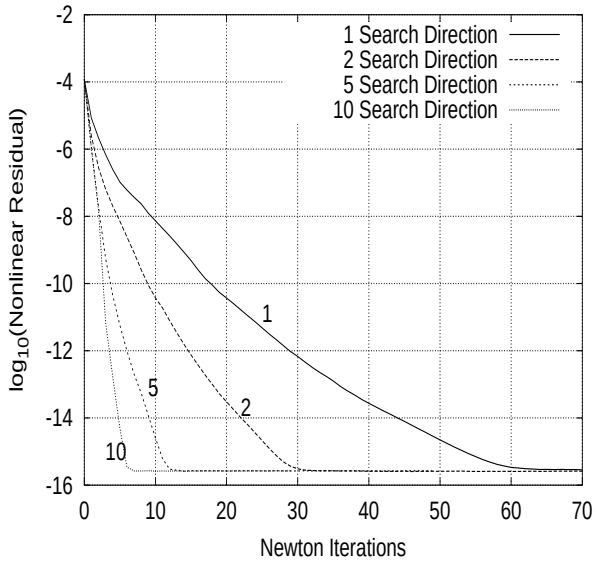

FIG. 6.1. *Effect of changing the number of search directions in PGMRES*

However, increasing the number of search directions by one incurs the following additional computational overhead :

- A single evaluation of the nonlinear residual on the fine grid.
- A single evaluation of the preconditioner, $\mathcal{P}_{\mathcal{N}}$.
- Additional matrix-vector and vector-vector products required to compute the extra search direction and the optimal solution, $\mathbf{x}^{(m)}$ in a larger Krylov subspace.
- Additional storage for the extra search direction.

Hence, the choice of the number of search directions was decided based on minimizing the CPU time required for a given level of residual reduction. Figure 6.2 plots the variation of the nonlinear residual against runtime for different choices of the number of search directions. It was determined that the use of five search directions was nearly optimal for most cases.

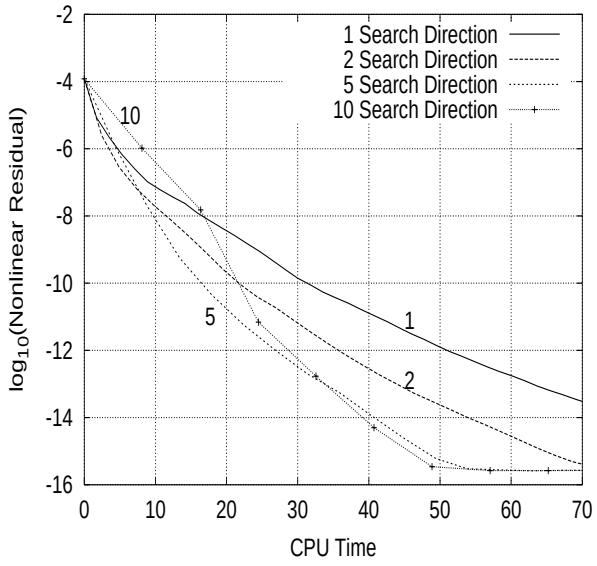

FIG. 6.2. *Effect of changing the number of search directions in PGMRES on the runtime*

7. Runtime Comparison of Different Schemes. In this section we examine the computational efficiency of two time-discretization schemes, RK64 and BDF2 as a function of accuracy levels. We simultaneously investigate the relative performances of the different implicit solution techniques discussed in this paper. Finally, we show that the combined improvements in efficiency obtained by using higher-order schemes and better nonlinear solvers such as LMG can result in up to an order of magnitude improvement in overall solution efficiency.

In order to compare the different schemes, we compare the runtimes required to advance the solution from an initial time T_i to a specified final physical time T_f, given an error tolerance in the final solution. In this study, we assume the error in the lift coefficient, C_L, to be a good measure of the integral error in the final solution. To measure the error in C_L, the numerical solution obtained using the RK64 scheme and $\Delta t = 0.0125$ was assumed to be numerically accurate with zero error. Finally, we choose 3 error tolerances, 10^{-2}, 10^{-3} and 10^{-4}, which we deem to be representative of engineering error tolerances, and make a detailed comparison of the different schemes. We choose a time interval $T_f - T_i = 1$, noting that the ratios of the runtimes of the various schemes should remain invariant with arbitrary choices of the time interval.

