Diagonally Implicit Runge-Kutta Methods for Ordinary Differential Equations. A Review

Christopher A. Kennedy
Private Professional Consultant, Palo Alto, California

Mark H. Carpenter
Langley Research Center, Hampton, Virginia

March 2016
Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA’s STI. The NASA STI Program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collection of aeronautical and space science STI in the world. Results are published in both non-NASA channels and 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 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 PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored 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 also include creating custom thesauri, building customized databases, and organizing and publishing research results.

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

- Access the NASA STI program home page at [http://www.sti.nasa.gov](http://www.sti.nasa.gov)
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at (757) 864-9658
- Phone the NASA STI Help Desk at (757) 864-9658
- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199
Diagonally Implicit Runge-Kutta Methods for Ordinary Differential Equations. A Review

Christopher A. Kennedy
Private Professional Consultant, Palo Alto, California

Mark H. Carpenter
Langley Research Center, Hampton, Virginia

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23681-2199

March 2016
Acknowledgments

Helpful suggestions from Ernst Hairer, Alan Hindmarsh, Marc Spijker, and Michel Crouzeix are greatly appreciated. Matteo Parsani kindly and thoroughly reviewed this paper, making many helpful recommendations.

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:
NASA STI Program / Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
757-864-6500
Abstract

A review of diagonally implicit Runge-Kutta (DIRK) methods applied to first-order ordinary differential equations (ODEs) is undertaken. The goal of this review is to summarize the characteristics, assess the potential, and then design several nearly optimal, general purpose, DIRK-type methods. Over 20 important aspects of DIRK-type methods are reviewed. A design study is then conducted on DIRK-type methods having from two to seven implicit stages. From this, 15 schemes are selected for general purpose application. Testing of the 15 chosen methods is done on three singular perturbation problems. Based on the review of method characteristics, these methods focus on having a stage order of two, stiff accuracy, L-stability, high quality embedded and dense-output methods, small magnitudes of the algebraic stability matrix eigenvalues, small values of \( a_{ii} \), and small or vanishing values of the internal stability function for large eigenvalues of the Jacobian. Among the 15 new methods, ESDIRK4(3)6L[2]SA is recommended as a good default method for solving stiff problems at moderate error tolerances.

Contents

1 Introduction 3

2 Background 6
   2.1 Order Conditions 8
   2.2 Simplifying Assumptions 10
   2.3 Error 16
   2.4 Linear Stability 17
   2.5 Nonlinear Stability 26
   2.6 Internal Stability 30
   2.7 Dense Output 32
   2.8 Conservation, Symplecticity and Symmetry 32
   2.9 Dissipation and Dispersion Accuracy 35
   2.10 Memory Economization 37
   2.11 Regularity 37
   2.12 Boundary and Smoothness Order Reduction 39
   2.13 Efficiency 41
   2.14 Solvability 41
   2.15 Implementation 42
   2.16 Step-Size Control 46
   2.17 Iteration Control 52
   2.18 Stage-Value Predictors 55
   2.19 Discontinuities 66
   2.20 Software 69

3 Early Methods 69
4 Two- and Three-stage Methods ($S_1 = 2$)  
4.1 Second-Order Methods  
4.1.1 ESDIRK  
4.1.2 SDIRK  
4.2 Third-Order Methods  
4.2.1 ESDIRK  
4.2.2 SDIRK  
4.3 Fourth-Order Methods  
4.3.1 ESDIRK  

5 Three- and Four-stage Methods ($S_1 = 3$)  
5.1 Third-Order Methods  
5.1.1 ESDIRK  
5.1.2 QESDIRK  
5.1.3 SDIRK  
5.2 Fourth-Order Methods  
5.2.1 ESDIRK  
5.2.2 SDIRK  
5.3 Fifth-Order Methods  
5.3.1 ESDIRK  

6 Four- and Five-stage Methods ($S_1 = 4$)  
6.1 Third-Order Methods  
6.1.1 ESDIRK  
6.1.2 QESDIRK  
6.1.3 SDIRK  
6.2 Fourth-Order Methods  
6.2.1 ESDIRK  
6.2.2 QESDIRK  
6.2.3 SDIRK  
6.3 Fifth-Order Methods  
6.3.1 ESDIRK  
6.4 Sixth-Order Methods  
6.4.1 EDIRK  

7 Five- and Six-stage Methods ($S_1 = 5$)  
7.1 Fourth-Order Methods  
7.1.1 ESDIRK  
7.1.2 QESDIRK  
7.1.3 SDIRK  
7.1.4 QSDIRK  
7.2 Fifth-Order Methods  
7.2.1 ESDIRK  
7.2.2 SDIRK  
7.3 Sixth-Order Methods
1 Introduction

The diagonally implicit Runge-Kutta (DIRK) family of methods is possibly the most widely used implicit Runge-Kutta (IRK) method in practical applications involving stiff, first-order, ordinary differential equations (ODEs) for initial value problems (IVPs) due to their relative ease of implementation. In the notation introduced by Butcher [59], they are characterized by a lower triangular $A$-matrix with at least one nonzero diagonal entry and are sometimes referred to as semi-implicit or semi-explicit Runge-Kutta methods. This structure permits solving for
each stage individually rather than all stages simultaneously. Applications for these methods include fluid dynamics [36, 38, 39, 93, 94, 208, 222, 240, 248, 438, 440, 458], chemical reaction kinetics and mechanisms [49, 271], medicine [421], semiconductor design [31, 425], plasticity [449], structural analysis [133, 331], neutron kinetics [130], porous media [33, 137], gas transmission networks [107], transient magnetodynamics [306], and the Korteweg-de Vries [369] and Schrödinger [135] equations. They may also represent the implicit portion of an additive implicit-explicit (IMEX) method [25, 44, 47, 48, 83, 160, 164, 200, 205, 255, 289, 334, 337, 356, 460, 461, 468] or be used toward the solution of boundary value problems [99]. In addition to first-order ODE applications, they are used for solving of differential algebraic equations (DAEs) [8, 88, 89, 91, 92, 187, 202, 203, 263, 272, 273, 277, 279, 280, 305, 339, 345, 400, 450, 451], partitioned half-explicit methods for index-2 DAEs [24, 304], differential equations on manifolds [330], delay differential equations [9, 234, 239, 265, 419, 441], second-order ODEs [10, 14, 111, 113, 153, 156, 168, 213, 216, 224, 231–233, 235, 335, 364–368, 381, 407, 446], Volterra integro-differential equations of the first kind [3], Volterra functional differential equations [447], quadratic ODEs [229] and stochastic differential equations [58]. In certain contexts, DIRK methods may be useful as starting algorithms to implicit methods having a multistep character [101]. They may also be used in interval computations [298] and in the design functionally fitted [333] and block methods [98].

In the context of general linear methods (GLMs), the basic DIRK structure forms the basis of certain two-step Runge-Kutta methods [32, 243], diagonally implicit multistage integration methods (DIMSIMs) [66], diagonally implicit single-eigenvalue methods (DIMSEMs) [140], Multistep Runge-Kutta [55], Almost Runge-Kutta [74], inherent Runge-Kutta stability methods (IRKS) [67], and second derivative IRKS methods [73]. For the class of methods known as multiderivative Runge-Kutta (Turán) methods, two-derivative methods using a DIRK-type structure have recently been derived [430]. Parallel DIRK (PDIRK) methods [112, 131, 151, 216, 217, 226, 237] also exist but are not generally composed of lower triangular A matrices and are, therefore, not of immediate concern here as are diagonally split Runge-Kutta (DSRK) methods [34].

Beginning with the implicit Euler method, implicit midpoint rule, trapezoidal rule and the Hammer-Hollingsworth “exact” trapezoidal rule, to the first effectively multistage DIRK-type method introduced nearly 40 years ago by Butcher [60], much has been learned about IRK methods, in general, and DIRK-type methods in particular. Applied to stiff problems, high stage-order, as opposed to classical order, stiff accuracy and L-stability have proven particularly useful [194]. Although DIRK-type methods may be constructed to be stiffly-accurate and L-stable, their stage-order may not exceed two. This fundamentally limits their utility to settings requiring only low to moderate error tolerances. As most technological applications do not require exceedingly stringent error tolerances, this limitation may often be inconsequential. If a stage-order of three or greater is desired, fully implicit Runge-Kutta (FIRK) methods may be considered but are substantially more expensive to implement. Hence, DIRK-type methods represent a line in sand that one crosses only when high stage-order is worth the additional implementation cost or moving to FIRKs.
or multistage methods with a multistep or multiderivative character. Surprisingly, the majority of papers on DIRK-type methods have focused on stage-order one methods rather than the potentially more useful stage-order two methods. Another common problem with methods offered in the past is the focus on some particular attribute or attributes while ignoring the many others of relevance. Many published schemes do not include error estimators, dense output, or stage-value predictors. Arguably, what is needed are methods that are optimized across a broad spectrum of characteristics. Invariably, requiring one attribute may preclude another; hence, priorities must be established. For example, with DIRK-type methods, one may impart algebraic stability to the method or stage-order two, but not both. Other properties like regularity are largely beyond the control of the scheme designer; therefore, little effort might be expended trying to improve this aspect of the method. Order reduction from stiffness, spatial boundaries of space-time partial differential equations (PDEs), or nonsmoothness favors higher stage-order methods. Although most applications of DIRK-type methods involve stiffness, implicitness may be desired even in the absence of stiffness. Symplectic and symmetric methods favor implicit schemes but constitute very specialized applications [188].

The goal of this paper is to comprehensively summarize the characteristics and assess the potential and limitations of DIRK-type methods for first-order ODEs in an applied fashion. This review establishes what matters, what has been done, what is possible and at what cost; then, scheme coefficients are derived for methods with two to seven implicit stages and orders primarily from three to five. Methods of order greater than five seem of dubious value when applied to stiff problems if the stage order may be no greater than two. It is hoped that these new schemes represent nearly optimal DIRK-type methods for moderate error tolerance, general-purpose contexts. While optimal methods are useful, the overall solution procedure will be suboptimal without careful consideration of implementation issues. No substantial focus is made on designing custom methods for niche applications like symplectic or high phase-order methods. It is beyond the scope of the current work to properly place DIRK-type methods within the broader class of IRK methods, or further, within the still broader class of multistep, multistage, multiderivative methods.

This review is organized beginning with a brief introduction in section 1. Section 2 reviews over 20 important aspects of DIRK-type methods relevant to proper design and application of such methods. These include the general structure, order conditions, simplifying assumptions, error, linear stability, nonlinear stability, internal stability, dense output, conservation, symplecticity, symmetry, dissipation and dispersion accuracy, memory economization, regularity, boundary and smoothness order reduction, efficiency, solvability, implementation, step-size control, iteration control, stage-value predictors, discontinuities and existing software. It is hoped that upon considering each of these aspects that new and existing methods may be placed in a broad and proper perspective. Special attention is paid to the topic of stage-value predictors because of the prospect of considerable cost savings from the use of well designed predictors. However, a complete investigation of different approaches to stage value predictors, including methods, is beyond the scope of this paper. The same is true for dealing with discontinuities. In section 3, early activ-
ity on the topic of DIRK-type methods is summarized. Sections 4–9 focus on both new and existing methods with two, three, four, five, six and seven implicit stages. Due to the large numbers of possible methods, detailing all coefficients of each scheme is not feasible; however, abscissae are often given as they are the most difficult to determine. From these methods, a small group of nearly complete methods (minimally, those with error estimators and dense output) are offered for general-purpose situations and compared against what is arguably the standard method of this class: SDIRK4 [193]. Singular perturbation test problems for both new and existing DIRK-type methods are given in section 10 along with references to other tests of DIRK-type methods. A discussion of test results and conclusions are presented in sections 11 and 12, respectively. A comprehensive reference list and three appendixes are included; appendix A lists the relevant order conditions, appendix B lists dense-output coefficients and appendix C lists method properties. Citations to literature will be made both for those works that are immediately relevant to the topic at hand and also those needed for further reading. No distinction will be made between the two so as to avoid unnecessary distraction. Scheme coefficients are provided to at least 25 digits of accuracy using Mathematica [453, 454].

2 Background

DIRK-type methods are used to solve ODEs of the form

\[
\frac{dU}{dt} = F(t, U(t)), \quad U(a) = U_0, \quad t \in [a, b]
\]  

(1)

and are applied over \(s\)-stages as

\[
\begin{align*}
F_i &= F_i(t_i, U_i), \quad U_i = U^{[n]} + (\Delta t) \sum_{j=1}^{s} a_{ij} F_j, \quad t_i = t^{[n]} + c_i \Delta t, \\
U^{[n+1]} &= U^{[n]} + (\Delta t) \sum_{i=1}^{s} b_i F_i, \quad \hat{U}^{[n+1]} = U^{[n]} + (\Delta t) \sum_{i=1}^{s} \hat{b}_i F_i.
\end{align*}
\]

(2)

where \(i = 1, 2, \ldots, s\), \(F_i = F_i^{[n]} = F(U_i, t^{[n]} + c_i \Delta t)\). Here, \(\Delta t > 0\) is the step-size, \(U^{[n]} \simeq U(t^{[n]})\) is the value of the \(U\)-vector at time step \(n\), \(U_i = U_i^{[n]} \simeq U(t^{[n]} + c_i \Delta t)\) is the value of the \(U\)-vector on the \(i\)th-stage, and \(U^{[n+1]} \simeq U(t^{[n]} + \Delta t)\). Both \(U^{[n]}\) and \(U^{[n+1]}\) are of order \(p\). The \(U\)-vector associated with the embedded scheme, \(\hat{U}^{[n+1]}\), is of order \(\hat{p}\). This constitutes a \((p, \hat{p})\) pair. Each of the respective Runge-Kutta coefficients \(a_{ij}\) (stage weights), \(b_i\) (scheme weights), \(\hat{b}_i\) (embedded scheme weights), and \(c_i\) (abscissae or nodes), \(i, j = 1, 2, \ldots, s\) are real and are constrained, at a minimum, by certain order of accuracy and stability considerations.

DIRK-type methods may be divided into several classes [283]. In the ensuing acronyms, S, E and Q denote singly, explicit first stage, and quasi, respectively. Representative four-stage DIRK, SDIRK and QSDIRK methods appear in Butcher
of a that sometimes with stage-order two methods, choices will be discussed throughout section 2. QESDIRKs are motivated by the fact iterative solution of the stage and step equations. Various trade-offs between these the stability function. It also provides for a constant iteration matrix during the between (E)SDIRKs and (E)DIRKs when *s* reduces the number of implicit stages to *s* having an explicit first stage is primarily to allow stage-order two methods. It also with the circumflex ˆ used to denote the embedded method. The motivation for a truncated version of the row simplifying assumptions so as to mimic stage-order three behavior. A QSDIRK is similar to a QESDIRK except one would try to mimic stage-order two behavior. These methods might be implemented by using a device by Krogh [274]. One may also investigate the merits and consequences of allowing the *a*~*ij*~ to vary with QESDIRKs [227]. A further constraint on the coefficients to

\[
\begin{array}{cccc|cccc|cccc}
 c_1 & a_{11} & 0 & 0 & 0 & c_1 & \gamma & 0 & 0 & 0 & c_1 & 0 & 0 & 0 \\
 c_2 & a_{21} & a_{22} & 0 & 0 & c_2 & a_{21} & \gamma & 0 & 0 & c_2 & a_{21} & \gamma & 0 \\
 c_3 & a_{31} & a_{32} & a_{33} & 0 & c_3 & a_{31} & a_{32} & \gamma & 0 & c_3 & a_{31} & a_{32} & \gamma \\
 c_4 & a_{41} & a_{42} & a_{43} & a_{44} & c_4 & a_{41} & a_{42} & a_{43} & \gamma & c_4 & a_{41} & a_{42} & a_{43} & \gamma \\
 \hline
 b_1 & b_2 & b_3 & b_4 & b_1 & b_2 & b_3 & b_4 & b_1 & b_2 & b_3 & b_4 \\
 \hline
 \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \hat{b}_4 & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \hat{b}_4 & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \hat{b}_4 \\
\end{array}
\]

With an explicit first stage, from left to right, EDIRK, ESDIRK and QESDIRK methods are given by

\[
\begin{array}{cccc|cccc|cccc}
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 c_2 & a_{21} & a_{22} & 0 & 0 & c_2 & a_{21} & \gamma & 0 & 0 & c_2 & a_{21} & a_{22} & 0 \\
 c_3 & a_{31} & a_{32} & a_{33} & 0 & c_3 & a_{31} & a_{32} & \gamma & 0 & c_3 & a_{31} & a_{32} & \gamma \\
 c_4 & a_{41} & a_{42} & a_{43} & a_{44} & c_4 & a_{41} & a_{42} & a_{43} & \gamma & c_4 & a_{41} & a_{42} & a_{43} & \gamma \\
 \hline
 b_1 & b_2 & b_3 & b_4 & b_1 & b_2 & b_3 & b_4 & b_1 & b_2 & b_3 & b_4 \\
 \hline
 \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \hat{b}_4 & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \hat{b}_4 & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \hat{b}_4 \\
\end{array}
\]

