AIAA-2013-2831

21st AIAA Computational Fluid Dynamics Conference, June 25, 2013, San Diego, CA

High-Order Implicit Time Integration for Unsteady Compressible Fluid Flow Simulation

Pieter D. Boom,* and David. W. Zingg †

Institute for Aerospace Studies, University of Toronto, Toronto, Ontario, M3H 5T6, Canada

This paper presents an overview of high-order implicit time integration methods and their associated properties with a specific focus on their application to computational fluid dynamics. A framework is constructed for the development and optimization of general implicit time integration methods, specifically including linear multistep, Runge-Kutta, and multistep Runge-Kutta methods. The analysis and optimization capabilities of the framework are verified by rederiving methods with known coefficients. The framework is then applied to the derivation of novel singly-diagonally-implicit Runge-Kutta methods, explicit-first-stage singly-diagonally implicit Runge-Kutta methods, and singly-diagonally-implicit multistep Runge-Kutta methods. The fourth-order methods developed have similar efficiency to contemporary methods; however a fifth-order explicit-first-stage singly-diagonally-implicit Runge-Kutta method is obtained with higher relative efficiency. This is confirmed with simulations of van der Pol's equation.

I. Introduction

Many important fluid dynamic phenomena of interest are inherently unsteady. Examples include transition, turbulence, and aeroacoustics. In order to accurately simulate the underlying mechanisms governing these flows, time-accuracy is essential. These flows can often be quite complex and require simulations with a high degree of accuracy. Even simulations of laboratory-scale flows tend to become large quickly. Recent advances in computer architecture, parallel computing, and numerical methods continue to push the envelope of large-scale numerical simulation of unsteady flows; however, these simulations remain very expensive, both in terms of computational resources and time.

One of the distinct tools required for unsteady fluid simulation is an efficient time integration method. Spatially discretizing the governing partial differential equations describing the flow, generates a coupled system of nonlinear ordinary differential equations (ODEs). The numerical solution of ODEs dates back to Newton and Leibniz. With the development of differential calculus came the desire to numerically approximate increasingly complex differential equations.[22] Since then, numerous seminal contributions to the study have been made by: Euler, Runge and Kutta, Adams, Bashforth, and Moulton, Gear, Butcher, and many more. The methods they pioneered continue to be the foundation of numerical codes inside computational fluid dynamics (CFD) and throughout many other disciplines.

The study of numerical integration methods remains a vary active field of study. Recent works have investigated a new superclass of methods, called general linear (GL) methods.[2] This class includes traditional methods, such as linear multistep (LM) and predictor-corrector (PC) methods, including Runge-Kutta (RK) methods, as special cases, but also opens the door to radically new methods. Butcher's Diagonally Implicit MultiStage Integration Methods (DIMSIM),[6] and Hewitt and Hill's[23] algebraically stable methods are two examples which do not explicitly operate on solution values, but are very efficient and have desirable stability properties.

These new developments are slowly making their way into CFD. Bijl et al.[1] and Carpenter et al.[12] considered the Modified Extended Backwards Difference Formula (MEBDF) of Cash,[13] a general linear

*PhD Candidate, AIAA Student Member.

†Professor and Director, Tier 1 Canada Research Chair in Computational Aerodynamics and Environmentally Friendly Aircraft Design, J. Armand Bombardier Foundation Chair in Aerospace Flight, Associate Fellow AIAA.

Copyright © 2013 by Pieter D. Boom, and David W. Zingg. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

method, Hu et al.[24] and Najafi-Yazdi et al.[28] developed optimized methods for wave propagation in aeroacoustics, while Vatsa et al.[30] optimized backward difference methods to improved efficiency, just to name a few. However, consider the 1st International Workshop on High-Order CFD Methods in 2012. The most common time integration method was the classical fourth-order explicit RK method and for more complex problems, the second-order backward difference (BDF2) formula or semi-implicit Runge-Kutta methods. Despite the continued development of temporal integration methods, classical methods are still very common in CFD. There is significant potential for increasing the efficiency of current unsteady fluid flow simulations by investigating novel temporal integration methods.

This paper explores the development of novel time integration methods tailored for application in CFD. An overview of time integration methods and their associated properties is provided in the context of CFD. A development and optimization framework is constructed and verified. The framework is then applied to the development of novel diagonally implicit RK and multistep Runge-Kutta (MRK) methods. Simulations of van der Pol's equation are used to verify the accuracy of the methods and to evaluate their relative efficiency.

II. Overview of Time Integration Methods

This section presents a brief review of time integration methods and their associated properties. Along with the basic theory, we present some of the advantages and disadvantages of each property in the context of CFD.

II.A. Numerical Time Integration Classes

Historically, there are two primary classes of time integration methods: LM methods and PC methods, including RK methods. The former are often more efficient, requiring a single function evaluation per time step, while the latter often have superior stability properties, especially in the case of high-order and implicit methods. In the mid $1960s$, generalizations of these methods began to appear as a way to address the shortcomings of each class. Hybrid[19,20] and cyclical[17] LM methods combine sequences of LM methods in order to produce better stability properties and to overcome Dahlquist's first and second barriers. Methods such as Almost RK,[8] Pseudo RK,[11] and Two-step RK[25] augment the classical form of RK methods to access information from previous time steps, thereby increasing their efficiency. This eventually led to more general derivations and the birth of a new super-class of methods, GL methods.[2] GL methods encompass LM and PC methods as well as their generalizations as subclasses, but also provide a framework and a set of analysis tools to develop more general methods.