The physical time step, Δt, chosen to advance the equations to T_f depends on:

1. The physical time-stepping scheme, in this case either BDF2 or RK64.
2. The error tolerance level

Figure 7.1 plots the variation of the error in C_L for the two different physical time-stepping schemes. In order to obtain Figure 7.1 the nonlinear equations at each time step were converged such that the rms error in the density residual was less than 10^{-10}, in order to avoid any contamination of the temporal error. The circle symbols in the Figure indicate the time steps used in the following study for both schemes to achieve the three different prescribed error tolerances.

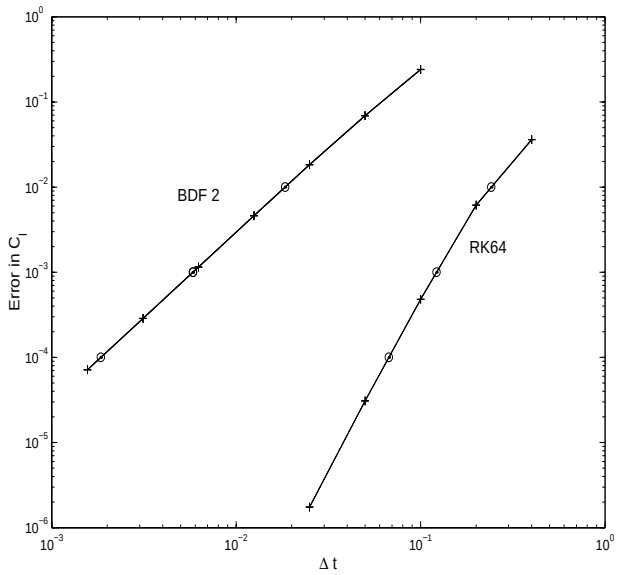

FIG. 7.1. *Comparison of time steps required for BDF2 and RK64 schemes*

It can be seen that for the range of error tolerances considered, BDF2 requires a much smaller time step than RK64. Furthermore, as RK64 is fourth-order accurate in time, the error decreases faster with a decrease in time step, making it more efficient at lower error tolerance levels.

Hence, the choice of the physical time-stepping scheme affects the overall efficiency of the simulation in three ways:

1. The number of time steps required to integrate to T_f, for a given temporal accuracy
2. The total number of non-linear solves per time step
3. The condition number of the non-linear systems to be solved. The non-linear systems produced by the RK scheme are generally more difficult to converge due to the larger time step involved with the higher-order scheme.

We now try to quantify the efficiency gains by using higher-order schemes and a more efficient implicit solution technique. A major contributor to the inefficiency of implicit methods is solving the nonlinear systems at each stage/step to inappropriate sub-iteration tolerances. If the nonlinear system is solved to tighter tolerances, the additional work does not increase the overall solution accuracy. We do not attempt to answer this question in any detail, but assume that it is sufficient to converge the nonlinear residual to 6 orders of magnitude less than the error in C_L. This level was determined through numerical experiments, by setting various convergence levels and measuring the final temporal error. While the ratio of residual convergence to error in C_L appears relatively large, this is an artifact of the fact that these quantities are inherently scaled in different manners, since C_L represents a global integrated quantity, as opposed to the average flow field residuals.

In Table 7.1 we present the runtimes for the different combination of schemes. In Figures 7.2 and 7.3 we show the nonlinear convergence characteristics of BDF2 and the second stage of RK64 for the three different C_L error tolerances considered. Note that the required levels of convergence are more stringent for the smaller time steps, as shown by the bar symbol on each line. In the actual computations, the residuals were converged only to these levels. In these figures, however, convergence down to machine accuracy is shown to illustrate the overall convergence behavior. These examples all utilize the nonlinear multigrid (NMG)

TABLE 7.1

Runtimes (in seconds on Pentium IV,1700MHZ) required for carrying out 1 *physical time step using the different nonlinear solvers, where time steps are chosen based on the given error tolerances on* C_L

Scheme	**Δt**	**NMG**	**LMG**	**PGMRES**
C_L Error $= 10^{-2}$				
BDF	0.01842	15.55	5.24	9.00
RK	0.24193	121.56	39.52	
C_L Error $= 10^{-3}$				
BDF	0.005825	12.45	4.59	9.01
RK	0.122047	131.4	43.84	
C_L Error $= 10^{-4}$				
BDF	0.001846	12.49	5.54	8.76
RK	0.067285	147.14	44.91	

scheme.