Not all authors have followed these conventions. Obviously, SDIRK, QSDIRK, EDIRK, ESDIRK and QESDIRK methods are subsets of DIRK methods. In the case of EDIRK, ESDIRK and QESDIRK methods, some authors prefer to decompose *A* = *a*~*ij*~, *b* = *b*~*i*~ and *c* = *c*~*i*~, into

\[
\begin{pmatrix}
 c \\
 A \\
 b^T
\end{pmatrix} =
\begin{pmatrix}
 0 & 0 & 0^T \\
 \hat{c} & a & \hat{A} \\
 b_1 & \hat{b}^T
\end{pmatrix}
\text{ with } \quad A =
\begin{pmatrix}
 0 & 0^T \\
 b_1 & \hat{A}
\end{pmatrix}
\]

(3)

where 0^T (composed of zeros), *a*^T*, *c*^T* and *b*^T* are vectors of length *(s-1)*, \{0, *c*^T*\} = *a*\_1, \{0, *c*^T*\} = *c*^T*, and \{*b*\_1, *b*^T*\} = *b*^T* and \hat{A} is a square matrix of dimension *(s-1) × (s-1)*. Hence, although *A* is not invertible, \hat{A} is often invertible. The superscript ˆ will be used to denote this decomposition and should not be confused with the circumflex ˆ used to denote the embedded method. The motivation for having an explicit first stage is primarily to allow stage-order two methods. It also reduces the number of implicit stages to *s*\_I = *s* – 1. Note that there is no distinction between (E)SDIRKs and (E)DIRKs when *s*\_I = 1. Forcing each diagonal element, *a*\_ii = \gamma, greatly facilitates designing A- and L-stable methods by virtue of simplifying the stability function. It also provides for a constant iteration matrix during the iterative solution of the stage and step equations. Various trade-offs between these choices will be discussed throughout section 2. QESDIRKs are motivated by the fact that sometimes with stage-order two methods, *c*\_2 can get unpleasantly large, and, for these methods, the leading order error of the second stage is driven by the magnitude of *a*\_22. The goal would be to reduce *a*\_22, as much as possible, below \gamma while applying a truncated version of the row simplifying assumptions so as to mimic stage-order three behavior. A QSDIRK is similar to a QESDIRK except one would try to mimic stage-order two behavior. These methods might be implemented by using a device by Krogh [274]. One may also investigate the merits and consequences of allowing the *a*\_ii~ to vary with QESDIRKs [227]. A further constraint on the coefficients to
these methods that is both simple and beneficial is the stiffly-accurate assumption of Prothero and Robinson [340], \( a_{ij} = b_j \), implying \( c_s = 1 \). Explicit first-stage methods using the stiffly-accurate assumption are sometimes called first same as last (FSAL) methods [209, 324, 396, 397]. Albrecht [4] considers these composite linear multistep methods. If a method has at least two identical values of \( c_i \), then it is called confluent, otherwise it is nonconfluent. Unlike reducible methods, methods that are DJ- (Dahlquist-Jeltsch [125]) or (H)S- (Hundsdorfer-Spijker [220]) irreducible cannot be manipulated into an equivalent method having fewer stages.

To identify certain schemes derived in this paper, a nomenclature similar to that originally devised by Dormand and Prince is followed [132,255]. For those schemes that are given names, they will be named DIRK\( p (\hat{p} I) s S[q] X, x \), where \( p \) is the order of the main method, \( \hat{p} \) is the order of the embedded method, \( I \) is included if internal error-control is offered, \( s \) is the number of stages, \( S \) is some stability characterization of the method, \( q \) is the stage-order of the method, \( X \) is used for any other important characteristic of the method, and \( x \) distinguishes between family members of some type of method.

### 2.1 Order Conditions

Order conditions for Runge-Kutta methods may be derived by using the theory of Runge-Kutta trees. For the DIRK-type methods considered in this work, one focuses on ODE trees [61, 64, 190, 286]. From these, one may construct the corresponding equations of condition or, as is more commonly stated, the order conditions. Due to the complex and tedious nature of deriving these order conditions, the task is sometimes done computationally using symbolic manipulation software [43, 90, 145, 163, 406]. Expressions for the equation of condition associated with the \( p \)th-order trees are of the form

\[
\tau_j^{(p)} = \frac{1}{\sigma} \Phi_j^{(p)} - \frac{\alpha}{p!} = \frac{1}{\sigma} \left( \Phi_j^{(p)} - \frac{1}{\gamma} \right), \quad \Phi_j^{(p)} = \sum_{i=1}^{s} b_i \Phi_{i,j}^{(p)}, \quad \alpha \sigma \gamma = p!, \quad (4)
\]

where \( \Phi_{i,j}^{(p)} \) and \( \Phi_j^{(p)} \) are Runge-Kutta coefficient products and their sums, and \( j \) represents the index of the order condition, i.e., at each order, there may have more than one order condition. A Runge-Kutta method is said to be of order \( p \) if the local truncation error satisfies

\[
U^{[n]} - U(t^{[n]}) = \mathcal{O} (\Delta t)^{p+1}. \quad (5)
\]

To achieve this, all order conditions must be satisfied up to and including order \( p \). The number of order conditions introduced at order \( p \), \( a_p \), is given by [64, 190]

\[
\sum_{p=1}^{\infty} a_p x^{p-1} = \prod_{p=1}^{\infty} (1 - x^p)^{-a_p}. \quad (6)
\]

For orders \( p = \{1, 2, 3, 4, 5, 6\} \), \( j \) ranges from 1 to \( a_p = \{1, 1, 2, 4, 9, 20\} \), respectively (see Sloane and Plouffe [401]; sequence A000081, formerly M1180 and N0454). All
methods constructed later in this paper use the row-sum convention, $Ae = c$, where $e = \{1, 1, \cdots, 1\}^T$. Verner [433] distinguishes four distinct types of order conditions for ODE contexts: quadrature, subquadrature, extended subquadrature and nonlinear conditions. Moving from order $p - 1$ to order $p$, one new quadrature condition is introduced, $(p - 2)$ for $p \geq 3$ new subquadrature conditions and $(2p^2 - p + 1)$ for $p \geq 4$ new extended subquadrature conditions are added. Nonlinear order conditions do not arise until fifth-order and number $a_p - (2p^2 - p), p \geq 5$.

Quadrature conditions allow solution of vector equations of the form $y'(t) = f(t)$ and may be written at order $k$ as

$$\tau_Q^{(k)} = \frac{1}{(k-1)!} b^T C^{k-1} e - \frac{1}{k!}, \quad k = 1, 2, \cdots, p, \quad (7)$$

where $C = \text{diag}(c)$ or $Ce = c$. Componentwise multiplication of two vectors, $a$ and $b$ will be denoted as $a \ast b$. Powers of the vector $c$ should be interpreted as componentwise multiplication. Hence $c^3 = c \ast c \ast c = C^3 e$ and $c^0 = e$. Powers of the $A$-matrix are given by $A^0 = I$, $A^1 = A$ and $A^2 = AA$.

Trees associated with quadrature conditions are often referred to as bushy trees. Subquadrature conditions permit solution of linear, nonhomogeneous, vector equations, $y'(t) = Ay(t) + f(t)$, where $A$ is a constant. They may be written at order $k + r$ as

$$\tau_{SQ}^{(k+r)} = \frac{1}{(k-1)!} b^T A^r C^{k-1} e - \frac{1}{(k+r)!}, \quad k - 1, r \geq 0, \quad 1 \leq k + r \leq p. \quad (8)$$

An important subset of the subquadrature conditions are the tall trees where $k = 1$; $\tau_{SQ_{all}}^{(r+1)} = b^T A^r e - 1/(r+1)!$, associated with the equation $y'(t) = Ay(t)$. Setting $r = 0$ recaptures the quadrature order conditions in (8). Extended subquadrature conditions extend these conditions to linear nonautonomous, nonhomogeneous vector equations, $y'(t) = A(t)y(t) + f(t)$. Defining $K = k + \sum_{j=0}^{m} k_j$ and $\kappa_i = \sum_{j=0}^{i} k_j$, at order $K$ they are

$$\tau_{ESQ}^{(K)} = \frac{1}{\sigma} b^T C^{k_0-1} AC^{k_1-1} \cdots AC^{k_m-1} e - \frac{\alpha}{K!}, \quad k, k_i > 0 \quad (9)$$

for $1 \leq K \leq p$ with $\sigma = (k_0 - 1)!(k_1 - 1)! \cdots (k_m - 1)!(k - 1)!$, $\gamma = (K - \kappa_m)(K - \kappa_{m-1}) \cdots (K - \kappa_0)K$ and $\alpha = K! / \sigma \gamma$. Setting $k_0 = k_1 = \cdots = k_m = 1$ retrieves the subquadrature conditions. All conditions not contained within these three categories are nonlinear order conditions. Their general form, given by Verner [433], is quite complicated and will not be given here. Note that at lower orders, extended subquadrature conditions collapse into subquadrature conditions ($k_i = 1, m = r$), and subquadrature conditions collapse into quadrature conditions ($r = 0$). Hence, even though the first nonlinear condition does not appear until fifth-order, lower-order methods retain their formal order on nonlinear problems. Appendix A lists Runge-Kutta order conditions up to sixth-order. Rather than increasing the order of method by considering the order conditions at the next order, Faragó et al. [146] attempt to achieve this using Richardson extrapolation.
Similar conditions apply to the embedded and dense output methods where $\hat{b}$ and $b(\theta)$ (See §2.7) replace $b$ in all of the preceding expressions. Another important aspect of order is the stage-order of a method, discussed in the following sections. For the stages, order conditions resemble those for the overall method (step conditions versus stage conditions)

$$t^{(p)}_{i,m} = \frac{1}{\sigma} \gamma^{(p)}_{i,m} - \frac{\alpha}{p!} c^p_i, \quad \gamma^{(p)}_{i,m} = \sum_{j=1}^{s} a_{ij} \Phi^{(p)}_{j,m}, \quad \alpha \sigma \gamma = p!. \quad (10)$$

### 2.2 Simplifying Assumptions

Simplifying assumptions are often made to facilitate the solution of Runge-Kutta order conditions and possibly to enforce higher stage-order. They were devised by Butcher [59, 60, 129, 190, 193] based on the structure of Runge-Kutta trees and may provide insight into particular methods. The three common ones are

- $B(p)$:
  $$\sum_{j=1}^{s} b_{c_i} c_i^{k-1} = \frac{1}{k}, \quad k = 1, \ldots, p, \quad (11)$$

- $C(\eta)$:
  $$\sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{c_j^k}{k}, \quad k = 1, \ldots, \eta, \quad (12)$$

- $D(\zeta)$:
  $$\sum_{i=1}^{s} b_{c_i} c_i^{k-1} a_{ij} = \frac{b_j}{k} (1 - c_j^k), \quad k = 1, \ldots, \zeta. \quad (13)$$

A fourth simplifying assumption, $E(\eta, \zeta)$, is not needed for DIRK-type methods [129]. Recently, Khashin [258, 434] has given simplifying strategies substantially different from those given by Butcher. However, they are only likely to offer new insight into methods having vastly different step and stage orders such as high-order explicit Runge-Kutta methods.

The first, (11), is related to the quadrature conditions. Enforcing $B(p)$, (7), forces $\tau_{Q}^{(k)} = 0, \quad k = 1, 2, \ldots, p$ for all orders up to and including $p$. Assumptions $C(\eta)$ and $D(\zeta)$ are sometimes referred to as the row and column simplifying assumptions, respectively. The stage-order of a Runge-Kutta method is the largest value of $q$ such that $B(q)$ and $C(q)$ are both satisfied. As its name implies, stage-order is related to the order of accuracy of the intermediate stage values of the $U$-vector, $U_i$, and equals the lowest order amongst all stages. Stage-order strongly influences the accuracy of methods when applied to stiff problems, boundary and smoothness order-reduction, error-estimate quality and stage-value predictor quality. While virtually all DIRK-type methods in the literature are at least stage-order one by virtue of satisfying the row-sum condition $Ae = c$ (Zlatev [466] does not), fewer satisfy $C(2)$. This requires that $a_{11} = 0$ and forces $a_{21} = a_{22}$. Hence, only EDIRK, ESDIRK and QESDIRK methods have the potential for a stage-order of two. It is impossible to satisfy $C(3)$ for DIRK-type methods because of the second stage where $q_2^{(3)} = 4a_{22}^3/3$, which implies that $a_{22} = 0$. 

10
Closely related to the assumption \( B(p) \), \( p^{(k)} \) is defined as
\[
p^{(k)} = b^T C^{k-1} e - \frac{1}{k}
\]  
where \( p^{(k)} = \tau^k[(k-1)!]. \) Similarly, \( q^{(k)} \) is closely related to row-simplifying assumption \( C(\eta) \) given in (12),
\[
q^{(k)} = AC^{k-1} e - \frac{1}{k} C^k e, \quad q^{(1)} = Ae - c
\]  
where \( q^{(1)} = 0 \) is simply the row-sum condition. Writing \( C(k,i) \) is meant to imply \( q^{(k)}_i = 0. \) With this, one may write the stage-order vector (SOV) equals \{p_1, p_2, \ldots, p_s \} where \( p_i \) is the largest value of \( k \) at each stage \( i \) for which \( q^{(k)}_i = 0, k = 1, 2, \ldots, p_i. \) For explicit first-stage methods, \( p_1 = p \) by convention. The stage-order may be seen to be \( q = \text{Min}\{p_1, p_2, \ldots, p_s \}. \) Slightly less demanding, the dominant stage-order (DSO) is the minimum value of \( p_1 \) such that the \( i \)-th stage satisfies \( b_i \neq 0 \) [433]. Cameron et al. [91] call this property the forward quasi-stage-order. Related to this, they also define a reverse quasi-stage-order using \( q^{(k)} = A^{-1} q^{(k)}. \)

Using \( q^{(k)} \), subquadrature (8) and extended subquadrature (9), conditions may be recast as
\[
\tau^{(k+r)}_{S\text{SQ}} = \frac{1}{(k-1)!} b^T A^{r-1} q^{(k)} + \frac{1}{k!} b^T A^{r-1} C^k e - \frac{1}{(k+r)!},
\]
\[
\tau_{E\text{SQ}}^{(K)} = \frac{1}{\sigma} b^T C^{k_0-1} AC^{k_1-1} \ldots AC^{k_{m-1}} q^{(k)}
\]
\[
+ \frac{1}{k \sigma} b^T C^{k_0-1} AC^{k_1-1} \ldots AC^{k_{m-1}} C^k e - \frac{\alpha}{K!}.
\]

One may apply \( q^{(k)} \) recursively to \( \tau_{E\text{SQ}}^{(K)} \) just as has been done with \( \tau^{(k+r)}_{S\text{SQ}}. \) Table 1 details the use of \( q^{(k)} \), and consequently \( C(\eta) \), for order conditions up to order six.

From a general discussion of simplifying conditions, a detailed analysis must be done to understand how they should be applied to DIRK-type methods. The goal is usually to force \( q^{(k)} = k = 1, 2, \ldots, \eta \) to vanish by imposing \( C(\eta) \), thereby satisfying many order conditions that might have been otherwise difficult to solve and, simultaneously, to increase the stage order. Focusing on fifth-order methods, it may be seen from Table 1 that employing \( B(5) \) ( \( \tau^{(1,2,3,4,5)}_1 = 0 \) ) and \( C(2) \) ( \( q^{(1,2)} = 0 \) with \( a_{ij} = 0, j = 1, 2, \ldots, s) \) reduces the unsatisfied order conditions to \( \tau^{(3)}_3 = \tau^{(5)}_4 = 0 \) or \( b^T q^{(3)} = b^T C q^{(3)} = b^T A q^{(3)} = b^T q^{(4)} = 0. \) For moderate order DIRK-type methods, only \( q^{(k)} \), \( k = 1, 2, 3 \) will be needed. If \( q^{(3)} \) is applied, it is usually used with an EDIRK, ESDIRK, or QESDIRK as \( q_i^{(3)} = 0, i = 3, 4, \ldots, s \) but \( q_2^{(3)} \neq 0. \) Because \( q_2^{(3)} \neq 0, \) the use of assumption \( C(3) \) in the design of EDIRK,
Table 1. Order conditions expressed using $q^{(k)}, k = 2, 3, 4, 5$ up to sixth-order for Runge-Kutta methods. Bushy tree order conditions $\tau_{1}^{(l)}, l = 1, 2, \cdots, 6$ are not included. Order conditions are denoted as to whether they are subquadrature (SQ), extended subquadrature (ESQ) or nonlinear (NL).