General linear (GL) methods have the potential to combine the efficiency of LM methods and the high-order stability of RK methods. Applied to the general non-autonomous differential equation $y' = f(y,t)$, the form of these methods can be written as

$$\begin{aligned} Y_k &= \textstyle\sum_{i=1}^{r} U_{ki}\mathbf{y}_i^{[n]} + h\sum_{j=1}^{s} A_{kj} f(Y_j, t^{[n]} + c_j h) \quad \text{for } k = 1,\ldots,s, \\ \mathbf{y}_k^{[n+1]} &= \textstyle\sum_{i=1}^{r} V_{ki}\mathbf{y}_i^{[n]} + h\sum_{j=1}^{s} B_{kj} f(Y_j, t^{[n]} + c_j h) \quad \text{for } k = 1,\ldots,r, \end{aligned}$$

where s is the number of internal stages with intermediate values Y, r is the number of external values y passed step-to-step, and A, B, U, and V are the coefficients of the method with abscissa c. Similar to Butcher tables for RK methods, GL methods are often presented in a partitioned matrix system of the form

$$\begin{bmatrix} Y_{s\times 1} \\ \mathbf{y}_{r\times 1}^{[n+1]} \end{bmatrix} = \begin{bmatrix} A_{s\times s} & U_{s\times r} \\ B_{r\times s} & V_{r\times r} \end{bmatrix} \begin{bmatrix} hF(Y)_{s\times 1} \\ \mathbf{y}_{r\times 1}^{[n]} \end{bmatrix}.$$

A few example GL methods are shown in Table 1.

GL methods are not in general self-starting and therefore require a starting method to generate the initial vector of external values. The starting procedure also influences the global order of the method and is required in the development of new methods. The external values are not restricted to be solution values at previous steps, and therefore, a unique starting procedure may be required for each new method. MRK methods, a subclass of GL methods, restrict $\mathbf{y}_i^{[n]}$ to $y^{[n-(i-1)]}$, where $y^{[n]}$ represents the solution at time step n. This limits the final potential of the method, but alleviates the need for unique starting procedures. MRK methods were first introduced by Guillou and Soulé in 1969[21] and can be seen as a direct extension of LM and RK method. Investigating MRK methods is a natural first step in the evaluation of current time

Method	Global Order	External Steps	Total Stages	Implicit Stages	Stability	$\|LTE\|$
MEBDF4[13]	4	3	3	3	L-Stable	$\sim 0.0892z^5$
DIMISM4[7]	4	4	4	4	L-Stable	$\sim 0.027z^5$
TSRK4[25]	4	2	3	3	L-Stable	$\sim 0.013z^5$
ARK4[9]	4	4	4	4	L-Stable	$\sim 0.013z^5$
HH4[23]	4	2	2	2	Algebraic	$\sim 0.046z^5$

Table 1. Some general linear methods and associated characteristics, $z = \lambda h$.

integration methods and the development of novel methods. MRK methods are the primary focus of this paper.

II.B. Order

Global order

Both low-order and high-order methods can achieve an arbitrary level of accuracy if they are consistent. High-order methods can often achieve this accuracy with a larger time step, but are also typically more computationally expensive, especially ones with similar stability properties to their low-order counterparts. The challenge is to determine where the higher rate of convergence outweighs the increased cost to generate the high-order update. In general, high-order methods become more efficient when the accuracy requirements of a simulation become stringent.

Stage order

Stage order refers to the minimum order of the individual internal stage approximations. This is relevant to high-order implicit integration of stiff ODEs where the convergence of the stiff components of the solution is tied to the stage order.[29] When the stage order is much lower than the global order of the method, the solution of stiff problems can suffer from order reduction. For example, this means that a globally fourth-order method with stage order one, could exhibit first-order convergence for these types of problems, and the potential benefit of high-order is lost. In this case, the order of the internal approximation must be increased to address the issue. In practice, most inviscid and laminar simulations are not stiff enough to exhibit order reduction, but unsteady Reynolds-averaged Navier-Stokes (URANS) simulations can be.[1,12] This is simulation and time step size dependent.

II.C. Stiff-Accuracy

One way of limiting order reduction is by setting $A_{s,i} = B_{1,i}$ and $U_{s,i} = V_{1,i}$. This eliminates the leading error term associated with the stiff component of the solution and is called stiff-accuracy.[10] It does not remove all the errors associated with low-order stage approximations, but increases the convergence of stiff variables by one order.

II.D. Stability

For high-order methods to be efficient, they must also have a suitably large stability region. The advantage of high-order methods is the ability to use larger time steps to achieve the same level of accuracy. If the stability bounds of the high-order method are too small, the larger time steps will not be stable.

A-Stability

Consider the linear test equation $y' = \lambda y$. Unconditional stability, or A-stability, states that the numerical solution to any inherently stable system, $Re(\lambda) \leq 0$, will remain stable in time independent of the size of the time step, h. Mathematically, this requires that the magnitude of the eigenvalues, σ_i, of the iteration (or stability) matrix given by

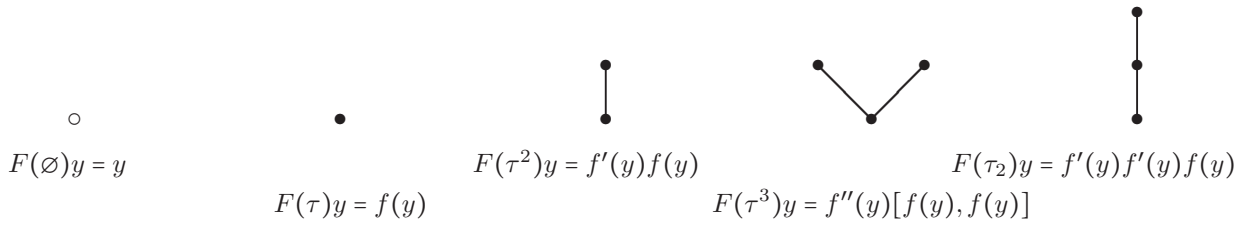

Figure 1. Trees of order $0 \to 3$ along with their corresponding elementary differentials

$$M(z) = V + zB(I - zA)^{-1}U,$$

be less than or equal to unity for all $z = \lambda h$ in the left-half complex plane, including the imaginary axis.