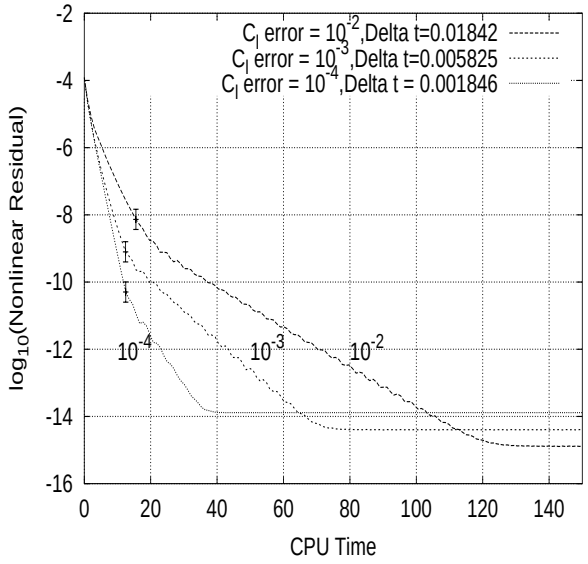

FIG. 7.2. *Comparison of the convergence of the nonlinear residual of BDF*2 *schemes for a single time step using NMG, where* Δt*'s are computed based on the given error tolerances on* C_L

We observe the following :

1. Faster convergence for smaller error tolerances, as Δt is reduced.
2. Slower overall convergence for RK64 relative to BDF2 for similar error tolerances, as larger time steps are used in RK64.
3. Although not shown, similar behavior is observed for the other two methods (LMG, PGMRES) as well.

In Tables 7.2 and 7.3 we quantify the efficiency gains obtained by shifting from BDF2 to RK64 for the two different nonlinear solvers, NMG and LMG respectively. The numbers in the different columns are ratios of the corresponding quantities used in RK and BDF respectively. The CPU time for a given time step is an order of magnitude larger for the RK schemes, and this ratio varies slowly with the increase in accuracy.

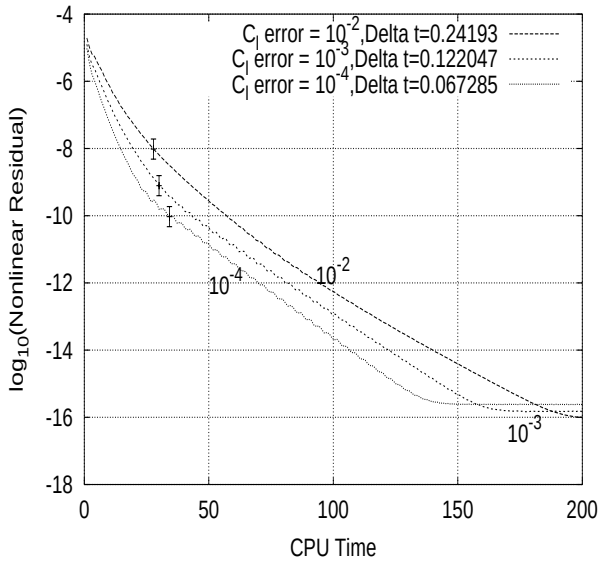

FIG. 7.3. *Comparison of the convergence of the nonlinear residual of RK64 schemes for the first stage of a single time step using NMG, where Δt's are computed based on the given error tolerances on C_L*

TABLE 7.2
BDF2 to RK64 Speedup factors for NMG

C_L **Error**	$\frac{\Delta t_{RK}}{\Delta t_{BDF}}$	$\frac{\textbf{Runtime}_{BDF}}{\textbf{Runtime}_{RK}}$ **for 1 time step**	**Overall Gain**
10^-2	13.13	0.128	1.68
10^-3	20.95	0.095	1.99
10^-4	36.45	0.085	3.1

This result is a function of the number of non-linear solutions required per time step, the relative stiffness of each non-linear problem, and the degree to which each system must be converged to maintain overall accuracy. However, the ratios of time steps between the two schemes is large and increases rapidly for higher accuracy levels, thus making the RK scheme more efficient overall, particularly at the more stringent error tolerances.

In Figure 7.4 we plot the variation of the nonlinear residual for a single time step of BDF2 using different nonlinear solvers as a function of the number of Newton iterations for LMG and PGMRES, and as a function of nonlinear multigrid cycles in the case of NMG. The inexact Newton methods exhibit faster convergence rates per iteration than the nonlinear multigrid NMG method. Furthermore, PGMRES is seen to outperform LMG as well, due to the fact that PGMRES produces a more accurate solution of the preconditioned linear system, equation (3.17) at each Newton iteration.