<table>
<thead>
<tr>
<th></th>
<th>$\tau_{2}^{(3)} = b^T q^{(2)} + 7_{1}^{(3)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQ</td>
<td>$\tau_{2}^{(3)} = b^T q^{(2)} + 7_{1}^{(3)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{2}^{(3)} = b^T b_{1}^{T} + 37_{1}^{(3)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_{2}^{(4)} = b^T q^{(3)} + 7_{1}^{(3)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_{2}^{(4)} = b^T Aq^{(2)} + 7_{3}^{(3)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{2}^{(6)} = b^T C^2 q^{(2)} + 67_{1}^{(3)}$</td>
</tr>
<tr>
<td>NL</td>
<td>$\tau_{2}^{(6)} = 2b^T (q^{(2)})^2 + b^T C^2 q^{(2)} + 37_{1}^{(3)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{4}^{(3)} = b^T C q^{(3)} + 47_{1}^{(3)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_{5}^{(5)} = b^T q^{(4)} + 7_{1}^{(3)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{5}^{(5)} = b^T C A q^{(2)} + 7_{5}^{(3)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{7}^{(5)} = b^T A C q^{(2)} + 37_{5}^{(3)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_{8}^{(5)} = b^T A q^{(3)} + 7_{5}^{(3)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_{9}^{(5)} = b^T A A q^{(2)} + 7_{8}^{(3)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{10}^{(6)} = b^T C^3 q^{(2)} + 107_{1}^{(3)}$</td>
</tr>
<tr>
<td>NL</td>
<td>$\tau_{10}^{(6)} = b^T C (q^{(2)})^2 + b^T C^3 q^{(2)} + 157_{1}^{(3)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{11}^{(6)} = b^T C^2 q^{(3)} + 107_{1}^{(3)}$</td>
</tr>
<tr>
<td>NL</td>
<td>$\tau_{11}^{(6)} = b^T Q^{(2)} A q^{(2)} + b^T Q^{(2)} q^{(3)} + 16 b^T C^3 q^{(2)} + 7_{1}^{(3)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{12}^{(6)} = b^T C A q^{(3)} + 7_{6}^{(3)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{13}^{(6)} = b^T A C a^{(2)} + 37_{6}^{(3)}$</td>
</tr>
<tr>
<td>NL</td>
<td>$\tau_{13}^{(6)} = b^T A (q^{(2)})^2 + b^T A C^2 q^{(2)} + 37_{6}^{(3)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{14}^{(6)} = b^T A C q^{(3)} + 47_{6}^{(3)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_{15}^{(6)} = b^T A q^{(4)} + 7_{6}^{(3)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{16}^{(6)} = b^T C A A q^{(2)} + 7_{13}^{(3)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{17}^{(6)} = b^T A C A q^{(2)} + 7_{13}^{(3)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{18}^{(6)} = b^T A A A q^{(2)} + 37_{15}^{(3)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_{19}^{(6)} = b^T A A q^{(3)} + 7_{15}^{(3)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_{20}^{(6)} = b^T A A A A q^{(2)} + 7_{19}^{(3)}$</td>
</tr>
</tbody>
</table>
ESDIRK and QESDIRK methods will be referred to as truncated. To complete the solving of the order conditions with this assumption, one need only solve \( b_2 q_2^{(3)} = b_2 c_2 q_2^{(3)} = \sum_{i=1}^{s} b_i a_{i2} q_2^{(3)} = 0 \) and \( b^T q^{(4)} = 0 \) or \( \tau_s^{(5)} = 0 \). The first three are solved by setting \( b_2 = \sum_{i=1}^{s} b_i a_{i2} = 0 \). The SOV is \( \{p, 2, 3, 3, \cdots, 3\} \) so that the stage-order is two but the DSO is then three. For DIRK and SDIRK methods, if \( q^{(2)} \) is applied, it is used with \( a_{11} \neq 0 \) as \( q_i^{(2)} = 0, i = 2, 3, \cdots, s \) and \( b_1 = 0 \) because \( q_i^{(2)} = a_{i1}^2 / 2 \neq 0 \). The SOV is \( \{1, 2, 2, \cdots, 2\} \) and the stage order is one. Verner (§2) [433] rewrites (16) and (17) under the assumption that the order conditions are satisfied (\( \tau_{j}^{(p)} \) vanishes), as well as certain other conditions at the same order, giving \( b^T A^{r-1} q^{(k)} = 0 \) and \( b^T C^{k_0-1} A C^{k_1-1} \cdots A C^{k_m-1} q^{(k)} = 0 \). Using this abbreviated format, Verner writes the fifth- and sixth-order nonlinear conditions as \( b^T (q^{(2)})^2 = b^T C (q^{(2)})^2 = b^T (q^{(2)} * q^{(3)}) = b^T (q^{(2)} * Aq^{(2)}) = b^T A (q^{(2)})^2 = 0 \). The extended form of these conditions is given in appendix A. Because row simplifying conditions do not contain \( b \), they affect the main, embedded and dense-output methods identically as well as certain stage-value predictors.

Closely related to assumption \( D(\zeta) \) given in (13), one may further define

\[
\mathbf{r}^{(k_0)} = b^T C^{k_0-1} A - \frac{1}{k_0} b^T \left( I - C^{k_0} \right).
\]  

Writing \( D(k, j) \) is meant to imply \( r_j^{(k)} = 0 \). Instead of operating on the top of the trees with \( q_j^{(k)} = 0 \), one may modify the bottom of the tree by using \( r_j^{(k)} = 0 \). With these assumptions, the subquadrature (8) and extended subquadrature (9) conditions are rewritten as

\[
\begin{align*}
\tau_{SQ}^{(k+r)} &= \frac{1}{(k-1)!} \mathbf{r}^{(1)} A^{r-1} C^{k-1} \mathbf{e} + \frac{1}{(k-1)!} b^T A^{r-1} C^{k-1} \mathbf{e} \\
&\quad - \frac{1}{(k-1)!} b^T C A^{r-1} C^{k-1} \mathbf{e} - \frac{1}{(k+r)!}, \\
\tau_{ESQ}^{(K)} &= \frac{1}{\sigma} \mathbf{r}^{(k_0)} C^{k_1-1} \cdots C^{k_m-1} A C^{k_1-1} \mathbf{e} + \frac{1}{k_0 \sigma} b^T C^{k_1-1} \cdots C^{k_m-1} A C^{k_1-1} \mathbf{e} \\
&\quad - \frac{1}{k_0 \sigma} b^T C^{k_0+k_1-1} \cdots C^{k_m-1} A C^{k_1-1} \mathbf{e} - \frac{\alpha}{k!}.
\end{align*}
\]  

As with the row simplifying assumption, \( \mathbf{r}^{(i)} \) may be applied recursively to \( \tau_{SQ}^{(k+r)} \) and \( \tau_{ESQ}^{(K)} \). When conditions required for \( D(k_0) \) are enforced, then \( \mathbf{r}^{(i)} = 0, i = 1, 2, \cdots, k_0 \). From (19), \( k_0 = 1 \) and (20), the original order condition splits into two conditions: one of order \( k_0 \) less than the original condition and the other of the same order. More specifically, for orders up to six, order conditions are expressed in Table 2 where \( \mathbf{r}^{(k_0)} \) is used. Note that most of the nonlinear order condition cannot be reduced by using the column simplifying assumption.

At orders of five and below, Table 2 suggests that only \( \mathbf{r}^{(1)} \) or \( k_0 = 1 \) is profitable to consider for DIRK-type-methods. Several authors [18,116] have, however, used \( D(2) \) and \( a_{ss} = 0 \). Obviously, one could combine the row and column simplifying assumptions where appropriate. For many of the DIRK-type methods designed in this
Table 2. Order conditions expressed using $r^{(k_0)}$ up to sixth-order for Runge-Kutta methods. Bushy tree order conditions $\tau_1^{(l)}, l = 1, 2, \ldots, 6$ are not included. Stiffly accurate DIRK-type implicit methods cannot use the column simplifying assumption. Order conditions are denoted as to whether they are subquadrature (SQ), extended subquadrature (ESQ) or nonlinear (NL).

<table>
<thead>
<tr>
<th></th>
<th>$\tau_2^{(3)} = r^{(1)}Ce + \tau_1^{(2)} - 2\tau_1^{(3)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQ</td>
<td>$\tau_2^{(4)} = r^{(2)}Ce + \frac{1}{2}\tau_1^{(2)} - 3\tau_1^{(4)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_3^{(4)} = \frac{1}{2}r^{(1)}C^2e + \frac{1}{7}\tau_1^{(3)} - 3\tau_1^{(4)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_4^{(4)} = r^{(1)}ACe + \tau_2^{(3)} - \tau_2^{(4)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_5^{(6)} = \frac{1}{2}r^{(3)}Ce + \frac{1}{5}\tau_1^{(2)} - 4\tau_1^{(5)}$</td>
</tr>
<tr>
<td>NL</td>
<td>$\tau_3^{(5)}$ - No Simplification</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_4^{(5)} = \frac{1}{2}r^{(2)}C^2e + \frac{1}{7}\tau_1^{(3)} - 6\tau_1^{(5)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_5^{(5)} = \frac{1}{7}r^{(1)}C^3e + \tau_1^{(4)} - 4\tau_1^{(5)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_6^{(5)} = r^{(2)}ACe + \frac{1}{7}\tau_2^{(3)} - \tau_2^{(5)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_7^{(5)} = r^{(1)}CACe + \tau_2^{(4)} - 2\tau_2^{(5)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_8^{(5)} = \frac{1}{2}r^{(1)}AC^2e + \tau_3^{(4)} - \tau_3^{(5)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_9^{(5)} = r^{(1)}AACe + \tau_4^{(4)} - \tau_5^{(6)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_10^{(6)} = \frac{1}{2}r^{(4)}Ce + (1/24)\tau_1^{(2)} - 5\tau_1^{(6)}$</td>
</tr>
<tr>
<td>NL</td>
<td>$\tau_3^{(6)}$ - No Simplification</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_11^{(6)} = \frac{1}{2}r^{(3)}C^2e + (1/6)\tau_1^{(3)} - 10\tau_1^{(6)}$</td>
</tr>
<tr>
<td>NL</td>
<td>$\tau_5^{(6)}$ - No Simplification</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_6^{(6)} = \frac{1}{2}r^{(2)}Ce + (1/2)\tau_1^{(4)} - 10\tau_1^{(6)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_7^{(6)} = \frac{1}{21}r^{(1)}C^4e + \tau_1^{(5)} - 5\tau_1^{(6)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_8^{(6)} = \frac{1}{2}r^{(3)}ACe + (1/6)\tau_2^{(3)} - \tau_2^{(6)}$</td>
</tr>
<tr>
<td>NL</td>
<td>$\tau_9^{(6)}$ - No Simplification</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{10}^{(6)} = r^{(2)}CACe + (1/2)\tau_2^{(4)} - 3\tau_2^{(6)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{11}^{(6)} = \frac{1}{2}r^{(1)}C^2ACe + \tau_2^{(5)} - 3\tau_2^{(6)}$</td>
</tr>
<tr>
<td>NL</td>
<td>$\tau_{12}^{(6)} = \frac{1}{2}r^{(1)}(ACe)^2 + \tau_3^{(5)} - \tau_3^{(6)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{13}^{(6)} = r^{(2)}AC^2e + (1/2)\tau_3^{(4)} - \tau_4^{(6)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{14}^{(6)} = \frac{1}{2}r^{(1)}CACe + \tau_4^{(5)} - 2\tau_4^{(6)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_{15}^{(6)} = \frac{1}{6}r^{(1)}AC^3e + \tau_5^{(5)} - \tau_6^{(6)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{16}^{(6)} = r^{(2)}AACe + (1/2)\tau_4^{(4)} - \tau_8^{(6)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{17}^{(6)} = r^{(1)}CAACe + \tau_6^{(5)} - 2\tau_8^{(6)}$</td>
</tr>
<tr>
<td>ESQ</td>
<td>$\tau_{18}^{(6)} = r^{(1)}ACACe + \tau_7^{(5)} - \tau_9^{(6)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_{19}^{(6)} = \frac{1}{2}r^{(1)}AAC^2e + \tau_8^{(5)} - \tau_13^{(6)}$</td>
</tr>
<tr>
<td>SQ</td>
<td>$\tau_{20}^{(6)} = r^{(1)}AAACe + \tau_9^{(5)} - \tau_{16}^{(6)}$</td>
</tr>
</tbody>
</table>
paper, the stiffly-accurate assumption and $C(2)$ will be used. The latter increases the stage-order, reduces the number and nonlinearity of the order conditions and facilitates the design of the embedded method. The column simplifying assumption $D(\zeta)$ will have no use with stiffly-accurate methods because, for $j = s$, one must enforce $r^{(k_0)}_s = b_sc_{k_0-1}a_{ss} - \frac{b_s}{k_s}(1 - c_{k_0}) = 0$. The stiffly-accurate assumption forces $c_s = 1$ and $a_{ss} = b_s$ so that $a_{ss} = b_s = 0$. Hence, this relation cannot be satisfied for stiffly-accurate DIRK-type methods. If one imposes assumption $D(1, j)$ on nonstiffly-accurate methods, then for $j = s$, $b_sa_{ss} - b_s(1 - c_s) = 0$, implying that either $b_s = 0$ or $c_s = 1 - a_{ss}$. If one further assumes $D(2, j)$, then $b_sc_{ss} = \frac{b_s}{2}(1 - c_s^2) = 0$. Avoiding $b_s = 0$, one may set $a_{ss} = 0$, implying $c_s = 1$ [18,116]. For functionally fitted methods, additional order conditions are involved when Runge-Kutta coefficients are allowed to vary with both time and step size. Ozawa [333] constructs functionally fitted, three-stage, stage-order two ESDIRKs of orders three and four having stability domains commensurate with those of explicit methods.

Lastly, simplifying assumptions $B(p)$, $C(\eta)$ and $D(\zeta)$ may be related through $p^{(k)}$, $q^{(k)}$ and $r^{(k_0)}$. Forming the scalar product of $r^{(k_0)}$ and $c^{l-1}$ gives
\[
    r^{(k_0)}c^{l-1} = b^Tc^{k-1}Ac^{l-1} - \frac{1}{k}b^Tc^{l-1} + \frac{1}{k}b^TC^{k+l-1}c^{l-1}
\]
where
\[
    b^Tc^{k-1} = p^{(k)} + \frac{1}{k}, \quad Ac^{k-1} = q^{(k)} + \frac{1}{k}c^{k}. \tag{22}
\]
Applying (22) to (21) gives
\[
    r^{(k_0)}c^{l-1} = b^Tc^{k-1}q^{(l)} + \frac{1}{l}c^{l} - \frac{1}{k}l(p^{(l)} + \frac{1}{l}) + \frac{1}{k}(p^{(k+l)} + \frac{1}{k+l})
\]
\[
    = b^Tc^{k-1}q^{(l)} + \frac{(k + l)}{kl}p^{(k+l)} - \frac{1}{k}p^{(l)}
\]
\[
    = b^Tc^{k-1}q^{(l)} + \frac{(k + l)!}{kl}r^{(k+l)} - \frac{(l - 1)!}{k}r^{(l)}. \tag{23}
\]
This result is essentially due to Butcher [60]. When both row and column simplifications are being applied simultaneously, (23) shows that some redundancies may occur. As these stage-order two methods will be of primary concern, $l = 1, 2$, which implies
\[
    r^{(k)}c = b^Tc^{k-1}q^{(1)} + \frac{(k + 1)!}{k}r^{(k+1)} - \frac{1}{k}r^{(1)}, \tag{24}
\]
\[
    r^{(k)}c = b^Tc^{k-1}q^{(2)} + \frac{(k + 2)!}{2k}r^{(k+2)} - \frac{1}{k}r^{(2)}. \tag{25}
\]
so that, in the latter case,
\[
    r^{(1)}c = b^Tq^{(2)} + 3r^{(3)} - r^{(1)}, \quad r^{(2)}c = b^Tcq^{(2)} + 6r^{(4)} - \frac{1}{2}r^{(2)}. \tag{26}
\]
With stage-order two methods, typically $r^{(1)}c = 0$ and $r^{(2)}c = 0$ since the RHS of the above equations vanish. Imposing either column simplifying assumptions is now easier as one component of $r^{(1)}$ and $r^{(2)}$ will automatically vanish when the other components vanish.
2.3 Error

The error in a \( p \)th-order Runge-Kutta scheme may be quantified in a general way by taking the \( L_2 \) principal error norm,

\[
A^{(p+1)} = \| \tau^{(p+1)} \|_2 = \sqrt{\sum_{j=1}^{2p+1} (\tau_j^{(p+1)})^2}
\]  

(27)

where \( \tau_j^{(p+1)} \) are the \( a_{p+1} \) error coefficients, (6), associated with the classical order of accuracy \( p + 1 \). For single-step, embedded schemes where \( \hat{p} = p - 1 \), additional definitions are useful such as

\[
\hat{\tau}_k^{(\hat{p}+2)} = \frac{1}{\hat{\sigma}} \sum_{i=1}^{s} \hat{b}_i \Phi_i^{(\hat{p})} k_{i,k} - \frac{\alpha}{\hat{p}!}, \quad \hat{A}^{(\hat{p}+1)} = \| \hat{\tau}^{(\hat{p}+1)} \|_2,
\]

(28)

\[
B^{(\hat{p}+2)} = \frac{\hat{A}^{(\hat{p}+2)}}{\hat{A}^{(\hat{p}+1)}}, \quad C^{(\hat{p}+2)} = \frac{\| \hat{\tau}^{(\hat{p}+2)} - \tau^{(\hat{p}+2)} \|_2}{\hat{A}^{(\hat{p}+1)}}, \quad E^{(\hat{p}+2)} = \frac{A^{(\hat{p}+2)}}{\hat{A}^{(\hat{p}+1)}}
\]

(29)

and \( D = \text{Max}\{ |a_{ij}|, |b_i|, |\hat{b}_i|, |c_i| \} \) where the superscript circumflex denotes the values with respect to the embedded method. No analogous measures have been proposed for methods having \( \hat{p} = p - 2 \). The order of the method, \( p \), refers to the global error while the local error is given by \( p + 1 \).