L-Stability

A-stability is a very attractive condition; however, high-frequency transients may be permitted to propagate for long time periods. Little or no damping of low-frequency modes is strongly desired; however, nonlinear interactions can generate non-physical high-frequency modes which contaminate the solution. An extension of A-stability is L-stability, which further requires that the spectral radius of the stability matrix tend to zero as $\lambda \to \infty$. This provides damping of high-frequency modes which may be non-physical, and perfect damping of modes at infinity. This is particularly useful for stiff simulations, such as in URANS.

II.E. Efficiency

Finally, the most important criterion, which in many ways incorporates all of the characteristics discussed above, is efficiency. Ultimately, the method that provides the least expensive means of obtaining a solution of predetermined accuracy, will be the method of choice. There are other considerations, for example, ease of implementation and memory usage; however a method which is not inherently efficient will not be used. For this exercise, efficiency will be evaluated by the L2 principal error norm, defined below, multiplied by $(s_\mathrm{i}/s_{\mathrm{i,ref}})^p$, where s_i is the number of implicit stages and p is the global order of the method, an estimate of the relative computational cost.[31]

III. Multistep Runge-Kutta Methods: Analysis and Construction

Much of the analysis of MRK methods is based on the rooted tree analysis developed by Butcher[4] for GL methods. Rooted trees are graphs with a unique node, the *root*, to which all other nodes are connected by a unique path. These trees are used to represent elementary differentials. In general, we define the elementary differential $F(t)y = f^{(m)}(F(t_1)y, \ldots, F(t_m)y)$ were $t = [t_1, \ldots, t_m]$ is a tree formed by joining the subtrees $t_1, \ldots, t_m$ by a single branch to the root. We define T to be the set of all rooted trees, T^* the set of all monotonically labeled trees and $\varnothing$ the empty tree. Figure 1 shows the trees of order $0 \to 3$ along with their corresponding elementary differentials. The order of a tree, $\rho(t)$, is defined as the number of nodes including the root, $\alpha(t)$ is the number of monotonic labellings, and $\alpha(u, t)$ is the number of monotonic labelings such that the subtree u, with the same root as t, is labeled first. Recursive definitions of $\rho(t)$, $\alpha(t)$, and $\alpha(u, t)$ can be found in Burrage.[3]

III.A. Starting Procedures and Butcher Series

Appropriate starting procedures for GL methods can be derived from Butcher series:

$$B(Q, y_\circ) = \sum_{t \in T^*} Q(t)[F(t)y_\circ]\frac{h^{\rho(t)}}{\rho(t)!}$$

where $Q : T \to \mathcal{R}$ is an arbitrary map. The Butcher series with $Q(t) = k^{\rho(t)}$ is the exact solution of the differential equation at $x_\circ + kh$, where $y_\circ = y(x_\circ)$. It can be shown that if the starting procedures for an MRK method are chosen to be Runge-Kutta processes, the associate maps are

$$Q(t) = [0, (-1)^p, \ldots, (1-r)^p]^T, \forall t \text{ with } \rho(t) = p \geq 0. \tag{1}$$

We can see from (1) that $Q(t) = Q(\tau^p)$ for all trees, t, of order p, where τ^p represents the "bushy" tree of order p. Therefore, we will define $Q_p = Q(\tau^p)$ from here on for simplicity.

III.B. Pre-Consistency and Consistency

Pre-consistency is the requirement that the internal stage approximations and the solution update be at least zeroth order. This can be interpreted as the requirement that if the analytical solution to a differential equation is a constant ($y' = 0$), and the numerical solution at the last n steps is exact, the solution at step $n+1$ will also be exact. For GL methods this is defined by[2]

$$VQ_0 = Q_0$$
$$UQ_0 = \mathbf{1},$$

where Q_0 is referred to as the preconsistency vector, and $\mathbf{1}$ is a column vector of ones.

Consistency further requires that the solution update be at least first order. This is equivalent to the requirement that the exact solution be recovered as the step size goes to zero and the number of steps goes to infinity. This is defined for GL methods as,

$$B\mathbf{1} + VQ_1 = Q_0 + Q_1,$$

where Q_1 is referred to as the consistency vector.

III.C. Zero-Stability and Convergence

Zero-stability is ascertained by considering the eigenvalues of the stability matrix at $z = 0$, which simplifies to $M(0) = V$. A method is considered zero-stable if $|\sigma_i(0)| \leq 1$, for all $i = 1, \ldots, r$, and any $\sigma_i(0) = 1$ is simple.[2,15] Strict zero-stability requires that one eigenvalue be equal to unity.

A method is convergent if and only if it is consistent and zero-stable.[2]

III.D. Global-Order Conditions

The required order conditions for a method of order p are,[5]

$$(V - I)Q_{\rho(t)} = \sum_{u \subset t} \binom{\rho(t)}{\rho(u)} \frac{\alpha(u,t)}{\alpha(t)} Q(u) - \rho(t) B \prod_{k=1}^{m} Y(t_k),$$

for all trees, $t = [t_1, \ldots, t_m]$, of order less than or equal to p, where $\binom{\blacksquare}{\blacksquare}$ represents the binomial coefficient and

$$Y(t) = UQ_{\rho(t)} + \rho(t) A \prod_{k=1}^{m} Y(t_k). \tag{2}$$

For a full derivation, refer to Burrage.[3]

III.E. Stage-Order Conditions

The stage-order conditions are derived from (2) and for stage order q are[5]

$$UQ_i = c^i - qAc^{i-1}, \qquad \forall i = 0, \ldots, q,$$

where c is the abscissa defined from the first-stage-order condition, also known as the stage-consistency condition.