In Figure 7.5 we plot the variation of the nonlinear residual for a single time step of BDF2 using different nonlinear solvers as a function of CPU time. We make the following observations:

1. LMG is computationally more efficient than NMG as it performs fewer nonlinear residual evaluations.
2. Although PGMRES has faster convergence in terms of Newton Iterations, the additional cost of each Newton iteration outweighs the gain in non-linear convergence.

Finally, in Figure 7.6 we show the runtimes required to reach $T_f - T_i = 1$ using the different schemes,

TABLE 7.3
BDF2 to RK64 Speedup factors for LMG

C_L **Error**	$\frac{\Delta t_{RK}}{\Delta t_{BDF}}$	$\frac{\textbf{Runtime}_{BDF}}{\textbf{Runtime}_{RK}}$ **for 1 time step**	**Overall Gain**
10^-2	13.13	0.133	1.75
10^-3	20.95	0.105	2.2
10^-4	36.45	0.123	4.5

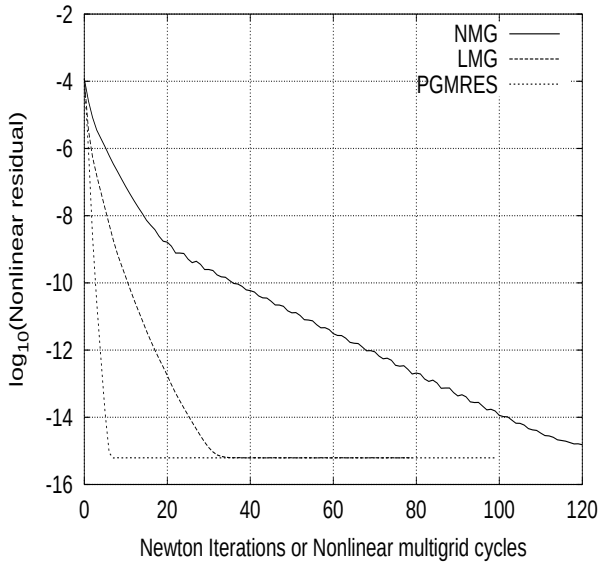

FIG. 7.4. *Comparison of the convergence of the nonlinear residual of BDF2 schemes using the 3 different nonlinear solvers*

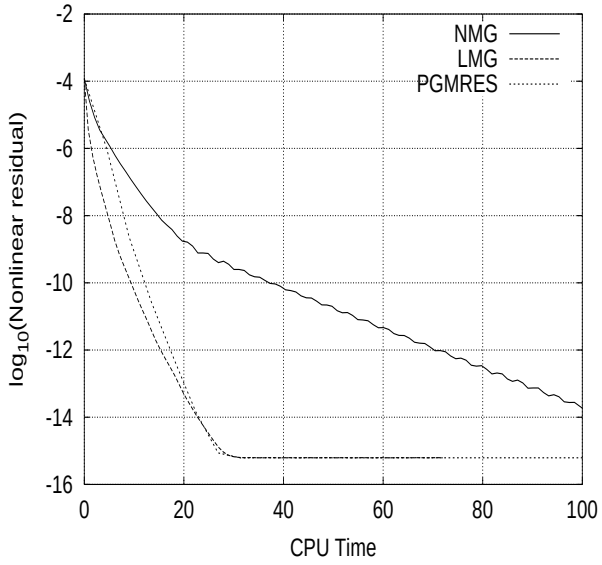

FIG. 7.5. *Comparison of the convergence of the nonlinear residual of BDF2 schemes using the 3 different nonlinear solvers against runtime*

TABLE 7.4
BDF2-NMG to RK64-LMG Speedup factors

C_L **Error**	$\frac{\Delta t_{RK}}{\Delta t_{BDF}}$	$\frac{\textbf{Runtime}_{BDF}}{\textbf{Runtime}_{RK}}$ **for 1 time step**	**Overall Gain**
10^-2	13.13	0.393	5.16
10^-3	20.95	0.284	5.95
10^-4	36.45	0.278	10.14

versus the error tolerance level in C_L. Figure 7.6 clearly indicates that the commonly used combination of BDF2 and nonlinear multigrid (NMG) exhibits the lowest computational efficiency for this problem. The most efficient solution strategy is obtained using the RK64 temporal discretization with the linear multigrid(LMG) solver, and these gains increase for higher accuracy levels.