As very stiff ODEs bear a strong resemblance to singular perturbation problems, it is worth noting the value of the stiffly-accurate assumption in this context [187, 281]. If \( p \) is the classical order of the DIRK-type method and \( q \) is the stage-order then the global error for the differential and algebraic variables of the method applied to a singular perturbation problem with stiffness parameter \( \varepsilon \) is given in Table 3 depending on whether or not the method is stiffly-accurate. Hairer and Wanner [187, 193] have assumed that \( A^{-1} \) exists and the linear stability function, discussed in the next section, satisfies \( |R(-\infty)| < 1 \). From the table, it may be seen that the classical order of a given method may not be observed in practice. This is order reduction due to stiffness and its implications for variable step-size codes, including an SDIRK code, are investigated by Mac Donald and Enright [295]. Kvaerno [281, 282] distinguishes three stiffness regimes of typical singular perturbation problems: one where the problem is nonstiff (\( \Delta t < \varepsilon \)), one where the index-2 DAE terms dominate the solution (\( \Delta t \gg \varepsilon \)), and one where the problem behaves like an index-1 DAE (\( \Delta t \gg \varepsilon \)). In a later section on boundary and smoothness order reduction, it

<table>
<thead>
<tr>
<th>Assumption</th>
<th>Differential</th>
<th>Algebraic</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_{si} \neq b_i )</td>
<td>( O((\Delta t)^p) + O(\varepsilon(\Delta t)^{q+1}) )</td>
<td>( O((\Delta t)^{q+1}) )</td>
</tr>
<tr>
<td>( a_{si} = b_i )</td>
<td>( O((\Delta t)^p) + O(\varepsilon(\Delta t)^{q+1}) )</td>
<td>( O((\Delta t)^p) + O(\varepsilon(\Delta t)^q) )</td>
</tr>
</tbody>
</table>
will be seen that order reduction may also occur due to spatial boundary conditions in a discretized PDE or nonsmooth coefficients contained within the right hand side (RHS) of the ODEs. The worst occurrences of this order reduction will result in global convergence rates of the minimum between \( p \) and \( q + 1 \). Several detailed convergence studies \([139,320]\) are also available.

### 2.4 Linear Stability

Linear stability of DIRK-type methods applied to ODEs is studied based on the equation \( U' = \lambda U \) by using the stability function

\[
R(z) = \frac{P(z)}{Q(z)} = \frac{\text{Det}[I - zA + ze \otimes b^T]}{\text{Det}[I - zA]} = 1 + zb^T[I - zA]^{-1}e, \tag{30}
\]

where \( I \) is the identity matrix and \( z = \lambda \Delta t \). Similar expressions may be written for the embedded and dense output methods by simply replacing \( b \) with \( \hat{b} \) and \( b(\theta) \), respectively. If \( t_0 = 0 \) and \( U(t_0) = 1 \), then after one step, \( U = \exp(\lambda \Delta t) = \exp(z) \). Therefore, one would like \( R(z) \) to approximate \( \exp(z) \). The motivation for studying this simple equation when actually trying to solve (1) is based on a two-part simplification \([412]\). One first linearizes (1) to \( U' = AU \) where \( A \) is the Jacobian \( \partial F/\partial U \) at any point \( t^* \) belonging to the time interval under consideration \([69]\).

Next, one selects an eigenvalue \( \lambda \), belonging to the spectrum of \( A \), where \( \lambda \) may be complex. On occasions, these simplifications may break down and render linear stability insufficient. Both numerator and denominator of the stability function are polynomials of degree, \( \deg \leq s \). An explicit first stage reduces the degree of the denominator by one. A method is called A-stable and its stability function is called \( L \)-acceptable. \( L \)-acceptable stability functions have

\[
|\Re(z)| \leq R(z) \leq 0 \text{ on the entire negative real axis, } -\infty \leq z \leq 0, \text{ and, consequently, the order of accuracy may not exceed one } [408]. \text{ If, in addition to A-stability, } R(z)_{z \to -\infty} = 0, \text{ then the method is called L-stable, and its stability function is called L-acceptable. L-acceptable stability functions have } \deg Q(z) > \deg P(z). \text{ Intermediate between A- and L-stability is what Crouzeix [120,121,123] terms strong A-stability. This requires that a method be A-stable and}
$R(z)_{\to -\infty} < 1$ which gives the method a desirable error growth function [170,171]. Less restrictive measures of linear stability may also be considered [193,286,457]. $A(\alpha)$-stable methods provide $|R(z)| \leq 1$ for all $z$ in a sector of $\pm \alpha$ radians above and below the negative real axis, $-\infty \leq z \leq 0$, within the complex left half-plane (LHP). If $\alpha = \pi/2$, then the method is simply $A$-stable. If that angle includes $\alpha = 0$, the negative real axis, then the method is called $A_0$-stable. Methods that are both $A(\alpha)$-stable and have $R(z)_{\to -\infty} = 0$ are termed $L(\alpha)$-stable. A simple way to assist in achieving $L$-stability is to set $p_0 = 0$ in (31) by using the stiffly-accurate assumption, $a_{sj} = b_j$ [340], which reduces the deg $P(z)$ from $s$ to $s - 1$. The stiffly-accurate assumption forces $U[n+1]$ to be computed implicitly, while also increasing the convergence rate (see Table 3) on stiff and DAE problems. Lastly, a stability function is said to be $P$-stable if $|R(i\nu)| = 1$ where $i = \sqrt{-1}$ and $\nu$ is a real number [152,212].

Given the stability function, (30), then

$$R_{m,n}(z) = \frac{P(z)}{Q(z)} = \frac{\sum_{i=0}^{m} p_i z^i}{\sum_{j=0}^{n} q_j z^j} = \frac{\sum_{i=0}^{s} p_i z^i}{\prod_{j=1}^{s} (1-a_{jj}z)}$$  \hspace{1cm} (31)

where $p_i$ and $q_i$ are real. Nørsett and Wolfbrandt [318,452] show that the maximum order of approximation to exp($z$) using $R_{m,n}(z)$ is $m + 1$ if all poles are real. Applied to SDIRKs and ESDIRKs, this implies $p \leq s_I + 1$. To better understand the maximum order of approximation for $A$-acceptable approximations to exp($z$), Wanner et al. [70,227,316,317,443,444] introduced order stars where the order star of an approximation to the exponential is given by $R(z)/\exp(z)$. Shirkov [386] attempts to design $L$-stable DIRK methods having $p = s, s = 1, 2, 3, 4$ with “second-order damped” (SOD), i.e., deg $Q(z) = \text{deg } P(z) + 2$ or $p = s_I + 2$ rather than the more common deg $Q(z) = \text{deg } P(z) + 1$. From Nørsett [308,309] and Crouzeix [120], SDIRK methods of order $p = \{2,3,4\}$ and $s = \{1,2,3\}$ have been derived. The two higher-order, A-stable methods are shown to be unique but a four-stage, fifth-order method is impossible. Nørsett and Wolfbrandt state that a DIRK with $p = 5$ and $s = 4$ must have two distinct values of $a_{ii}$. Cooper and Sayfy [116] and Shintani and Yoshida [385] derive A-stable SDIRK methods having $p = 5$ and $s = 5$. Relaxing the requirement of identical diagonal coefficients, Cooper and Sayfy [116] derive an A-stable DIRK with $p = 6$ and $s = 6$, but the last stage is computed explicitly as $a_{66} = 0$. It is unlikely that an A-stable $p = 6, s = 6$ SDIRK exists. Methods having $p = 5$ and $s = 4$ [18], $p = 5$ and $s = 5$ [266], $p = 6$ and $s = 5$ [18] have also been derived, yet none are A-stable. Al-Rabeh [18] shows that the maximum order of an $s$-stage, nonconfluent DIRK with nonzero weights cannot exceed $s + 1$ and does this only for $1 \leq s \leq 5$. Bales et al. [30] prove that for $Q(z) = \prod_{i=1}^{s_i} (1-a_{ii}z)$, if $a_{ii} \geq 1/2, i = 1,2,\ldots,s_I$ and $R(z)$ is at least an order $s_I$ approximation to the exponential, then $R(z)$ is $A_0$-acceptable. Associated with each stability function is an error constant, $C$, defined by [262,278,395] $\exp(z) - R(z) = Cz^{p+1} + O(z^{p+2})$, where $R(z)$ is a $p$th-order approximant to the exponential. Burrage [50] and Wanner [442] note that when simplifying assumptions are applied to the method, the error constant bears a close resemblance to the error constant of the entire Runge-Kutta method. In low stage-order methods, this correlation is generally lost. As
the error is indicative of the error near \( z = 0 \), a minimum error constant may not be minimal for \( |z| \gg 0 \) [395]. Orel [326] proves that the minimum error constant of any approximant to the exponential having only real poles has all poles equal. Iserles and Nørsett [227] suggest that it may be profitable to consider DIRK methods rather than SDIRK methods. They ask whether more methods may be generated that are A-stable and whether certain order barriers might be broken by allowing the \( a_{ii} \) to vary. The work of Orel [326] and Keeling [253] suggests that there is no benefit to permitting distinct, nonzero, diagonal entries in \( Q(z) \). Shirobokov seeks to minimize the value of the error constant of SDIRK [387] and DIRK [388–391] methods along a finite interval of the negative real-axis beginning at the origin. In a later paper, Shirobokov [392–394] seeks to minimize the magnitude of \( |R(z) - \exp(z)| \) within the class of fourth-order accurate \( A_0 \)-stable DIRK methods. Kalitkin and Poshivaylo [250] seek to optimize implicit Runge-Kutta methods by considering the inverse stability function of an explicit Runge-Kutta method.

Limiting the discussion to SDIRKs and ESDIRKs, one may derive some useful analytical results regarding \( R(z) \) for \( A \)-stable methods in cases where the order of the method \( p \) and the number of implicit stages are related by \( s_I \leq p \leq s_I + 1 \). For L-stable methods, this becomes \( s_I - 1 \leq p \leq s_I \). Nørsett [309, 311] and others [50, 123, 193, 442] derive a compact expression for the SDIRK and ESDIRK stability function in terms of Laguerre polynomials. Using the definitions of the Laguerre polynomials and their \( m \)th-derivatives,

\[
L_s(x) = \sum_{k=0}^{s} \binom{s}{k} \frac{(-x)^k}{k!}, \quad L_s^{(m)}(x) = \sum_{k=0}^{s-m} \binom{s}{k+m} \frac{(-1)^{k+m}x^k}{k!},
\]

where

\[
\binom{s}{k} = \frac{s!}{(s-k)!k!},
\]

one may write

\[
P(z) = \sum_{j=0}^{s_I} p_j z^j = (-1)^{s_I} \sum_{j=0}^{s_I} L_{s_I}^{(s_I-j)} \left( \frac{1}{\gamma} \right) (\gamma z)^j
\]

\[
= \sum_{j=0}^{s_I} z^j \sum_{k=0}^{j} \binom{s_I}{k} (\gamma)^k \frac{k!}{(j-k)!},
\]

\[
Q(z) = (1 - \gamma z)^{s_I} = \sum_{k=0}^{s_I} q_k z^k = \sum_{k=0}^{s_I} \binom{s_I}{k} (-\gamma)^k z^k
\]

with error constant

\[
C = \frac{\gamma^{s_I} (-1)^{s_I+1} L_{s_I+1}^{(1)} \left( \frac{1}{\gamma} \right)}{(s_I+1)} = \frac{1}{s_I+1} \sum_{k=0}^{s_I} \binom{s_I+1}{k+1} \frac{(-\gamma)^{s_I-k}}{k!}.
\]

These results merit a quick explanation. The expression for \( Q(z) \) follows directly from its definition, \( Q(z) = \text{Det} [I - zA] \), and the diagonal structure of \( A \). From
[227], \( P(z) \) is defined by

\[
P(z) = \sum_{j=0}^{m} p_j z^j = (1 - \gamma z)^n \exp(z) + O\left(z^{m+1}\right)
\]  

(37)

since \( R(z) \) is a restricted approximant to the exponential. Matching terms after expanding \( (1 - \gamma z)^n \exp(z) \) in a series provides the form of \( P(z) \). This result may then be recast in terms of Laguerre polynomials.

Forcing the error constant to vanish gives the \( \gamma \) for which the stability function is an \((s_I + 1)\)th-order approximation to \( \exp(z) \) [309]. If one restricts this to cases where \( \deg P(z) + 1 = \deg Q(z) \) and \( s_I - 1 \leq p \leq s_I \), then the method will have \( R(-\infty) = 0 \). In this case, \( p_{s_I} = 0 \) and

\[
P(z) = \sum_{j=0}^{s_I-1} p_j z^j = (-1)^{s_I} \sum_{j=0}^{s_I-1} L_{s_I}^{(s_I-j)} \left(\frac{1}{\gamma}\right) (\gamma z)^j
\]

\[
= \sum_{j=0}^{s_I-1} z^j \sum_{k=0}^{j} \binom{s_I}{k} \frac{(-\gamma)^k}{(j-k)!}
\]

(38)

with \( p_{s_I} = C = 0 \)

\[
C = p_{s_I} = (-\gamma)^{s_I} L_{s_I} \left(\frac{1}{\gamma}\right) = \sum_{k=0}^{s_I} \binom{s_I}{k} \frac{(-\gamma)^{s_I-k}}{k!} = \sum_{k=0}^{s_I} \binom{s_I}{k} \frac{(-\gamma)^k}{(s_I-k)!}
\]

(39)

Higher-order error terms are given by Koto [262]. Butcher [67] also considers the case \( p = s - 2 \). Concepción [110] gives the necessary and sufficient conditions for the A-acceptability of \( R_{k-1,k}(z) \) approximates to the exponential having a single real root, \( \gamma^{-1} \) \((a_{ii} = \gamma)\). This is relevant to L-stable SDIRKs and ESDIRKs. Given the form of the stability function, one may enforce A- or I-stability. A Runge-Kutta method is imaginary axis or I-stable if \(|R(z)| \leq 1\) for \( \Re(z) = 0 \). Alt [20] determines that if \( R(z) \) has no poles in the complex LHP and the stability function is I-acceptable (the Runge-Kutta method is I-stable) then the method is A-stable. An analytic stability function for DIRK-type methods simply requires that \( a_{ii} \geq 0 \). Nørsett [310] devises the E-polynomial to test for I-stability,

\[
E(y) = Q(+iy)Q(-iy) - P(+iy)P(-iy)
\]

\[
= E_0 + E_2 y^2 + \cdots + E_{2s} y^{2s} = \sum_{j=0}^{s} E_{2j} y^{2j}
\]

(40)

where \( i = \sqrt{-1} \). Contemporaries of Nørsett; Alt [19, 20, 22], Crouzeix [120, 121], Kurdi [278] and Miller [299] use the identical principle in a slightly different form: \( |R(iy)| \leq 1 \) for all real \( y \). In general for DIRK-type methods, \( Q(+iy)Q(-iy) = \prod_{j=1}^{s_I} [1 + (a_{jj} y)^2] \); however, for SDIRKs and ESDIRKs, \( Q(+iy)Q(-iy) = [1 + (\gamma y)^2]^{s_I} \). I-stability requires that \( E(y) \geq 0 \) for all values of real \( y \), which requires \( E(y) \) to have only imaginary roots. It is sufficient, but not necessary, to have \( E_{2j} \geq \)
0, \ j = 1, 2, \cdots, s. For order \ p \geq 2j, E_{2j} = 0. First-stage explicit methods reduce the
degree of \ Q(z) \ by one, while stiffly-accurate methods reduce the degree of \ P(z) \ by
one by forcing \ p_s = 0. Hence, \ E_{2s} = 0. Further, if L-stability is desired, then \ p_{s_{I-1}}
must also vanish. Related but slightly different, Cooper and Sayfy [116] analyse
A-stability for DIRK-type methods with one or more \ a_{ii} \ = 0. This may cause one
stage value to be updated explicitly and may result in severe internal instability
on stiff problems. Al-Rabeh [17] investigates the requirements for A-stability when
\ p = s \ and all diagonal coefficients are not equal. An alternative, yet less practical,
approach to establishing I- and A-stability is given by Scherer and Wendler [182,
191, 303, 358, 359]. Defining \ C = A - \frac{1}{2}e b^T \ and \ Q = \{e, Ae, \cdots, A^{s-1}e\},
then a necessary and sufficient condition for I-stability is that there exists a symmetric
matrix \ R \ such that \ Re = b \ and \ Q^T(\mathcal{R}C + C^T\mathcal{R})Q \ are non-negative definite. If, in
addition, \ Q^T\mathcal{R}Q \ is non-negative definite, then the method is also A-stable. These
conditions may be simplified a bit in cases where the stability function is minimal by
setting \ Q = I. Minimality is a stronger requirement on a method than irreducibility
and requires that \ deg \ P(z) \ and \ deg \ Q(z) \ are not both less than \ s \ [358]. A- and
L-stable, first-stage explicit DIRK-type methods are not minimal. Note that for
\ Q = I \ and \ \mathcal{R} = B = \text{diag}(b), \ then \ (\mathcal{R}C + C^T\mathcal{R}) \ equals \ the \ algebraic \ stability \ matrix
\mathcal{M} \ given by (47).