III.F. Stability

The approach taken in this paper to enforce A-stability is to ensure that the stability contour remains in the right-half complex z-plane, including the imaginary axis. The stability contour is determined by setting the absolute value of the eigenvalues of the stability matrix to unity, and solving for the associated complex z-coordinates. All possible eigenvalues with absolute value unity are obtained by rewriting the eigenvalues in the form $\sigma = \exp^{i\theta}$, for $\theta \in [0, 2\pi)$. The condition for A-stability then reduces to forcing the real component of the z-coordinates to be greater than or equal to zero for all $\theta \in [0, 2\pi)$. The additional constraint for L-stability is that the spectral radius of the stability matrix approach zero as $z \to \infty$.

IV. Multistep Runge-Kutta Methods: Optimization

The construction of MRK methods is very complex, and deriving methods with optimal coefficients is difficult. Therefore, numerical optimization is employed to guide the search for novel methods. This section presents a framework for the development and optimization of implicit MRK methods, including LM and RK methods. Many of the features and analysis tools built into the framework are directly applicable to more general GL methods, into which future investigations are planned.

IV.A. Initial Solution and Design Variables

Initially, the general parameters of the desired method must be set. These include: the number of internal stages and external values, the order and stage order, the form of A, i.e. whether it is full, lower triangular, or strictly lower triangular, and whether or not to impose stiff-accuracy. Given these parameters, condition equations for preconsistency, consistency, stage-consistency, global order, stage order, and the additional requirement of L-stability can be constructed. The solution to these condition equations determine many of coefficients defining the method. The remaining undetermined coefficients, along with the abscissa, are used as design variables in the optimization.

IV.B. Objective Function

The key metric used to evaluate the performance of a new method is, in this case, efficiency. Global order and the number of implicit stages play a large role in the efficiency of a resulting method; however, they are set *a priori*. Therefore, to find an optimal method, with a given set of parameters, the L2 principal error norm of order $p+1$,

$$Er(p+1) = \sqrt{\sum_{\forall t | \rho(t)=p+1} (er(t))^2},$$

is used as the objective function to be minimized, where,

$$er(t) = (V - I)Q_{\rho(t)} - \sum_{u \subset t} \binom{\rho(t)}{\rho(u)} \frac{\alpha(u,t)}{\alpha(t)} Q(u) + \rho(t) B \prod_{k=1}^{m} Y(t_k),$$

is the violation of the order condition associated with tree t, where as before,

$$Y(t) = UQ_{\rho(t)} + \rho(t) A \prod_{k=1}^{m} Y(t_k).$$

IV.C. Constraints

The conditions for zero and A-stability are posed as inequalities and are enforced through nonlinear constraints.

Zero-stability

Zero-stability is determined by the eigenvalues of the matrix V, defined here as λ_V. The constraints for zero-stability are

$$|\lambda_V| \leq 1.$$

A-stability

The stability domain, $\theta \in [0, 2\pi)$, is discretized with n points to which the conditions for A-stability are applied. Barring an unusual stability contour, the number of points required is relatively small. Initial validation was done with 60 non-equidistant points.

IV.D. Optimization

The Sequential-Quadratic-Programming (SQP) method of Maple's nonlinear optimization construct is used to optimize the undetermined coefficients remaining after the solution to the initial condition equations, and to enforce the constraints. The design variables, the undetermined coefficients, are initialized with a random vector of values between two predetermined factors, set for this study between 0 and 1. Bounds of ±2 are placed on the design variables to prevent divergence, and finite differences are used to evaluate the gradients.

IV.E. Verification Results

To verify the optimization code, a few common time-integration methods with known coefficients are rederived using the framework discussed above.

Fourth-order SDIRK4(s=3)

To verify the solution of the condition equations, the three-stage fourth-order Singly-Diagonally-Implicit Runge-Kutta method, SDIRK4(s=3), is rederived. It is globally fourth-order accurate, with stage-order one, and is not stiffly accurate.

The general form of a three-stage SDIRK method contains 14 free coefficients, including the abscissa, and the initial parameters generate 15 nonlinear condition equations to be satisfied. The nonlinearity arises from the abscissa in the high-order conditions. The solution to this system generates 3 distinct sets of coefficients, but only one is A-stable. This solution corresponds exactly to the coefficients of SDIRK4(s=3) presented in the literature,[14]

$$\left[\begin{array}{ccc|c} (1+\alpha)/2 & 0 & 0 & 1 \\ -\alpha/2 & (1+\alpha)/2 & 0 & 1 \\ 1+\alpha & -(1+2\alpha) & (1+\alpha)/2 & 1 \\ \hline 1/6\alpha^2 & 1-1/3\alpha^2 & 1/6\alpha^2 & 1 \end{array}\right],$$

where $\alpha = 2\cos(\pi/18)/\sqrt{3}$.

Second-order ESDIRK2(s=3)

To verify the optimization code, the three-stage second-order Explicit-first-stage, Singly-Diagonally-Implicit Runge-Kutta method, ESDIRK2(s=3), is rederived. ESDIRK2 is a stiffly-accurate second-order method with an explicit first stage, permitting stage-order two, and a constant diagonal coefficient.