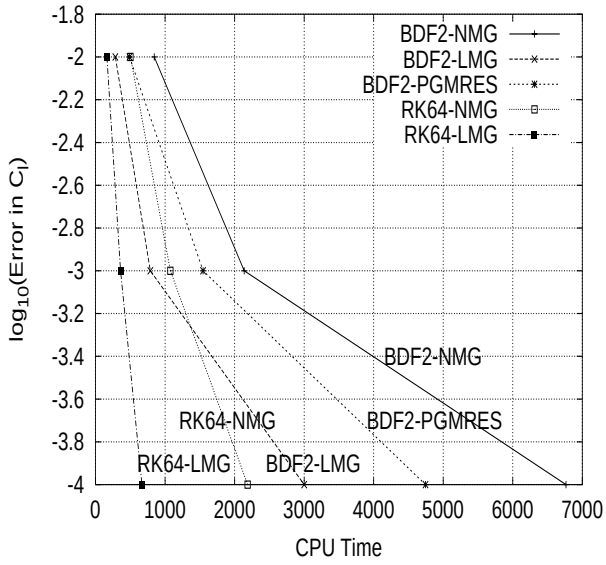

FIG. 7.6. *Runtimes required to reach $T_f = 1$ using the different schemes plotted against the error tolerance level in C_L*

In Table 7.4, we emphasize the overall gains obtained in shifting from the BDF2-NMG to the RK64-LMG solution strategy, as a function of the temporal accuracy. The gains obtained between the RK scheme and the BDF scheme and the gains obtained between the various non-linear solution strategies are multiplicative, producing a larger overall gain than for either method used alone. This compounded efficiency gain increases with solution accuracy, yielding up to an order of magnitude improvement for the highest accuracy considered.

8. Conclusions. For the test problem considered, higher-order implicit multi-stage Runge-Kutta schemes have been shown to produce higher accuracy at reduced cost as compared to BDF2. Additionally, inexact Newton solution strategies have been shown to be well suited for solving the non-linear systems which arise from temporal discretizations at each time step. The efficiency gains of both approaches are multiplicative, resulting in large potential savings when both methods are used in tandem. The combination of RK64 with Linear Multigrid method (LMG) worked very well for the error tolerances considered. The preconditioned GMRES (PGMRES) algorithm studied in this work provided the fastest asymptotic convergence among all methods but was found to be non-competitive due to the slower initial convergence of the method when

only partial convergence of the linear systems is required. In cases where more accurate linear system solutions are required, PGMRES may be more competitive. Overall efficiency of the time-integration schemes is greatly affected by the degree to which the non-linear systems at each time step are converged. The levels of convergence adopted in this work were determined a posteriori, and are therefore not predictive. A more exact quantification of the required convergence levels will be required in order to construct efficient time-dependent solution strategies.Future work will also include the use of temporal error estimation techniques coupled with dynamically adaptive time-step selection.

Acknowledgments. The authors would like to thank the information given by Bijl and Carpenter regarding the various time integration schemes.

REFERENCES

[1] B. van Leer, C. H. Tai, and K. G. Powell, *Design of optimally-smoothing multistage schemes for the Euler equations*, AIAA Paper 89-1933 (1989).

[2] H. Bijl, M. H. Carpenter, and V. N. Vatsa, *Time Integration Schemes for the Unsteady Navier-Stokes equations*, AIAA Paper 2001-2612, (2001).

[3] R. Dembo, S. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM Journal on Numerical Analysis, 19 (1982).

[4] G. H. Golub and C. F. V. Loan, *Matrix Computations*, John Hopkins University Press, 3 ed., 1996, ch. 10, pp. 508–513.

[5] E. Hairer, S. Norsett, and G. Wanner, *Solving ordinary differential equations I: nonstiff problems*, Springer-Verlag, 1991.

[6] B. R. Hutchinson and G. D. Raithby, *A multigrid method based on the additive correction strategy*, Numerical Heat Transfer, 9 (1986), pp. 511–537.

[7] A. Jameson, *Time-dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings*. AIAA Paper 91-1596, July 1991.

[8] C. A. Kennedy and M. H. Carpenter, *Additive Runge-Kutta schemes for Convection-Diffusion-Reaction equations*, Tech. Report NASA/TM-2001-211038, NASA Langley Research Center, 2001.

[9] D. J. Mavriplis, *Multigrid techniques for unstructured meshes*, in VKI Lecture Series, no. 1995-02 in VKI-LS, March 1995.

[10] ——, *Multigrid Strategies for Viscous Flow Solvers on Anisotrpic Unstructured Meshes*, Journal of Computational Physics, 145 (1998), pp. 141–165.