With the \ E- \ polynomial, one may establish a priori the bounds on \ \gamma \ that will
result in either A-stable or L-stable methods \ [50, 193, 311] \ provided that \ p \approx s.
These bounds are given in Tables 4 and 5. Entries in the tables arise via several
different avenues. All A-stable values of \ \gamma \ in Table 4 entries, except \ \gamma = (3 +
\sqrt{3})/6, \ correspond to roots of \ E_{2s} \ or \ E_{2j_{\text{min}}} \ where \ j_{\text{min}} \ is the minimum value that
corresponds to a nonzero value of \ E_{2j}. While these two criteria are not sufficient to
ensure that \ E(y) \ \geq 0, \ they correspond to the observed values of \ \gamma \ in Table 4. The
\ \gamma = (3 + \sqrt{3})/6 \ entry results from satisfying order condition \ \tau_{2}^{(3)} \ . Ranges of \ \gamma \ for
A-stability are those where both \ E_{2s} \ or \ E_{2j_{\text{min}}} \ remain non-negative. For \ s \ odd, if a
root of \ E_{2j_{\text{min}}} \ is in this range, then order \ p = s + 1 \ can be achieved. No A-stable
methods were found for \ 9 \geq s_{I} \geq 20.

Finding L-stable values of \ \gamma, \ shown in Table 5, involves the values for the roots of
\ p_{s_{I}}, \ E_{2s}, \ E_{2j_{\text{min}}} \ and the roots to the discriminant of \ E(y). Since \ E_{2s} = \gamma^{2s}, \ E_{2s} \ will
remain positive for all positive values of \ \gamma. Boundaries in the range of \ \gamma \ correspond
to a root of either \ E_{2j_{\text{min}}} \ or the discriminant of \ E(y). Some care should be taken at
the values of \ \gamma \ which separate L-stable from non-L-stable methods as some boundary
points have \ |R(iy)| = 1 \ at discrete points on the imaginary axis [170]. Aggregating
and sorting these roots according to their magnitude, if \ E(y) \ \geq 0 \ between two
adjacent roots, then the method is L-stable for values of \ \gamma \ between these adjacent
root values. Further, if a root of \ p_{s_{I}} \ resides within this range, then the order of the
method is \ p = s. An alternative procedure to determine I-stability for any DIRK-
type method is to simply compute all roots of \ E(y) \ for a given value of \ \gamma \ and ensure
that all roots are imaginary. No L-stable methods were found for \ 12 \geq s_{I} \geq 20.

Rather than seek L-stable stability functions given in (31) by setting \ m = s_{I} - 1
and \ n = s_{I}, \ one may seek stability functions where \ m = s_{I} - 2 \ and \ n = s_{I},
zR(z)_{z \rightarrow -\infty} = 0, \ in \ hopes \ of \ having \ stronger \ damping \ of \ stiff \ scaled \ eigenvalues.

21
Table 4. Bounds on $\gamma$ for A-stable SDIRKs and ESDIRKs from orders two to twelve where $p$ is the order of accuracy and $s_I$ is the number of implicit stages.

<table>
<thead>
<tr>
<th>$s_I, p$</th>
<th>A-stable</th>
</tr>
</thead>
<tbody>
<tr>
<td>2, 2</td>
<td>$1/4 \leq \gamma \leq \infty$</td>
</tr>
<tr>
<td></td>
<td>$\gamma = (3 + \sqrt{3})/6$</td>
</tr>
<tr>
<td>2, 3</td>
<td>$1/3 \leq \gamma \leq 1.068579021301628806418834$</td>
</tr>
<tr>
<td></td>
<td>$\gamma = 1.068579021301628806418834$</td>
</tr>
<tr>
<td>3, 3</td>
<td>$0.3943375672974064411272872 \leq \gamma \leq 1.2805797612753054573024841$</td>
</tr>
<tr>
<td>3, 4</td>
<td>$0.2465051931428202746001423 \leq \gamma \leq 0.36180339887498948204587$</td>
</tr>
<tr>
<td></td>
<td>$0.4207825127659933063870173 \leq \gamma \leq 0.4732683912582953244555885$</td>
</tr>
<tr>
<td>5, 5</td>
<td>$0.2840646380117982930387010 \leq \gamma \leq 0.5409068780733081049137798$</td>
</tr>
<tr>
<td>5, 6</td>
<td>$\gamma = 0.4732683912582953244555885$</td>
</tr>
<tr>
<td>6, 6</td>
<td>$\gamma = 0.4732683912582953244555885$</td>
</tr>
<tr>
<td>7, 7</td>
<td>-</td>
</tr>
<tr>
<td>7, 8</td>
<td>-</td>
</tr>
<tr>
<td>8, 8</td>
<td>$0.2170497430943030918315779 \leq \gamma \leq 0.2647142465800596850440755$</td>
</tr>
<tr>
<td>8, 9</td>
<td>-</td>
</tr>
</tbody>
</table>

Since the maximum order is $p = m + 1$, methods are sought with $p = s_I - 1$. Shirkov [386] calls this second-order damped (SOD) L-stability. Lobatto IIIC methods have this same property. Values for $\gamma$ to construct such methods are derived based on the roots of the polynomial

$$p_{s_I-1} = \sum_{k=0}^{s_I-1} \binom{s_I}{k} \frac{(-\gamma)^k}{(s_I - k - 1)!} = 0 \quad (41)$$

and given in Table 6. No second-order damped L-stable methods with $p = s_I - 1$ were found for $12 \geq s_I \geq 20$ however, methods do exist at $s_I = 12$ for $p = s_I - 2$.

An important question to ask is: which linear stability criterion is the best. As different users will solve different problems and these problems may give rise to both real and imaginary eigenvalues, A-stability would appear to be a minimal request. Prothero and Robinson [340] often found this to be inadequate. They preferred stiffly-accurate, L-stable methods. Because it is not an onerous task to incorporate A-stability or even L-stability into the Runge-Kutta coefficients, it is difficult to justify using less stable methods. Alexander [5, 6, 97] advocates the use of stiffly-accurate, L-stable methods, as does Kværnø [281, 282], and finds them of generally greater practical value than nonlinearly (algebraically) stable methods (See §2.5). This favorable sentiment is reinforced by the contents of Table 3. Lambert [285, 286] criticizes L-stability as being too strong, particularly in cases where eigenvalues slightly enter the complex right half-plane (RHP) and advocates instead, strictly A-stable methods. Strictly A-stable methods, such as Gauss, Lobatto IIIA and
Table 5. Bounds on $\gamma$ for L-stable SDIRKs and ESDIRKs from orders two to twelve where $p$ is the order of accuracy and $s_I$ is the number of implicit stages.

<table>
<thead>
<tr>
<th>$s_I, p$</th>
<th>L-stable</th>
</tr>
</thead>
<tbody>
<tr>
<td>2, 2</td>
<td>$\gamma = (2 \pm \sqrt{2})/2$</td>
</tr>
<tr>
<td>3, 2</td>
<td>$0.1804253064293985641345831 \leq \gamma \leq 2.1856000973550400826291400$</td>
</tr>
<tr>
<td>3, 3</td>
<td>$\gamma = 0.43586652150845899941601945$</td>
</tr>
<tr>
<td>4, 3</td>
<td>$0.2236478009341764510696898 \leq \gamma \leq 0.5728160624821348554080014$</td>
</tr>
<tr>
<td>4, 4</td>
<td>$\gamma = 0.5728160624821348554080014$</td>
</tr>
<tr>
<td>5, 4</td>
<td>$0.2479946362127474551679910 \leq \gamma \leq 0.6760423932262813288723863$</td>
</tr>
<tr>
<td>5, 5</td>
<td>$\gamma = 0.2780538411364523249315862$</td>
</tr>
<tr>
<td>6, 5</td>
<td>$0.183914653675175163231436 \leq \gamma \leq 0.3341423670680504359540301$</td>
</tr>
<tr>
<td>6, 6</td>
<td>$\gamma = 0.3341423670680504359540301$</td>
</tr>
<tr>
<td>7, 6</td>
<td>$0.2040834517158857633717906 \leq \gamma \leq 0.378864894485328340258853$</td>
</tr>
<tr>
<td>7, 7</td>
<td>-</td>
</tr>
<tr>
<td>8, 7</td>
<td>$0.1566585993970439483924506 \leq \gamma \leq 0.2029348608433776737779349$</td>
</tr>
<tr>
<td>8, 8</td>
<td>$0.051941719494007117460614 \leq \gamma \leq 0.234373159605835579475589$</td>
</tr>
<tr>
<td>9, 8</td>
<td>$\gamma = 0.234373159605835579475589$</td>
</tr>
<tr>
<td>9, 9</td>
<td>-</td>
</tr>
<tr>
<td>10, 9</td>
<td>-</td>
</tr>
<tr>
<td>10, 10</td>
<td>-</td>
</tr>
<tr>
<td>11, 10</td>
<td>$0.1468989308591125260680428 \leq \gamma \leq 0.1657926100980560571096175$</td>
</tr>
<tr>
<td>11, 11</td>
<td>$0.1937733662800920635754554 \leq \gamma \leq 0.1961524231108803003116274$</td>
</tr>
</tbody>
</table>

23
Table 6. Values for $\gamma$ in second-order damped L-stable SDIRKs and ESDIRKs.

<table>
<thead>
<tr>
<th>$s_1, p$</th>
<th>Second-order damped L-stable</th>
</tr>
</thead>
<tbody>
<tr>
<td>3, 1</td>
<td>$0.2113248654051871177454256 \leq \gamma \leq 0.7886751345948128822545744$</td>
</tr>
<tr>
<td>3, 2</td>
<td>$(3 \pm \sqrt{3})/6$</td>
</tr>
<tr>
<td>4, 2</td>
<td>$0.1432025528455778629375372 \leq \gamma \leq 0.983396079525512501554340$</td>
</tr>
<tr>
<td>4, 3</td>
<td>$0.3025345781826507712164413$</td>
</tr>
<tr>
<td>5, 3</td>
<td>$0.1749814971185374260730290 \leq \gamma \leq 0.3888576711028921132337183$</td>
</tr>
<tr>
<td>5, 4</td>
<td>$0.3888576711028921132337183$</td>
</tr>
<tr>
<td>6, 4</td>
<td>$0.1903760890384496829539382 \leq \gamma \leq 0.4535842098396461592528162$</td>
</tr>
<tr>
<td>6, 5</td>
<td>$0.2168805435476052775934965$</td>
</tr>
<tr>
<td>7, 5</td>
<td>$0.1495110383364076987694180 \leq \gamma \leq 0.2579552415971074838409368$</td>
</tr>
<tr>
<td>7, 6</td>
<td>$0.2579552415971074838409368$</td>
</tr>
<tr>
<td>8, 6</td>
<td>$0.1635807847918329355329535 \leq \gamma \leq 0.2905565031843389842415167$</td>
</tr>
<tr>
<td>8, 7</td>
<td>$0.1690246378620602674461848$</td>
</tr>
<tr>
<td>9, 7</td>
<td>$0.1312239057862475302028635 \leq \gamma \leq 0.1929678039823110580718429$</td>
</tr>
<tr>
<td>9, 8</td>
<td>$0.1929678039823110580718429$</td>
</tr>
<tr>
<td>10, 8</td>
<td>$0.1421375515417974712667375 \leq \gamma \leq 0.2126998008020526288594700$</td>
</tr>
<tr>
<td>10, 9</td>
<td>$\cdots$</td>
</tr>
<tr>
<td>11, 9</td>
<td>$0.1507669010690233069364339 \leq \gamma \leq 0.1541460739322757936552230$</td>
</tr>
<tr>
<td>11, 10</td>
<td>$0.1541460739322757936552230$</td>
</tr>
<tr>
<td>12, 10</td>
<td>$0.1251361537650653200858842 \leq \gamma \leq 0.1673932884265689320384167$</td>
</tr>
<tr>
<td>12, 11</td>
<td>$\cdots$</td>
</tr>
</tbody>
</table>
Lobatto IIIB methods, are those where \( E(y) = 0 \) and the stability boundary is exactly the imaginary axis. Both symmetric and symplectic DIRK-type methods are also strictly \( A \)-stable.

Additional motivations may also be considered in the construction of desirable stability functions. In solving discretized PDEs using an approximately factorized Newton iteration method, maximum step-sizes may not be governed by traditional issues like \( A \)- or \( L \)-stability but rather by the spectral radius of the \( A \)-matrix, \( \rho(A) \). Van der Houwen and Sommeijer [215] construct \( A \)- and \( L \)-stable SDIRK methods of orders two and three in two, three, or four stages, which minimize \( \rho(A) \). At second-order, optimal methods have \( |P(iy)| = |Q(iy)| \). Minimal \( \rho(A) \) is generally achieved for all \( a_{ii} = \gamma \). Another possible motivation related to space-time PDEs is to focus on the locations of the zeros of \( P(z) \) [424]. The goal might be to determine the location of scaled eigenvalues \( z_i = \lambda_i(\Delta t) \), which correspond to poorly resolved spatial information, and to set \( P(z_i) = 0 \). This approach has been done for explicit Runge-Kutta (ERK) methods and requires \( s \gg p \) to be able to exert any meaningful control.

Three further stability concepts to be mentioned are \( S \)-, \( D \)- and AS-stability. \( S \)- and strong \( S \)-stability were introduced by Prothero and Robinson [340] to extend the concepts of \( A \)- and \( L \)-stability to the nonhomogeneous test equation \( y'(t) = g'(t) + \lambda(y(t) - g(t)) \). One may also speak of \( (\alpha) \)-stable methods. Alexander [5] and others [92, 209, 280] derive several strongly \( S \)-stable SDIRKs; however, Zhao et al. [465] claim that the concept of \( S \)-stability is fundamentally flawed and that no consistent, well-defined Runge-Kutta method is \( S \)-stable. Alexander states that an \( A \)-stable DIRK-type method with positive \( a_{ii} \) and invertible \( A \)-matrix is \( S \)-stable if and only if \( |R(-\infty)| < 1 \). An \( S \)-stable method is strongly \( S \)-stable if and only if it is stiffly-accurate. Decomposition- or \( D \)-stability [129] is not a traditional stability measure in that it is not concerned with the propagation of errors. It is concerned with the boundedness of stiff, linear, nonautonomous problems. Hairer [183] states that \( D \)-stability requires that the eigenvalues of \( a_{ij} \) lie in the complex RHP \( (\Re(\lambda_{a_{ij}}) > 0) \). If \( a_{ij} \) is singular (EDIRK, ESDIRK, or QESDIRK methods), then only reducible or confluent methods can be \( D \)-stable. If a Runge-Kutta method is irreducible and \( \deg Q(z) = s \), then \( A \)-stability implies \( D \)-stability. AS-stability [57, 77, 123] is the linear analog of the more familiar nonlinear \( BS \)-stability concept and plays an essential role in the convergence analysis of stiff nonlinear problems with constant coefficients. A method is called AS-stable if \( (1 - zA)^{-1} \) is nonsingular for all \( z \) in the complex LHP and \( \|z b^T(1 - zA)^{-1}\| \) is uniformly bounded in an inner product norm within the complex LHP. Stability analysis of nonautonomous linear problems focuses on \( AN \)-stability using the K-function

\[
K(Z) = 1 + b^T Z (I - AZ)^{-1} e = \frac{\det [I - (A - e \otimes b^T)Z]}{\det [I - AZ]},
\]

where \( Z = \text{diag}\{z_1, z_2, \ldots, z_s\} \) and which, surprisingly, is more properly investigated within the purview of nonlinear stability. Following Burrage and Butcher [53], the K-function for Runge-Kutta methods is given as