The general form of a stiffly-accurate three-stage ESDIRK method contains nine free coefficients, including one in the abscissa. Solving the required condition equations determines all but one coefficient. In practice the value of this coefficient can be determined analytically using the concept of A-acceptability and the E-polynomial of Ehle;[18] however, for the purpose of verifying the optimization code and implementation of the stability constraints, it is left undetermined. The result of the numerical optimization matches the coefficients of ESDIRK2 to 10 decimal places:

$$\left[\begin{array}{ccc|c} 0 & 0 & 0 & 1 \\ 0.2928932187884473 & 0.2928932187884473 & 0 & 1 \\ 0.3535533905454085 & 0.3535533905454085 & 0.2928932187884473 & 1 \\ \hline 0.3535533905454085 & 0.3535533905454085 & 0.2928932187884473 & 1 \end{array}\right].$$

Second-order BDF2OPT(θ = 0.5)

A similar optimization is carried out for BDF2OPT[30] to verify the ability of the code to handle multistep methods. This is a second-order linear multistep method developed by Vatsa et al.[30] by taking a linear combination of BDF2 and BDF3. The goal was to lower the truncation error of BDF2, while maintaining A-stability, by adding an additional step. The solution of the condition equations leaves one free coefficient to be optimized. This can be determined analytically using the stability definitions of Dahlquist,[16] but is left undetermined to test the optimization code.

The results of the optimization match to 8 decimal places the coefficients of BDF2OPT(θ = 0.5), the three-step A-stable linear multistep method with the lowest truncation error:[30]

$$\left[\begin{array}{c|ccc} 0.5999999987499999 & 1.5000000031250000 & -0.6000000050000000 & 0.1000000018750000 \\ \hline 0.5999999987499999 & 1.5000000031250000 & -0.6000000050000000 & 0.1000000018750000 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right]$$

V. Development of new methods

In this section, new methods are developed using the tool described above, with emphasis on L-stable stiffly-accurate fourth-order implicit methods. The fourth-order six-stage explicit-first-stage singly-diagonally-implicit Runge-Kutta (ESDIRK4) scheme of Kennedy and Carpenter[26] is chosen as a reference for this discussion due to its high efficiency and widespread use. The method is L-stable, stiffly-accurate, and has an explicit first stage enabling stage order two and reducing the number of implicit stages to five. Its L2 principal error norm is $Er(5) \approx 0.18936$. The reader should be aware that the method of Kennedy and Carpenter was developed in the context of implicit/explicit additive Runge-Kutta schemes with embedded methods, error controllers, dense output, and stage-value predictors. Also, rational coefficients were specifically chosen rather than fully optimizing with respect to an error norm. In contrast, the present study does not consider the above-mentioned features and presents fully optimized coefficients. Therefore, the conclusions drawn below should be carefully weighed in light of this.

V.A. Singly-Diagonally-Implicit Runge-Kutta (SDIRK) methods

Fourth-order SDIRK(s=5)

The minimum number of stages required to obtain an A-stable fourth-order SDIRK scheme is three, as shown above. However, this method is not L-stable or stiffly-accurate, and has an L2 principal error norm of $Er(5) \approx 20.8114$, which is approximately two orders of magnitude larger than the reference ESDIRK4 method. In order to obtain a stiffly-accurate fourth-order method, the number of stages must be increased to five. The general form of a stiffly-accurate five-stage SDIRK scheme contains 15 free coefficients, including the abscissa. The solution to the initial condition equations determines 12 of these coefficients, leaving three to be optimized. In this case, the three free variables are c_1, c_3, and c_4 of the abscissa. The coefficients of the optimized L-stable method are found in Table 2. It has an L2 principal error norm of $Er(5) \approx 0.13192$, approximately 30% smaller than the reference method. Having the same number of implicit stages as ESDIRK4, the optimized method should be more computationally efficient. However, SDIRK methods are limited to stage order one; therefore, the reference method is still likely a better choice for very stiff problems.

V.B. Explicit-First-Stage Singly-Diagonally-Implicit Runge-Kutta (ESDIRK) methods

As mentioned above, ESDIRK methods have a unique advantage over SDIRK methods in that they are able to obtain stage order two. The methods developed in the section all take advantage of this property.

Fifth-order ESDIRK(s=6)

Given the additional flexibility in the present study relative to the study which generated the reference method, the condition equations of a five-stage fourth-order stiffly-accurate ESDIRK method were solved. Four distinct sets of coefficients were obtained. However, the only L-stable set of coefficients corresponds

to the five-stage method determined by Kennedy and Carpenter,[26] which has a relatively large L2 principal error norm. Similarly to Kennedy and Carpenter, a six-stage fourth-order stiffly-accurate method was then pursued. The optimization of the free coefficients yielded a method with an infinitesimal L2 principal error norm, indicating that a fifth-order method may exist with the same number of stages. Indeed a solution to the fifth order conditions is obtained. Of the 20 initial free coefficients, only three remain to be optimized, c_2, c_4, and c_5 of the abscissa. The resulting optimized fifth-order method, which has an L2 principal error of $Er(6) \approx 0.60321$, is found in Table 3. Since the number of implicit stages is the same, we would expect this method to be significantly more efficient than the ESDIRK reference method, especially as the time step size is reduced.

V.C. Singly-Diagonally-Implicit Multistep Runge-Kutta (SDIMRK) methods

This subsection presents an initial investigation of singly-diagonally-implicit MRK methods. Future work will include a comparison with explicit-first-stage singly-diagonally implicit MRK methods.