[11] ——, *Directional agglomeration multigrid techniques for high-Reynolds number viscous flows*, AIAA Journal, 37 (1999), pp. 1222–1230.

[12] ——, *An assessment of linear versus non-linear multigrid methods for unstructured mesh solvers*, Journal of Computational Physics, 175 (2002), pp. 302–325.

[13] N. D. Melson, M. D. Sanetrik, and H. L. Atkins, *Time-accurate Navier-Stokes calculations with multigrid acceleration*, in 6th Copper Mountain Conf. on Multigrid Methods, 1993, pp. 423–439. NASA Conference Publication 3224.

[14] V. A. Mousseau, D. A. Knoll, and W. J. Rider, *Physics-Based Precondtioning and the Newton-Krylov Method for the Non-equilibrium Radiation Diffusion*, Journal of Computational Physics, 160 (2000), pp. 743–765.

[15] E. J. Nielsen, W. K. Anderson, R. Walters, and D. E. Keyes, *Application of Newton-Krylov*

methodology to a three-dimensional Euler code, in Proceedings of the 12th AIAA CFD Conference, AIAA Paper 97-1953, June 1997.

[16] Y. Saad and M. H. Schultz, *A generalized minimal residual algorithm for solving non-symetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), p. 856.

[17] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, 1997, ch. 35.

[18] E. Turkel, *Preconditioning-squared methods for multidimensional aerodynamics*, in Proceedings of the 13th AIAA CFD Conference, Snowmass, CO, June,1997, pp. 856–866.

[19] V. Venkatakrishnan, *Implicit schemes and parallel computing in unstructured grid CFD*, in VKI Lecture Series VKI-LS 1995-02, Mar. 1995.

[20] L. B. Wigton, N. J. Yu, and D. P. Young, *GMRES Acceleration of Computational Fluid Dynamics Codes*, in Proceedings of the 7th AIAA CFD Conference, July 1985, pp. 67–74.

Appendix A. Butcher Coefficients for RK64. The Butcher coefficients for the RK64 scheme used in the paper are given in the table below. RK64 is a 6 stage with scheme, with 4^{th} order accuracy and 5 implicit stages. As in all ESDIRK schemes, $b_j = a_{6j}$

0	0	0	0	0	0
0.25	0.25	0	0	0	0
0.137776	−0.055776	0.25	0	0	0
0.144636866	−0.2239319076	0.4492950416	0.25	0	0
0.098258783284	−0.59154424282	0.8102105383	0.28316440571	0.25	0
0.15791629516	0	0.18675894052	0.68056529531	−0.275240531	0.25

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 2002	Contractor Report

4. TITLE AND SUBTITLE
HIGHER ORDER TIME INTEGRATION SCHEMES FOR THE UNSTEADY NAVIER-STOKES EQUATIONS ON UNSTRUCTURED MESHES

5. FUNDING NUMBERS
C NAS1-97046
WU 505-90-52-01

6. AUTHOR(S)
Giridhar Jothiprasad, Dimitri J. Mavriplis, and David A. Caughey

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
ICASE
Mail Stop 132C
NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
ICASE Report No. 2002-44

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-2199

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/CR-2002-211967
ICASE Report No. 2002-44

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Dennis M. Bushnell
Final Report
Submitted to the Journal of Computational Physics.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified–Unlimited
Subject Category 64
Distribution: Nonstandard
Availability: NASA-CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The efficiency gains obtained using higher-order implicit Runge-Kutta schemes as compared with the second-order accurate backward difference schemes for the unsteady Navier-Stokes equations are investigated. Three different algorithms for solving the nonlinear system of equations arising at each timestep are presented. The first algorithm (NMG) is a pseudo-time-stepping scheme which employs a non-linear full approximation storage (FAS) agglomeration multigrid method to accelerate convergence. The other two algorithms are based on Inexact Newton's methods. The linear system arising at each Newton step is solved using iterative/Krylov techniques and left preconditioning is used to accelerate convergence of the linear solvers. One of the methods (LMG) uses Richardson's iterative scheme for solving the linear system at each Newton step while other (PGMRES) uses the Generalized Minimal Residual method. Results demonstrating the relative superiority of these Newton's methods based schemes are presented. Efficiency gains as high as 10 are obtained by combining the higher-order time integration schemes with the more efficient nonlinear solvers.

14. SUBJECT TERMS
Jacobian-free Newton, Runge-Kutta methods, Navier-Stokes, multigrid, unstructured grid

15. NUMBER OF PAGES
28

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		