\[
K(Z) = 1 + e^T BZ [I - AZ]^{-1} e,
\]
where $B = \text{diag}(b^T)$. A Runge-Kutta is AN-stable if $|K(Z)| \leq 1$, $\Re(z_i) \leq 0$ for all $z_i, i = 1, 2, \ldots, s$ such that $z_i = z_j$ whenever $c_i = c_j$. Defining $K_{\text{int}} = (I - AZ)^{-1} e$ and $e = (I - AZ) K_{\text{int}}$, then

$$K(Z) = 1 + e^T B Z K_{\text{int}}, \quad \tilde{K}(Z) = 1 + \hat{K}_{\text{int}}^T \hat{Z} B^T e$$

with the superscript $\hat{\cdot}$ denoting the complex conjugate. Therefore, since $(B^T e)(e^T B) = BB^T$,

$$|K(Z)| = \tilde{K}(Z) K(Z) = 1 + e^T B Z K_{\text{int}} + \hat{K}_{\text{int}}^T \hat{Z} B^T e + \hat{K}_{\text{int}}^T \hat{Z} B B^T K_{\text{int}}$$

$$= 1 + \hat{K}_{\text{int}}^T \left(B Z + \hat{Z} B^T\right) K_{\text{int}}$$

$$- \hat{K}_{\text{int}}^T \hat{Z}^k (B^T A + A^T B - BB^T) Z K_{\text{int}},$$

where expressions for $e$ and its transpose were used. Defining the symmetric algebraic stability matrix as

$$M = B^T A + A^T B - BB^T, \quad M_{ij} = b_i a_{ij} + b_j a_{ji} - b_i b_j, \quad i, j = 1, 2, \ldots, s,$$

then

$$|K(Z)| = 1 - 2 \Re\left(\hat{K}_{\text{int}}^T B Z K_{\text{int}}\right) - \hat{K}_{\text{int}}^T \hat{Z} M Z K_{\text{int}}$$

$$= 2 \left(\sum_{i=1}^{s} b_i \Re(z_i) |K_{\text{int}}^{(i)}|^2\right) - \sum_{i,j=1}^{s} \hat{K}_{\text{int}}^{(i)} z_i M_{ij} z_j K_{\text{int}}^{(j)},$$

where $K_{\text{int}}^{(i)}$ are the components of the vector $K_{\text{int}}$, which will be seen below to be the nonautonomous internal stability function. If the Runge-Kutta method is AN-stable then, $|K(Z)| \leq 1$ or

$$|K(Z)| - 1 = 2 \left(\sum_{i=1}^{s} b_i \Re(z_i) |K_{\text{int}}^{(i)}|^2\right) - \sum_{i,j=1}^{s} \hat{K}_{\text{int}}^{(i)} z_i M_{ij} z_j K_{\text{int}}^{(j)} \leq 0$$

for all $z_i$ such that $z_i = z_j$ whenever $c_i = c_j$ where $\Re(z_i) \leq 0$. It will be seen in the next section that this result has much in common with algebraic stability except when the method is confluent.

### 2.5 Nonlinear Stability

Maximum norm and inner product norm contractivity on nonlinear problems as well as nonlinear stability [51–53, 89, 122, 125, 126, 129, 192, 193, 251, 252, 266, 267, 287, 300, 312, 313, 362, 456, 469] may be considered for DIRK-type methods. Stability and contractivity are related, but contractivity is generally a stronger requirement [125, 126, 268, 409, 412]. For ODE systems where the difference between two solutions satisfies $||\tilde{U}(t + \Delta t) - U(t + \Delta t)|| \leq ||\tilde{U}(t) - U(t)||$ in some norm (a dissipativity condition) for $\Delta t > 0$, one might reasonably demand of the numerical method that $||\tilde{U}^{(n)} - U^{(n)}|| \leq \alpha n^\beta ||\tilde{U}^{(0)} - U^{(0)}||$ in that same norm where $\alpha$ and $\beta$ denote positive real constants and $n \geq 1$ denotes the number of time steps [412]. In the
case where \( \beta = 0 \) and \( \alpha = 1 \), one has contractivity. If \( \beta = 0 \) but \( \alpha \) is of moderate size, one has strong stability, but if both \( \alpha \) and \( \beta \) are of moderate size then one has only weak stability. Nonlinear stability of implicit Runge-Kutta methods is generally characterized in terms of algebraic-, B-, BN-, or AN-stability and requires a dissipativity condition on the underlying ODEs. Each of these measures implies A-stability. The first three also imply unconditional contractivity (unlimited step size) in an inner product norm, \( r_{\mathcal{F}_2} = \infty \) where \( \mathcal{F} \) denotes a nonlinear RHS, for both confluent and nonconfluent Runge-Kutta methods [270]. For nonconfluent methods, each of these four concepts is equivalent. For confluent methods, B- and BN-stability are equivalent to algebraic-stability but AN-stability is not [129, 193, 300]. It is sufficient for the present purposes to focus simply on algebraic-stability. In the inner product norm case, one constructs the matrices \( \mathcal{M} \) and \( \mathbf{B} = \text{diag}(\mathbf{b}) \) or \( \mathbf{b} = \mathbf{B} \mathbf{e} \) using the algebraic stability matrix as \( \mathcal{M} = \mathbf{B}^{-1/2} \mathbf{M} \mathbf{B}^{-1/2} \) where the former expression is used only when \( \mathbf{b} > 0 \). An analogous construction may be made for the embedded method by replacing \( \mathbf{b} \) with \( \hat{\mathbf{b}} \).

\[
\begin{bmatrix}
\gamma \\
1 - \gamma
\end{bmatrix} = \begin{bmatrix}
1 - 2\gamma & \gamma \\
\gamma & 1 - \gamma
\end{bmatrix}
\begin{bmatrix}
\gamma & 0 & 0 \\
\gamma & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\gamma & 0 & 0 \\
2\gamma & \gamma & 1 - 4\gamma \\
1 - \gamma & 2\gamma & \gamma
\end{bmatrix}
\begin{bmatrix}
\frac{1}{6(1-2\gamma)^2} & \frac{1}{3(2\gamma - 1)^2} & \frac{1}{6(1-2\gamma)^2}
\end{bmatrix}
\]

where \( \gamma = \frac{2 + \sqrt{3}}{6} \approx 0.7887 \) and \( R(-\infty) = 1.0 \) for the two-stage method, and \( \gamma = \frac{3 + 2\sqrt{3} \cos \left( \frac{\pi}{6} \right)}{6} \approx 1.069 \) and \( R(-\infty) = -0.732 \) for the three-stage method. The two-stage method also appears in a paper by Makinson [296], but in the context of Rosenbrock methods. Burrage and Butcher [53] present an interesting SDIRK method

\[
\begin{align*}
\gamma & \\
\gamma + b_1 & \\
\gamma + b_1 + b_2 & \\
\vdots & \\
\gamma + \sum_{i=1}^{s-1} b_i & \\
\gamma + \sum_{i=1}^{s} b_i & \\
\end{align*}
\begin{bmatrix}
\gamma & 0 & 0 & \cdots & 0 \\
b_1 & \gamma & 0 & \cdots & 0 \\
b_1 & b_2 & \gamma & \cdots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \ddots \\
b_1 & b_2 & \cdots & b_{s-1} & \gamma \\
\end{bmatrix}
\]

where the resulting algebraic-stability matrix is diagonal with entries \( M_{ii} = b_i(2\gamma - b_i) \) and \( i = 1, 2, \ldots, s \). We remark that the form of this Butcher array is reminiscent of the van der Houwen approach to memory storage reduction [256]. Burrage [51] presents second-, third-, and fourth-order algebraically stable SDIRK methods up to four stages. At two stages, SDIRK-NCS23 given previously (with a different \( \gamma \)) is algebraically stable and second-order accurate for \( \gamma \geq 1/4, \gamma \neq 1/2 \) and is third-
order accurate for $\gamma = \frac{3+\sqrt{3}}{6}$. At three and four stages, Burrage gives complete class of nonconfluent algebraically stable SDIRKs of orders three and four are given. To design these algebraically stable methods, Burrage defines a symmetric matrix $R = V^T MV$ by using the algebraic-stability matrix and the van der Monde matrix, $V_{ij} = c_{j-1}^i, \quad i, j = 1, 2, \ldots, s$. It may be shown that all elements of $R_{ij}$ vanish for $i+j \leq p$, where $p$ is the classical order of the method. Assuming a nonconfluent method, the irreducible DIRK-type method is algebraically stable if $R \geq O$ and $b > 0$. Nørsett and Thomsen [312] derive an algebraically stable, three-stage SDIRK, with an A-stable embedded method of order two. Cameron [89] derives two algebraically and L-stable, four-stage, third-order DIRKs along with an algebraically stable, three-stage SDIRK method.

Cooper [115] offers a less restrictive form of nonlinear stability called weak B-stability. It requires the existence of a symmetric matrix $D$ such that $b \geq 0$ and $DA + A^T D - b^T b \geq O$, with $De = b$ and $d_{ij} \leq 0$ for $i \neq j$. He shows that the two-stage, second-order SDIRK method is both B-stable and weakly B-stable for $\gamma \geq 1/4$ but is weakly B-stable for $0 \leq b_2 \leq 1/2$ and B-stable for only $b_2 = 1/2$. Weak stability is closely related to what Spijker [411] calls weak contractivity. He simultaneously generalizes algebraic-stability and weak B-stability by considering a modified dissipativity condition to study contractivity and weak contractivity. Rather than weakening algebraic-stability, passive and lossless Runge-Kutta methods require more than algebraic-stability [158, 159, 319]. Passive methods require that all elements of $b$ be identical along with algebraic-stability. Lossless methods require the method to be both passive and symplectic.

An important question is whether algebraic-stability matters in a practical sense. Hairer [184] constructs a one-parameter family of A-stable, three-stage, fourth-order, fully implicit Runge-Kutta methods by using assumptions $C(1), D(1), c_i = \{0, 1/2, 1\}$ and $b_i \in \{1/6, 4/6, 1/6\}$. For a specific value of the parameter, the method becomes algebraically stable. The method was tested on Kreiss’s problem [129] which, in the limit of infinite stiffness, transforms from an ODE to an index-2 DAE. Behavior of the methods was best for choices of the parameter that nearly, but not exactly, corresponded to algebraic-stability. Scheme efficiency was relatively insensitive to the parameter at coarse tolerances but became strongly dependent on it at strict tolerances. Contractivity was a strong function of the parameter at all tolerances. For all values of the adjustable parameter, all internal stages were completely damped for $z \to -\infty$. It was not clear how changing the value of the adjustable parameter affected the satisfaction of index-2 DAE order conditions for the method. In another effort to assess the practical value of algebraically stable methods, Verwer [436] considers whether the stability property is preserved during implementation with a Newton-type solver. If users are willing to reevaluate the Jacobian at each stage, algebraic-stability is preserved; otherwise, it is generally not preserved. Spijker [412] asks whether methods exist that possess strong or weak stability but not contractivity (algebraic-stability) on (1), subject to a dissipativity condition. Kraaijevanger and Spijker [270] construct the DIRK
method,

\[
\begin{array}{ccc}
1/2 & 1/2 & 0 \\
3/2 & -1/2 & 2 \\
-1/2 & 3/2 & \\
\end{array}
\] (52)

with \( \alpha = 2 \) and \( \beta = 0 \) when the step-size remains constant. This implies that algebraic-stability may be an unnecessarily strong demand. For variable step-sizes, however, algebraic-stability is necessary in the method. Schmitt [360] and Hundsdorfer [218] (section 5.5.3) offer stability estimates for nonalgebraically stable methods. Schmitt concludes that nonlinear stability is necessary for the transient regions, but linear stability is sufficient for smoother regions of the solution. Hansen and Ostermann [196] consider algebraically stable DIRK methods in cases where the discretized space-time PDE solution lacks regularity and no restrictions are placed on the initial conditions. Convergence rates for second- and third-order SDIRKs are observed to approximately equal either the classical order of the method or the stage order plus one, depending on the stiffness.

Associated with algebraic-stability [62, 64, 114, 129, 193] are the concepts of B-convergence and BS-stability. The analysis of BS-stability extends that of B-stability by allowing nonuniform perturbations between each of the stages. In regard to DIRK-type methods, only a few remarks will be made. First, B-convergence on dissipative problems implies algebraic-stability [193]. Scholz [362] presents an A-stable, but not algebraically stable, third-order SDIRK, having stage-order one, which he claims is B-convergent of order four. Burrage and Hundsdorfer [56] show that the method SDIRK-NCS23 ( (50) with a different \( \gamma \) ) is B-convergent of order 2, or the stage-order plus one, for \( \gamma = 1/4 \) or \( \gamma = 1/2 \). For problems having a one-sided Lipschitz constant, \( \nu \geq 0 \), BS-stability is implied by algebraic-stability if the method is nonconfluent, hence, nonconfluent DIRK-type methods with \( a_{ii} > 0 \) may be BS-stable [219]. Calvo et al. [77] note that in constructing Runge-Kutta methods that satisfy all of the “natural requirements” for numerically solving stiff systems, one faces severe restrictions. This is reconciled against the fact that many methods that do not satisfy all of these restrictive natural requirements work well in practice. In this context, they then consider stiff semilinear problems and consider matters such as the consequences of relaxing BS- to AS-stability.

As algebraic-stability provides an unconditional contractivity radius on nonlinear problems, one may seek the less restrictive conditional contractivity [125, 126, 266, 300] where \( r_{F_2} < \infty \). To do this, one must relax the dissipativity condition associated with unconditionally contractive methods and replace it with a so-called circle condition. Denoting the minimum eigenvalue of \( M_{ii} \), (47), as \( \lambda_{\min}^M \) in cases where \( b_i > 0 \), the radius of conditional contractivity, determined in an inner product norm, of the DIRK-type method is given by \( r_{F_2} = -1/\lambda_{\min}^M \). The maximum order of a circle-contractive DIRK, where \( r_{F_2} > 0 \), is six but it is only four for an SDIRK [181].

The preceding discussion focused on properties considered in the context of inner product norms. Maximum norm contractivity applied to nonlinear problems is the most restrictive contractivity condition and limits DIRK-type methods to
first-order accuracy if unconditional contractivity, $r_{F_{\infty}} = \infty$, is desired [266]. As mentioned earlier, this order barrier does not change if the nonlinear problem is replaced with a linear one, where $r_{E_{\infty}} = \infty$. Relaxing unconditional to conditional contractivity so that the maximum-norm conditional contractivity radius, $r_{F_{\infty}} > 0$ rather than $r_{F_{\infty}} = \infty$, one requires that beyond $a_{ij}, b_i > 0$, that $(I - \xi A)$ is non-singular and $R(\xi) \geq 0$, $A(\xi) = A(I - \xi A)^{-1} \geq 0$, $B(\xi) = b^T(I - \xi A)^{-1} \geq 0$ and $E(\xi) = (I - \xi A)^{-1}e \geq 0$, where $r_{F_{\infty}}$ is the smallest value of $\xi$ for which all components of $R(\xi)$, $A(\xi)$, $B(\xi)$ and $E(\xi)$ are non-negative. Kraaijevanger [266] constructs a fifth-order EDIRK that is neither algebraically stable nor A-stable yet has nonvanishing contractivity radii of $r_{F_{\infty}} = 0.5906$ and $r_{F_2} = 2.489$. Ferracina and Spijker [148, 149] give the maximally contractive SDIRK in two stages, (78), with $r_{F_{\infty}} = 2$. They also show that any method which is optimal in the sense that $r_{F_{\infty}}$ is maximized also is optimal in the sense of maximizing the step-size for a total variation diminishing (TVD) Runge-Kutta process. In a later paper, Ferracina and Spijker [150] establish $r_{F_{\infty}}$ for SDIRKs at orders one through four and stage numbers from one to eight. Duraisamy et al. [134] also consider TVD SDIRKs. Ketcheson et al. [257] include fully implicit Runge-Kutta methods as well as DIRK-methods to order six while Gottlieb et al. [177, 178] also include multistep methods. Extension of BS-stability bounds to the max norm is given by Kraaijevanger [266]. Positivity of $a_{ij}$ and $b_i$ in a DIRK are also useful preserving the monotonicity behavior of linear differential equations describing dynamical systems during numerical integration [259]. Hundsdorfer and Spijker [221] relate the results of Kraaijevanger [266] to the topic of boundedness including the total variation bounded (TVB) property.