Fourth-order SDIMRK4(r=2, s=3)

Consider again the fourth-order SDIRK methods discussed above. A three-stage A-stable method can be obtained, but it is neither L-stable nor stiffly accurate. To obtain these additional characteristics, five stages are required. With SDIMRK methods, these characteristics can be obtained with three stages and two time levels. Solving the condition equations for the 12 free coefficients, including the abscissa, 12 distinct sets of coefficients are obtained. Only four of these sets are zero-stable, and only one is L-stable. Unfortunately, the L2 principal error norm of the L-stable method is $Er(5) \approx 137.9548$, which is three orders of magnitude larger than that of the reference ESDIRK method. If the requirement for L-stability is relaxed slightly, an L(89.42°)-stable method exists with an L2 principal error norm of $Er(5) \approx 3.1160$. This is still an order of magnitude larger than the reference solution, but requires 40% fewer stages. The coefficients of this method are found in Table 4. To compare the efficiency of the methods, the L2 principal error norm is scaled by $(s_\mathrm{i}/s_{\mathrm{i,ref}})^p$, where s_i is the number of implicit stages, and p is the global order of the methods. This yields a comparable error norm with respect to the cost of the methods of approximately 0.4038, which is about twice as large as the reference. This method, similar to SDIRK methods, also only has stage order one.

Fourth-order SDIMRK4(r=2, s=4)

Increasing the number of implicit stages by one, we can obtain an optimized four-stage two-time-level L-stable stiffly-accurate fourth-order SDIMRK method with an L2 principal error norm of $Er(5) \approx 0.3797$. This is about two times larger than the reference method, but requires 20% fewer implicit stages. The coefficients of the optimized method are found in Table 5. The relative efficiency of the method can be evaluated using the same analysis as above. A comparable error norm with respect to the cost of the methods is approximately 0.1555, approximately 20% smaller than the reference. This method, again like the previous SDIMRK method, only has stage order one.

VI. Unsteady Flow Test Cases

VI.A. Van der Pol's Equation

Simulation of van der Pol's equation is chosen as an initial testbed to verify the convergence rates of the new methods developed. Van der Pol's equation is a second-order nonlinear ODE:

$$y'' - \mu(1 - y^2)y' + y = 0,$$

which is solved as a first-order system,

$$\left\{ \begin{array}{c} y' = z \\ z' = \mu(1 - y^2)z - y \end{array} \right\},$$

where μ is called the stiffness parameter. For this exercise, the stiffness parameter is set to 10, the initial conditions are $y = 2$, and $z = -0.6666654321121172$, and the time domain is $t = [0, 0.5]$.[26]

Temporal convergence is based on the L2-error defined as

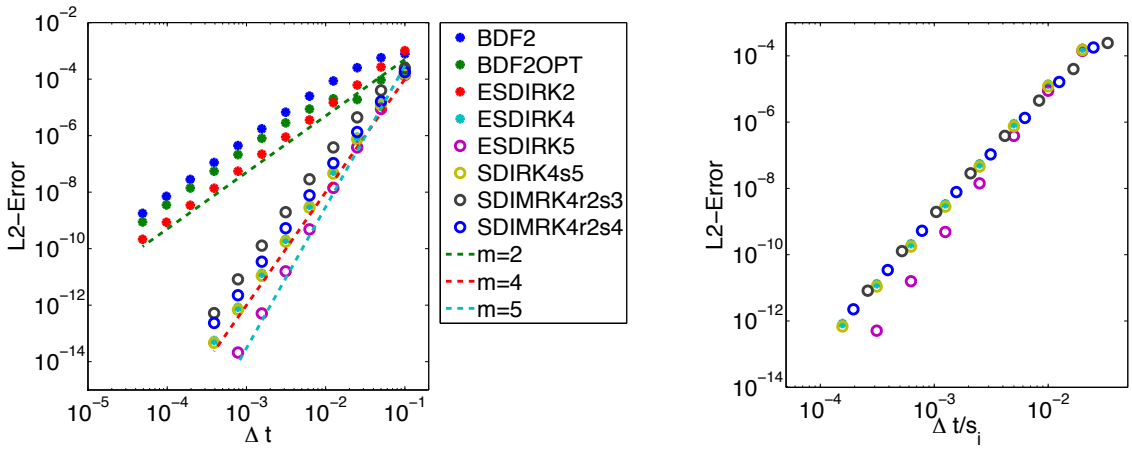

Figure 2. Van der Pol's equation: temporal convergence (left) and estimation of efficiency (right). Asterisks denote methods from the literature and circles denote methods developed in this work.

$$\text{L2-Error} = \sqrt{\frac{\sum_{k=1}^{N}(y_{k,\text{num}} - y_{k,\text{ref}})^2}{N}},$$

where $N = t_f/\Delta t = 0.5/\Delta t$, and the reference solution is obtained with an explicit seven-stage sixth-order RK method and a $\Delta t \approx 3.05 \times 10^{-6}$. Results of the temporal convergence study, obtained using step sizes between $\Delta t = 0.1$ and $\Delta t \approx 4.88 \times 10^{-5}$, are shown in Figure 2. The order of convergence of all methods is recovered. An estimation of the efficiency of the methods compared to the reference ESDIRK4 method is also shown in the same Figure. This estimation is obtained by plotting L2-error versus the time step size divided by the number of implicit stages, s_i. The fifth-order ESDIRK method is the most efficient; all of the fourth-order methods appear to have approximately the same relative efficiency.

VII. Conclusions

This paper discusses some of the properties required of an efficient time-integration method for stiff unsteady fluid flow simulations and explores some opportunities to develop novel methods to increase their efficiency. A robust time-integration development and optimization framework is constructed and verified by rederiving methods with know coefficients. This framework is then applied to develop some novel SDIRK, ESDIRK, and SDIMRK methods.

A five-stage L-stable stiffly-accurate fourth-order SDIRK method is developed with an L2 principal error norm approximately 40% lower than the reference ESDIRK4 method of Kennedy and Carpenter.[26] It only has stage order one, and does not incorporate all the additional features of the reference method, but for mildly stiff problems where order reduction in not an issue, the method provides an efficient alternative. A novel and efficient six-stage, L-stable, stiffly-accurate, fifth-order ESDIRK method is obtained. Along with a small L2 principal error norm, a full order of accuracy is gained above the reference method with the same number of implicit stages. This method is, therefore, theoretically the most efficient method considered in this study. This is further supported by simulations of van der Pol's equation. Finally, an initial investigation of SDIMRK methods found an efficient L(89.42°)-stable three-stage two-time-level method with a reasonably small error norm considering the number of implicit stages. Increasing the number of implicit stages by one significantly reduces the error norm and makes the method slightly more efficient than ESDIRK4.

Simulation of the van der Pol equation demonstrates the correct convergence rates of a few known methods as well as the novel methods developed in this work. A simple estimation indicates similar efficiency of the all the fourth-order methods, and superior efficiency of the new fifth-order ESDIRK method developed.

Future work includes further investigation and development of high-order implicit time integration methods such as SDIRK, ESDIRK, SDIMRK, ESDIMRK, cyclical implicit LM, Rosenbrock, and more general GL methods. An evaluation of additional features considered in other studies is planned, along with a study of different error measures to be used for the objective function. Finally, application to large-scale complex

unsteady flows of interest, such as simulation of the Taylor-Green vortex flow and transitional flow over a low-Reynolds number wing, is desired to demonstrate the efficiency of the novel methods developed.

Acknowledgments

Computations were performed on the GPC supercomputer at the SciNet HPC Consortium. SciNet is funded by the Canada Foundation for Innovation under the auspices of Compute Canada; the Government of Ontario; Ontario Research Fund-Research Excellence; and the University of Toronto.[27]

References

[1]Bijl, H., Carpenter, M.H., Vatsa, V.N. and Kennedy, C.A., "Implicit Time Integration Schemes for the Unsteady Compressible Navier-Stokes Equations: Laminar Flow", *Journal of Computational Physics*, Vol. 179, No. 1, 2002, pp. 313-329.

[2]Burrage, K., and Butcher, J.C., "Non-Linear Stability of a General Class of Differential Equation Methods", *BIT*, Vol. 20, 1980, pp. 185-203.

[3]Burrage, K., "Order Properties of Implicit Multivalue Methods for Ordinary Differential Equations", *IMA J. Num. Ana.*, Vol. 8, 1988, pp. 43-69.

[4]Butcher, J.C., "Coefficients for the Study of Runge-Kutta Processes", *J. Aus. Math. Soc.*, Vol. 3, No. 2, 1963, pp. 185-201.

[5]J.C. Butcher., "Implicit Runge-Kutta Processes", *J. Amer. Math. Soc.*, Vol. 18, No. 85, 1964, pp. 54-64.

[6]Butcher, J.C., "Diagonally-implicit multi-stage integration methods", *J. App. Num. Math.*, Vol. 11, 1993, pp. 347-363.

[7]Butcher, J. and Jackiewicz, Z., "Construction of diagonally implicit general linear methods of type 1 and 2 for ordinary differential equations", *J. App. Num. Math.*, Vol. 21, 1996, pp. 385-415.

[8]Butcher, J.C., "An introduction to "Almost Runge-Kutta" methods", *J. App. Num. Math.*, Vol. 24, No. 2-3, 1997, pp. 331-342.

[9]Butcher, J.C., and Rattenbury, N., "ARK methods for stiff problems", *J. App. Num. Math.*, Vol. 53, 2005, pp. 165-181.

[10]Butcher, J.C., "Numerical Methods for Ordinary Differential Equations", *Wiley*, West Sussex, 2003.

[11]Byrne, G. and Lambert, R.J., "Pseudo Runge-Kutta methods involving two points", *Journal Assoc. Comp. Mach.*, Vol. 13, No. 1, 1966, pp. 114-123.

[12]Carpenter, M.H., Kennedy, C.A., Bijl, H., Viken, S.A. and Vatsa, V.N., "Fourth-Order Runge-Kutta Schemes for Fluid Mechanics Applications", *Journal of Scientific Computing*, Vol. 25, No. 1-2, 2005, pp. 157-194.

[13]Cash, J., "The integration of Stiff initial value problems in ODEs using modified extended backward differentiation formulae", *Computers and Mathematics with Applications*, Vol. 9, No. 5, 1983, pp. 645-657.

[14]M.A. Crouzeix. "Sur l'approximation des équations différentielle opérationelles linéaires par des méthodes de Runge-Kutta", *PhD Thesis*, Université de Paris VI, Pairs, 1975.

[15]D. Dahlquist. "Convergence and stability in the numerical integration of ordinary differential equations", *Math. Scand.*, Vol. 4, 1956, pp. 33-50.

[16]Dahlquist, D., "A special stability problem for linear multistep methods", *BIT*, Vol. 3, 1963, pp. 27-43.

[17]Donelson III, J., and Hansen, E., "Cyclic Composite Multistep Predictor-Corrector Methods", *SIAM J. Numer. Anal.*, Vol. 8, No. 1, 1971, pp. 137-157.

[18]Ehle, B.L., "High order A-stable Methods for the Numerical Solution of Systems of DEs", *BIT*, Vol. 8, 1968, pp. 276-278.