**2.6 Internal Stability**

Beyond traditional stepwise stability, it may be useful to control the stability associated with each stage in addition to each step. This is particularly true for large scaled eigenvalues, $z$, associated with stiff problems [255, 436]. IMEX additive Runge-Kutta methods are especially sensitive to the internal stability of the implicit method [255]. Kurdi [278] appears to have been the first to consider the internal stability of DIRK-type methods. To determine the vector of internal stabilities of Runge-Kutta methods, one evaluates

$$R_{\text{int}}(z) = (I - zA)^{-1}e = \{R_{\text{int}}^{(1)}(z), R_{\text{int}}^{(2)}(z), \ldots, R_{\text{int}}^{(s)}(z)\}^T = \left\{\begin{array}{c}
P_{\text{int}}^{(1)}(z) \\
P_{\text{int}}^{(2)}(z) \\
\vdots \\
P_{\text{int}}^{(s)}(z) \\
Q_{\text{int}}^{(1)}(z) \\
Q_{\text{int}}^{(2)}(z) \\
\vdots \\
Q_{\text{int}}^{(s)}(z) \end{array}\right\}^T.$$

(53)

The primary concern will be the value of $R_{\text{int}}^{(i)}(-\infty)$. For DIRK-type methods, $P_{\text{int}}^{(i)}(z)$ is generally a polynomial of degree $i - 1$ in $z$ because a stage-wise version of the stiffly-accurate assumption holds while $Q_{\text{int}}^{(i)}(z)$ is generally of degree $i$. Consequently, DIRKs with $a_{ii} \neq 0$ and SDIRKs have $R_{\text{int}}^{(i)}(-\infty) = 0$. EDIRKs, ESDIRKs
and QESDIRKs with $Q^{(i)}_{\text{int}}(z)$ reduced to degree $i - 1$ because $a_{11} = 0$, then they do not generally satisfy $R^{(i)}_{\text{int}}(-\infty) = 0$. If EDIRK, ESDIRK and QESDIRK methods satisfy $C(1)$, then $R^{(1)}_{\text{int}}(-\infty) = 1$. If the EDIRK, ESDIRK, or QESDIRK satisfies $C(2)$, then $R^{(2)}_{\text{int}}(-\infty) = -1$. One may also consider the E-polynomial, (40), at internal stages to determine stage-wise I-stability by using

$$E^{(i)}_{\text{int}}(y) = Q^{(i)}_{\text{int}}(iy)Q^{(i)}_{\text{int}}(-iy) - P^{(i)}_{\text{int}}(iy)P^{(i)}_{\text{int}}(-iy).$$

(54)

To assess the internal AN-stability with regard to nonautonomous linear equations, one may consider evaluating the Runge-Kutta K-function [266] at internal stages

$$K_{\text{int}}(Z) = (I - AZ)^{-1}e = \{K^{(1)}_{\text{int}}(Z), K^{(2)}_{\text{int}}(Z), \cdots, K^{(s)}_{\text{int}}(Z)\}^T.$$  

(55)

One could also consider a stage-wise analog to the algebraic-stability matrix. Following section 4.2 of Dekker and Verwer [129], one may derive the following step relations

$$\|U^{[n+1]}\|^2 - \|U^{[n]}\|^2 = 2(\Delta t)\sum_{j=1}^{s} b_j < U_j, F_j > - (\Delta t)^2 \sum_{j,k=1}^{s} (b_ja_{jk} + b_ka_{kj} - b_jb_k) < F_j, F_k >$$

(56)

and stage

$$\|U_i\|^2 - \|U^{[n]}\|^2 = 2(\Delta t)\sum_{j=1}^{s} a_{ij} < U_j, F_j > - (\Delta t)^2 \sum_{j,k=1}^{s} (a_{ij}a_{jk} + a_{ik}a_{kj} - a_{ij}a_{ik}) < F_j, F_k >$$

(57)

relations where

$$M^{(i)}_{jk} = a_{ij}a_{jk} + a_{ik}a_{kj} - a_{ij}a_{ik}$$

(58)

is the internal algebraic-stability matrix for stage $i$. Hence, for internal algebraic stability on stage $i$ of an irreducible method, it is necessary that $M^{(i)} \geq O$ and that $a_{ij} \geq 0$. This matrix has been considered by Burrage and Butcher [53] but it does not appear to have ever been considered in this context.

Internal stability influences on methods have been studied in several different contexts. Calvo et al. [76] consider, in part, the effect of internal stability on the convergence of IRK methods applied to stiff, nonlinear, dissipative ODEs. González-Pinto et al. [173–175] factor internal stability into the design of starting algorithms for implicit Runge-Kutta methods. It is for the preservation of internal stability that it is unwise to use $a_{ii} = 0$ on any stage but the first.

Several additional internal stability concepts include ASI-stability [57,77,123] for linear problems and BSI-stability [64,129,193] for nonlinear problems. A method is
ASI-stable if the matrix \((I - zA)\) is nonsingular for all \(z\) in the complex LHP and its inverse is uniformly bounded in the complex LHP in an inner product norm. If there exists a positive definite, diagonal matrix \(D\), such that \(DA + A^T D\) is positive definite, then the Runge-Kutta method is BSI-stable \([128]\). This requirement is only achievable for DIRK-type methods if \(a_{ii} > 0\) \([219]\). Kraaijevanger \([266]\) gives BSI-stability bounds valid in all norms, not just an inner product norm. Internal S-stability has been considered by Verwer \([435]\) and internal D-stability is discussed by Dekker and Verwer \([129]\).

2.7 Dense Output

On occasion, it may be necessary to determine the value of the integration vector at time locations not coincident with a time step \((t[n])\) or stage location \((t[n] + c_i \Delta t)\). Dense output \([190, 332]\) provides a mechanism that allows high-order interpolation of the integration variables at any point, \(t[n] + \theta \Delta t\), inside the current integration step where \(0 \leq \theta \leq 1\) by using

\[
U(t[n] + \theta \Delta t) \cong U[n] + (\Delta t) \sum_{i=1}^{s} b^*_i(\theta) F_i, \quad b^*_i(\theta) = \sum_{j=1}^{p^*} b^*_i F_j, \quad b^*_i(\theta = 1) = b_i, \quad (59)
\]

where \(p^*\) is the lowest order of the interpolant on the interval \(0 \leq \theta \leq 1\). By construction, \(b^*_i(\theta = 0) = 0\). Order conditions at order \(m\) and the stability function \(R^*(z, \theta)\) for the dense-output method are given by \([35]\)

\[
\tau_{ij}^{(m)} = \frac{1}{\sigma} \sum_{i} b^*_i(\theta) \Phi_{i,j}^{(m)} - \frac{\alpha \theta^m}{m!}, \quad R^*(z, \theta) = \frac{\text{Det} \left[ I - zA + ze \otimes b^{*T}(\theta) \right]}{\text{Det} \left[ I - zA \right]}, \quad (60)
\]

Setting \(m = p\) and \(\theta = 1\) retrieves the standard Runge-Kutta order conditions. As with the main and embedded formulas, one may write terms like \(A^*(p^*+1) = A^*(p^*+1)(\theta)\) to access the truncation error of the dense output method. Relatively few DIRK-type methods appearing in the literature include dense output \([89, 193, 255, 280, 382–384, 426]\). As with embedded methods, dense output methods could be enhanced by using a two-step procedure \([176]\).

2.8 Conservation, Symplecticity and Symmetry

All consistent DIRK, SDIRK and ESDIRK methods conserve linear first integrals \([188]\). To conserve quadratic and cubic first integrals, one focuses on the symmetric matrices

\[
M_{ij} = b_i a_{ij} + b_j a_{ji} - b_i b_j \quad (61)
\]

\[
M_{ijk} = b_i a_{ij} a_{jk} + b_j a_{jk} a_{ki} + b_k a_{ki} a_{ij} - (b_i b_j a_{ik} + b_j b_k a_{kj} + b_k b_i a_{jk}) + b_i b_j b_k \quad (62)
\]

with magnitudes that may be defined as \(||M_{ij}|| = \left( \sum_{i,j=1}^{s} M_{ij} M_{ij} \right)^{1/2}\) and \(||M_{ijk}|| = \left( \sum_{i,j,k=1}^{s} M_{ijk} M_{ijk} \right)^{1/2}\). Methods having \(M_{ij} = 0\) conserve certain quadratic first
integrals, but no Runge-Kutta method can simultaneously achieve $M_{ij} = 0$ and $M_{ijk} = 0$ [85, 188, 230]. Instead, minimizing their magnitudes seems prudent.

Symplectic DIRK methods [188, 342, 355, 420, 422, 464] satisfy $M_{ij} = O$ and, consequently, must have the structure

$$
\begin{array}{c|cccccc}
  c_1 & b_1/2 & 0 & 0 & 0 & \cdots & 0 \\
  c_2 & b_1 & b_2/2 & 0 & 0 & \cdots & 0 \\
  c_3 & b_1 & b_2 & b_3/2 & 0 & \cdots & \vdots \\
  \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\
  c_{s-1} & b_1 & b_2 & b_3 & \cdots & b_{s-1}/2 & 0 \\
  c_s & b_1 & b_2 & b_3 & \cdots & b_{s-1} & b_s/2 \\
\end{array}
$$

(63)

where $b_i \neq 0$. This implies that EDIRKs, ESDIRKs and QESDIRKs cannot be symplectic. Further, for SDIRKs, symplecticity requires that $b_1 = b_2 = \cdots = b_s = 2\gamma$. The stability function for symplectic, DIRK-type methods is given by [152]

$$
R(z) = \frac{(1 + a_{11}z)(1 + a_{22}z) \cdots (1 + a_{ss}z)}{(1 - a_{11}z)(1 - a_{22}z) \cdots (1 - a_{ss}z)}
$$

(64)

and, hence, $R(z)R(-z) = 1$, $E(y) = 0$ and the stability function is P-stable; $|R(i\nu)| = 1$ where $\nu$ is a real number. Qin and Zhang [342, 464] give two useful low-order symplectic (S)DIRKs of order two in one- and two-stages

$$
\begin{array}{c|c|c|c}
  1/2 & 1/2 & 1/2 \\
  1/4 & 1/2 & 1/2 \\
  1/2 & 1/2 & 1/2 \\
\end{array}
\begin{array}{c|c|c|c|c}
  1/2 + a & 1/2 + a & 0 & 0 \\
  1+2a & 1-2a & -2a & 0 \\
  1+2a & 1-4a & -2a & 1+a \\
\end{array}
$$

(65)

The first method is the one-stage, A-stable Gauss method. The second method is A-stable also. A three-stage, fourth-order method in (65) is offered by Sanz-Serna and Abia [354] and Sun [420], in three stages where $a = (\sqrt{2} - 1) \approx 0.4142$ and $a_{22} \approx 0.5132$. Negative entries on the diagonal make them of little use for stiff problems because the stability function is not analytic in the complex LHP and, therefore, A-stability is impossible. The maximum order of a symplectic DIRK is four [420]. Jiang [247] solved the constraint equations of the four-stage, fourth-order symplectic DIRK and found nine solutions, one of which has already been given by Qin and Zhang [342]. All solutions have at least one negative $b_i$ and, consequently, $a_{ii}$. Franco et al. [154] also derive a fourth-order symplectic DIRK where $a_{22} = a_{44} < 0$ but in five stages rather than three or four

$$
\begin{array}{c|cccc}
  b_1/2 & b_1/2 & 0 & 0 & 0 \\
  b_1 + b_2/2 & b_1 & b_2/2 & 0 & 0 \\
  1/2 & b_1 & b_2 & b_3/2 & 0 \\
  1 - b_1 - b_2/2 & b_1 & b_2 & b_3 & b_2/2 \\
  1 - b_1/2 & b_1 & b_2 & b_3 & b_2 & b_1/2 \\
\end{array}
$$

(66)
with $b_1 = 1.452223059167653$, $b_2 = -2.150611289942164$ and $b_3 = 2.396776461549028$. Generally, symplecticity is lost in a variable step-size implementation without some supplementary procedure [185, 188].

Since this Butcher array is quite restrictive, one could consider a pseudo-symplectic approach like that of Aubry and Chartier [26] or the low-drift approach of Moan [301]. The former approach is greatly facilitated by simplifying assumption $D(1)$. As stiffly-accurate methods do not permit the use of $D(1)$, constructing pseudo-symplectic DIRK-type methods that are also stiffly-accurate will be particularly difficult. Examples of such an approach do not appear to exist. Lossless methods [158] require methods that are not only symplectic but also algebraically stable with elements of $b$ being identical. As with symplectic methods, lossless methods cannot be L-stable.

Symmetric methods are appropriate for solving reversible equations. From Stetter [416], Hairer et al. [188], Kulikov [276, 277] and Franco and Gómez [152] symmetric five- and four-stage DIRKs are given by

$$
\begin{align*}
&c_1 & a_{11} & 0 & 0 & 0 & 0 \\
&c_2 & b_1 & a_{22} & 0 & 0 & 0 \\
&1/2 & b_1 & b_2 & b_3/2 & 0 & 0 \\
&1 - c_2 & b_1 & b_2 & b_3 & b_2 - a_{22} & 0 \\
&1 - c_1 & b_1 & b_2 & b_3 & b_2 & b_1 - a_{11} \\
\end{align*}
$$

This is a slight variation of (51). For $M \geq O$, $a_{11} = a_{ss}$, $a_{22} = a_{s-1,s-1}$, $\cdots$, $a_{ii} = a_{s+1-i,s+1-i}$, which implies $b_i = 2a_{ii}$. By symmetry, the order of a symmetric DIRK must be even, and if the method is both irreducible and algebraically stable, then its order is no greater than two [276]. Franco and Gómez [152] derive four- and five-stage symmetric DIRKs to solve mildly stiff periodic problems. All four symplectic methods listed in (65) and (66) are also symmetric. Like pseudo-symplectic methods, one may also attempt pseudo-symmetric methods [105].
2.9 Dissipation and Dispersion Accuracy

A phase-lag analysis of DIRK-type methods is based on linear ODEs with oscillatory solutions (imaginary axis eigenvalues) and usually considers the test equation [154, 155, 212, 262]

\[
\frac{dU}{dt} = i\omega U + \delta e^{i\omega t}, \quad \omega \neq \omega_p,
\]

where the natural frequency \(\omega\), the forcing frequency \(\omega_p\), and \(\delta\) are real numbers. The form of the equation containing the natural frequency is motivated by the temporal evolution of a single wave, \(U = e^{i\omega t}\). Differentiating both sides yields \(U' = i\omega U\).

Phase-lag error consists of a linearly increasing (in time) homogeneous phase error associated with \(\omega\) and a constant inhomogeneous phase error associated with \(\omega_p\). Hence, homogeneous phase error is the primary concern for long-term integration. Methods are generally optimized to minimize the dispersive and dissipative errors associated with the free oscillations rather than the forced or combined oscillations.

The linear stability function, associated with the test equation, \(R(z) = P(z)/Q(z)\), is \(R(z) = 1 + \sum_{i=1}^{\infty} r_i z^i\), \(P(z) = 1 + \sum_{i=1}^{\delta} p_i z^i\) and \(Q(z) = 1 + \sum_{i=1}^{\delta} q_i z^i\), or, componentwise using the multiplication and division of power series, \(r_i = p_i - \sum_{k=1}^{i} q_k r_{i-k}, \quad p_i = \sum_{k=1}^{i} r_k q_{i-k}\), where \(z = i\omega(\Delta t)\). Note that \(r_i = 1/i!, \quad i = 1, 2, \cdots, p\) where, again, \(p\) is the order of the method. Defining \(\nu = \omega(\Delta t)\), one may write the real and imaginary portions of the stability function,

\[
\mathcal{R} = \Re(R(i\nu)) = A(\nu^2) = \sum_{j=1}^{\infty} (-1)^j r_{2j} \nu^{2j},
\]

and also write the homogeneous phase or dispersion

\[
\phi(\nu) = \nu - \arg(R(i\nu)) = \nu - \arctan \left( \frac{\mathcal{I}}{\mathcal{R}} \right) = \sum_{j=0}^{2j+1} \phi_{2j+1} \nu^{2j+1} = O(\nu^{\text{disp}+1})
\]

and homogeneous amplification or dissipation errors as [212]

\[
\alpha(\nu) = 1 - |R(i\nu)| = 1 - \sqrt{\mathcal{R}^2 + \mathcal{I}^2} = \sum_{j=0}^{2j} \alpha_{2j} \nu^{2j} = O(\nu^{\text{diss}+1})
\]

where \(p_{\text{disp}}\) and \(p_{\text{diss}}\) are the respective homogeneous dispersion and dissipation orders of accuracy. Hence, \(R(i\nu) = \mathcal{R} + i\mathcal{I} = (1 - \alpha(\nu)) e^{i(\nu - \phi(\nu))}\). A Runge-Kutta method is dispersive of order \(p_{\text{disp}}\), where \(p_{\text{disp}}\) is even, if [155, 212]

\[
\frac{r_0}{(j-0)!} - \frac{r_1}{(j-1)!} + \frac{r_2}{(j-2)!} - \cdots + (-1)^j r_j \frac{1}{(j-j)!} = \sum_{i=0}^{j} \frac{(-1)^i r_i}{(j-i)!} = 0,
\]

where \(j = 1, 3, 5, \cdots (p_{\text{disp}} - 1)\). A Runge-Kutta method is dissipative of order \(p_{\text{diss}}\), where \(p_{\text{diss}}\) is odd, if the E-polynomial satisfies \(E(\nu) = O(\nu^{p_{\text{diss}}+1})\) or \(E_{2j} = 0, \quad 2j \leq p_{\text{diss}}\).
For SDIRKs and ESDIRKs, in cases where \( s_I \leq p \leq s_I + 1 \) and \( p_{s_I} \neq 0 \), or where \( s_I - 1 \leq p \leq s_I \) and \( p_{s_I} = 0 \), then \( R(z) \), \( P(z) \) and \( Q(z) \) become

\[
ri = pi - \sum_{k=1}^{i} \binom{s_I}{k} (-\gamma)^k r_{i-k}, \quad pi = \sum_{k=1}^{i} rk \binom{s_I}{i-k} (-\gamma)^{i-k}, \quad qi = \binom{s_I}{i} (-\gamma)^i. \tag{75}
\]

Additionally, the E-polynomial terms are given by

\[
E_{2j} = \binom{s_I}{j} \gamma^{2j} - p_j^2 - 2 \sum_{i=1}^{s} p_{j-i} p_{j+i} = 0, \quad j = 0, 1, \ldots, (p_{\text{diss}} - 1)/2, \tag{76}
\]

where \( p_k \) is generally nonzero for \( 0 \leq k \leq s_I \).