[19]Gear, C.W., "Hybrid Methods for Initial Value Problems in Ordinary Differential Equations", *SIAM J. Numer. Anal.*, Vol. 2, No. 1, 1964, pp. 69-86.

[20]Gragg, W.B., and Stetter, H.J., "Generalized Multistep Predictor-Corrector Methods", *J. Ass. Comp. Mach.*, Vol. 11, No. 2, 1964, pp. 188-209.

[21]Guillou, A. and Soulié, J.L., "La Résolution Numérique des Problèmes Différentiels aux Condition Initiales par des Méthodes de Collocation", *R.I.R.O.*, Vol. R-3, 1969, pp. 17-44.

[22]Haier, E., Nørsett, S.P., and Wanner, G., "Solving Ordinary Differential Equation I: Nonstiff Problems", *Springer-Verlag*, 1993.

[23]Hewitt, L. and Hill, A., "Algebraically stable diagonally implicit general linear methods", *J. App. Num. Math.*, Vol. 60, 2010, pp. 629-636.

[24]Hu, F.Q., Hussaini, M.Y., and Manthey, J.L., "Low-dissipation and low-dispersion Runge-Kutta schemes for computational acoustics", *J. Comp. Phys.*, Vol. 124, No. 1, 1996, pp.177-191.

[25]Jackiewicz, Z. and Tracogna, S., "Two-Step Runge-Kutta Methods for Ordinary Differential Equations", *SIAM J. Num. Anal.*, Vol. 32, No. 5, 1995, pp. 1390-1427.

[26]Kennedy, C. and Carpenter, M., "Additive Runge-Kutta Schemes for Convection-Diffusion-Reaction Equations", *J. App. Num. Math.*, Vol. 44, No. 1-2, 2003, pp. 139-181.

[27]Loken, C., Gruner, D., Groer, L., Peltier, R., Bunn, N., Craig, M., Henriques, T., Dempsey, J., Yu, C.-H., Chen, J., Dursi, L.J., Chong, J., Northrup, S., Pinto, J., Knecht, N. and Van Zon, R., "SciNet: Lessons Learned from Building a Power-efficient Top-20 System and Data Centre", *Journal of Physics - Conference Series*, Vol. 256, No. 1, 2010, 012026.

[28]Najafi-Yazdi, A., and Mongeau, L., "A low-dispersion and low-dissipation implicit Runge-Kutta scheme", *J. of Comp. Phys.*, Vol. 233, No. 1, 2013, pp. 315-323.

[29]Prothero, A. and Robinson, A., "On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations", *Math. of Comp.*, Vol. 28, 1974, pp. 145-162.

[30]Vatsa, V., Carpenter, M. and Lockard, D., "Re-evaluation of an Optimized Second Order Backward Difference (BDF2OPT) Scheme for Unsteady Flow Applications", *48th AIAA Aerospace Sciences Meeting*, 2010-0122, Jan, 2010, Orlando, FL.

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

0.2479918251579609	0	0	0	0	1
0.6351176551064315	0.2479918251579609	0	0	0	1
0.2550906884612272	−0.0321316227845224	0.2479918251579609	0	0	1
0.4830415144073407	−0.0245416157211815	−0.1025114005213252	0.2479918251579609	0	1
0.8547186509604199	−0.3432093403548739	−1.4874886955097541	1.7279875597462471	0.2479918251579609	1
0.8547186509604199	−0.3432093403548739	−1.487488695509754	1.7279875597462471	0.2479918251579609	1

Table 2. Coefficients of the optimized fourth-order SDIRK4(s=5)

0	0	0	0	0	0	1
0.2780538411364465	0.2780538411364465	0	0	0	0	1
0.3137405401502951	0.4363327154020044	0.2780538411364465	0	0	0	1
0.2741986534107860	−0.0164268277321164	0.0048197082596452	0.2780538411364465	0	0	1
−0.2441776975175844	−3.3203529439447852	0.0477747285706825	3.2974431145814931	0.2780538411364465	0	1
−0.2786732780227907	1.8929947094010862	−0.1280948204262490	−1.3574693381380240	0.5931888860495311	0.2780538411364465	1
−0.2786732780227907	1.8929947094010862	−0.1280948204262490	−1.3574693381380240	0.5931888860495311	0.2780538411364465	1

Table 3. Coefficients of the optimized fifth-order ESDIRK5(s=6)

0.3781157714639682	0	0	1.1268744185479192	−0.1268744185479192
0.188647323132050	0.3781157714639682	0	1.1900727431400351	−0.1900727431400351
1.239967976276309	−0.6471245367633625	0.3781157714639682	1.0290407890230846	−0.0290407890230846
1.239967976276309	−0.6471245367633625	0.3781157714639682	1.0290407890230846	−0.0290407890230846
0	0	0	1	0

Table 4. Coefficients of the optimized fourth-order SDIMRK4(s=3,r=2)

0.2200265793795681	0	0	0	1.0462707164341444	−0.0462707164341444
0.0372874705511078	0.2200265793795681	0	0	1.0589996006952576	−0.0589996006952576
−2.4939012868198909	2.4122830553518624	0.2200265793795681	0	1.2255368955206809	−0.2255368955206809
−1.3681633961675930	2.4420771360486791	−0.3184471035454308	0.2200265793795681	1.0245067842847765	−0.0245067842847765
−1.3681633961675930	2.4420771360486791	−0.3184471035454308	0.2200265793795681	1.0245067842847765	−0.0245067842847765
0	0	0	0	1	0

Table 5. Coefficients of the optimized fourth-order SDIMRK4(s=4,r=2)