Van der Houwen and Sommeijer [212] were among the first to consider phase lag optimized implicit Runge-Kutta methods. They construct an \( s \)-stage family of third-order SDIRKs with enhanced dispersion orders given by

\[
\begin{array}{cccccccccc}
gamma & \gamma & 0 & 0 & 0 & \cdots & 0 \\
c_2 & c_2 - \gamma & \gamma & 0 & 0 & \cdots & 0 \\
c_3 & 0 & c_3 - \gamma & \gamma & 0 & \cdots & 0 \\
c_4 & 0 & 0 & c_4 - \gamma & \gamma & \ddots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & 0 \\
c_s & 0 & 0 & \cdots & 0 & c_s - \gamma & \gamma \\
\end{array}
\tag{77}
\]

Denoting the DIRK methods as DIRK\((s, p, p_{\text{disp}}, p_{\text{diss}})\), two A-stable methods with enhanced dispersion order are derived: SDIRK\((3, 3, 6, 3)\) and SDIRK\((4, 3, 8, 3)\). Their performances are then contrasted against SDIRK-NC323 and SDIRK-NC34 given in (50). Franco et al. [155] reconsider these two methods but also concern themselves with dissipation order. A-stable methods, SDIRK\((3, 3, 4, 5)\) and SDIRK\((4, 3, 4, 7)\), are derived following the Butcher array structure given by van der Houwen and Sommeijer. They show that the A-stable, SDIRK\((4, 3, 6, 5)\) method does not exist. The methods are then compared against the SDIRK\((3, 3, 6, 3)\) and SDIRK\((4, 3, 8, 3)\) of van der Houwen and Sommeijer. It is remarked that dissipation order may be more useful than dispersion order. In another paper on phase-lag optimized methods, Franco et al. [154] contrast SDIRK\((3, 3, 4, 5)\), DIRK\((3, 4, 4, \infty)\) (65), a newly constructed SDIRK\((5, 4, 6, 7)\) derived based on a modification of the values of \( b_i \) in (77), and DIRK\((5, 4, 6, \infty)\), (66). Each SDIRK method is A-stable, but the DIRK methods are symmetric and symplectic. Results favored the SDIRKs over the symmetric DIRKs. Although Koto [262] does not actually derive SDIRK methods with higher dispersion order, he does establish some of what is possible based on \( R(z) \) and \( \gamma \). Increased dissipation order is not considered. Table 7 summarizes the methods developed to date. Franco and Gómez [152] show that the maximum dispersion order for a symmetric SDIRK is two. They also derive four- and five-stage symmetric DIRKs with increased dispersion order. Each have negative diagonal entries; hence, the methods cannot be A-stable.
Table 7. Phase error optimized, A-stable, SDIRK methods.

<table>
<thead>
<tr>
<th>Reference</th>
<th>$s$</th>
<th>$p$</th>
<th>$p_{\text{disp}}$</th>
<th>$p_{\text{disss}}$</th>
<th>$\gamma$</th>
<th>$R(-\infty)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[212]</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0.788675134594812882254574</td>
<td>+1.000</td>
</tr>
<tr>
<td>[212]</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>0.975674588694440299995000</td>
<td>−0.679</td>
</tr>
<tr>
<td>[212]</td>
<td>4</td>
<td>3</td>
<td>8</td>
<td>3</td>
<td>1.129726566183898058039659</td>
<td>−0.627</td>
</tr>
<tr>
<td>[262]</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[262]</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>0.451551228989386200139028</td>
<td>+0.914</td>
</tr>
<tr>
<td>[262]</td>
<td>6</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>0.505024581119690721793836</td>
<td>+0.816</td>
</tr>
<tr>
<td>[155]</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>1.068579021301628806418834</td>
<td>−0.630</td>
</tr>
<tr>
<td>[155]</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[155]</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>1.280579761275305457302484</td>
<td>−0.596</td>
</tr>
<tr>
<td>[154]</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>0.473268391258295324455589</td>
<td>+0.837</td>
</tr>
</tbody>
</table>

2.10 Memory Economization

Although economization of computer memory usage is an appealing goal, it is less of an issue with DIRK-type methods than it is with ERK methods at the level of scheme coefficients. This is because the majority of the memory burden associated with the implicit integration method is caught up in the solution of coupled nonlinear stage and step equations. Compromising the Runge-Kutta coefficients by using a van der Houwen or Williamson low storage strategy [256] to avoid saving several function or derivative values may strongly degrade the quality of the scheme and result in only small net memory usage savings. For this reason, little effort is expended in this paper to reduce memory usage via the Runge-Kutta coefficients. Efforts in this direction, however, have been made by van der Houwen and Sommeijer [215] and Yoh and Zhong [459, 460]. Energy might be better expended on memory reduction of the nonlinear equation solver. Enright [141] offers an approach to reduce memory usage for direct solvers. The common approach to reducing memory usage for the algebraic equation solver while performing indirect solves is to use a Krylov subspace technique [46].

2.11 Regularity

It is natural to ask that a numerical integrator preserve invariant objects of the ODEs being integrated such as fixed points [186]. Runge-Kutta methods inherit all fixed points of the underlying ODEs but may create additional spurious ones [225]. From Iserles et al. [228], the regularity of integration methods for first-order ODEs focuses on numerical solution bifurcations. The solution bifurcations may give rise to spurious steady, or fixed point, solutions ($R^{[1]}$-regularity), spurious period doubling solutions ($R^{[2]}$-regularity) and spurious invariant curves ($R^{[H]}$-regularity). The majority of regularity results consider constant and positive step-sizes. It is possible that step-size selection may reduce or eliminate spurious fixed points [186], an idea confirmed for ERK methods [27,197]. It is also possible that as the number
of equations being integrated increases, there is a reduced chance of computing a spurious fixed point [27]. Methods that are not regular are termed irregular.

Runge-Kutta methods which are both $R^{[1]}$- and $R^{[2]}$-regular ($R^{[1,2]}$-regular) can be no higher than second-order [228]. Iserles [225] shows that for an irreducible Runge-Kutta method with an order greater than or equal to two, then $\sum_{i=1}^{s} a_{ii} = 1/2$ is required for $R^{[1]}$-regularity. From Table 4, it may be seen that this requirement cannot be met for A-stable SDIRK and ESDIRK methods having $p = s_{f}$ or $p = s_{f} + 1$ over third-order and L-stable methods having $p = s_{f} - 1$ or $p = s_{f}$ over second-order. The maximum order of an A-stable, $R^{[1]}$-regular Runge-Kutta method is four, while the maximum order of any $R^{[1]}$-regular DIRK method having $a_{ii} \geq 0$ is three [186]. Jackiewicz et al. [244] propose a more severe regularity requirement, stage-wise ($R^{[i]}$-regularity), or strong $R^{[1]}$-regularity. The appeal of strong $R^{[1]}$-regularity is that there is a procedure whereby one may construct a regular method; there is currently no such procedure for $R^{[1]}$-regularity. The requirement is $a_{ik} - a_{jk} = \xi b_{k}$ for all $i \neq j$. Summing over $k$ and using $\mathbf{Ae} = \mathbf{c}$ and $\mathbf{b} = 1$, one has $c_{i} - c_{j} = \xi$; hence, $a_{ik} - a_{jk} = (c_{i} - c_{j})b_{k}$. Because $(c_{i} - c_{j})$ is antisymmetric, it has $s(s + 1)/2$ unique components. Summing over $k$, it is seen that at least one value of $k$ is not independent. Further, one may take linear combinations of the previous equations to create many of the others. Ultimately, this constitutes $s(s + 1)/2$ additional conditions: $j = 1, i = 2, 3, \ldots, s$, and $k = 2, 3, \ldots, i$. At two stages, Jackiewicz et al. [244] offer the following one parameter family of second-order, A-stable, strongly regular SDIRKs

$$
\begin{bmatrix}
\frac{1}{4} & \frac{1}{2} & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
-1 & 2 & -1 \\
\end{bmatrix}
\begin{bmatrix}
\xi \\
\xi \\
\xi \\
\end{bmatrix}
$$

At $c_{2} = 3/4$, $A^{(3)} = \sqrt{5}/96$, the method, also given in (65), is strongly regular, A-stable, algebraically stable, it satisfies $R_{\text{int}}(-\infty) = \{0, 0\}$, and is symplectic and symmetric, but it is not $L$-stable as $R(-\infty) = 1$. It also forms the first two stages of SDIRK4 [193]. It will be referred to as SDIRK2(1)2A[1]Alg. At $c_{2} = 5/6$, third-order error is minimized. Gan [162] also considers two-stage, strongly-regular DIRK methods.

Less severe forms of $R^{[1]}$-regularity are offered by Vadillo [432]: Bifurcation (B-regularity), Real (R-regularity) and BR-regularity. B-regularity allows a constructive approach using the matrix $\mathcal{B}(\lambda)$ and the parameter $\lambda$; i.e.,

$$
\mathcal{B}(\lambda) = \begin{bmatrix}
(I - \lambda A) & \mathbf{e} \\
\mathbf{b}^{T} & 0 \\
\end{bmatrix}.
$$

It requires that $\det[\mathcal{B}(\lambda)] \neq 0$ for all $\lambda \neq 0$. BR-regularity relaxes this requirement slightly by restricting $\lambda$ to be real. Vadillo derives the following two-parameter family of third-order, B-regular, SDIRKs

$$
\begin{bmatrix}
\frac{1}{6} & \frac{1}{6} & 0 & 0 \\
\frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 \\
\frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\
\end{bmatrix}
\begin{bmatrix}
\frac{0}{(6c_{3} - 1)(3 - 24c_{2} + 24c_{2}^{2} + 2c_{3})} \\
\frac{0}{3(4c_{2} - 3)(6c_{2} - 1)} \\
\frac{0}{2(6c_{2} - 1)(c_{2} - c_{3})} \\
\end{bmatrix}
$$

38
for which $\text{Det}[B(\lambda)] = -1$. There are no values of $c_2$ and $c_3$ that yield an A-stable method. Weaker still, Xiao et al. [455] defines weak $R^{[1]}$-regularity that is achieved by all consistent Runge-Kutta methods.

Xiao et al. [455] also define a weak $R^{[2]}$-regularity. Runge-Kutta methods that have $M > O$ and $b \geq 0$ are weakly $R^{[2]}$-regular for positive step-sizes. In terms of Hopf bifurcations, Iserles et al. [228] state that symplectic Runge-Kutta methods with positive weights, $b_i \geq 0$, are $R^{[H]}$-regular. Further, a consistent Runge-Kutta method is $R^{[H]}$-regular if and only if $R(z)$ is symmetric [$R(z)R(-z) = 1$ or (64)], and the method is A-stable. Methods that are $R^{[1,2,H]}$-regular are termed strictly regular.

### 2.12 Boundary and Smoothness Order Reduction

A common application of DIRK-type methods is to semidiscretized PDEs. The resulting initial boundary value problem (IBVP) solution may fail to exhibit the classical global convergence rate that one might expect. Crouzeix [120,123] appears to have been the first to point this out. The phenomenon is attributable to the spatial boundaries (space-time problems) due to the interaction of the boundary conditions with the low stage-order information. It may also occur due to a lack of spatial smoothness of the RHS for the same reason. The degree of order reduction depends on the type of problem being integrated, the type of boundary condition being applied and the spatial smoothness of the solution. A cursory summary of boundary and smoothness order reduction results relevant to DIRK-type methods is provided, focusing primarily on its severity under different circumstances. Global convergence rates generally appear as the minimum of the classical order $p$ and the stage-order $q$, plus some constant, i.e., a best- and worst-case scenario.

Boundary order reduction of implicit Runge-Kutta methods in the context of linear equations, $u'(t) = A(t)u(t) + f(t)$ (nonautonomous) and $u'(t) = Au(t) + f(t)$ (autonomous), is the best understood. One of the earliest efforts on this matter was by Verwer [437]. He considers the implicit Euler, the one-stage, second-order Gauss method, SDIRK-NCS23 and SDIRK-NC34 methods on the nonhomogeneous problem using either homogeneous or inhomogeneous Dirichlet boundary conditions. Empirically derived global convergence rates were $\text{Min}(p,q+2+1/3)$ and $\text{Min}(p,q+1+1/4)$ in these two cases, respectively. If the value of $A$ was nonsmooth, this is further reduced to $\text{Min}(p,q+1)$. Ostermann and Roche [328] performed an analytical study of the autonomous nonhomogeneous problem, requiring $A(\alpha)$-stable methods and, under certain restrictions, derived global convergence rates. For step-sizes $h_0 \leq h \leq H$, homogeneous boundary conditions gave rise to global convergence rates of $\text{Min}(p,q+2+\nu)$, while nonhomogeneous conditions yielded $\text{Min}(p,q+1+\nu)$. The reader is referred to Ostermann and Roche [328] for the definition of $\nu$. Smaller step-sizes $h \leq h_0$ return the classical convergence rate $p$. Refining this, Lubich and Ostermann [290] solved the one-dimensional inhomogeneous heat equation and determined global order to be $\text{Min}(p,q+1+\beta)$. Dirichlet conditions produced $\beta = 5/4$ and $\beta = 1/4$ in the homogeneous and inhomogeneous cases, respectively. Corresponding results for Neumann conditions were
found to be $\beta = 7/4$ and $\beta = 3/4$. Increasing the dimensionality and considering the nonautonomous problem, Lubich and Ostermann [292] find that the convergence rate is unaffected by periodic boundary conditions; however, Neumann conditions have lower convergence rates, $\text{Min}(p, q + 1 + \beta)$, in two and three dimensions where $\beta = 1/4$, than in one dimension where $\beta = 5/4$. Lubich and Ostermann [293] then showed that the boundary order reduction was confined to the boundary, and that the interior manifested superconvergence to the classical order of the method. Nonsmooth data also exhibited localized order reduction. Results from each of these papers assumed an invertible $A$-matrix and strongly $A(\alpha)$-stable methods. Zouraris [467] derives a global convergence rate of $\text{Min}(p, q + 1 + 1/2)$ for SDIRK-NC34 solving the nonautonomous problem with Neumann conditions. In the case of variable step-sizes applied to the nonautonomous problem, González and Ostermann [169] show that results are similar to the fixed step-size case. Using a different analysis tool than Ostermann and Roche [328], Alonso-Mallo and Palencia [15] consider $A(\alpha)$-stable methods but only require that the lower left $(s-1) \times (s-1)$ block of the $A$-matrix be invertible (thus allowing EDIRKs, ESDIRKs and QESDIRKs). On nonhomogeneous hyperbolic and parabolic problems, the global convergence rate was determined to be $\text{Min}(p, q + 1 + \nu)$ for $|R(-\infty)| < 1$ and $\text{Min}(p, q + \nu + \nu/m)$ for $|R(-\infty)| = 0$ where $m$ is the multiplicity of the root at $z = -\infty$ for the function $R(z) - 1 = 0$. Test problems using SDIRK-NC34 and homogeneous Dirichlet conditions on the autonomous parabolic equations showed $0 \leq \nu \leq 5/4$. Inhomogeneous boundary conditions lowered this to $0 \leq \nu \leq 1/4$. Solving an inhomogeneous hyperbolic equation with inhomogeneous Dirichlet conditions showed $0 \leq \nu \leq 1/2$. Trompert and Verwer [429] consider order reduction on linear PDEs in the presence of local spatial grid refinement while using a stage-order two ESDIRK.

To address the boundary order reduction associated with linear problems, corrective action may take place with the Runge-Kutta coefficients of the scheme or by correcting each of the stage and step values. To date, we are not aware of any efforts to remedy boundary order reduction by using Butcher coefficients for DIRK-type methods. Koto [264] does this with explicit Runge-Kutta methods but the resulting methods are relatively inefficient. Lubich and Ostermann [291] provide additional order conditions to deal with boundary order reduction for methods closely related to DIRK-type methods, Rosenbrock methods. Recently, Alonso-Mallo [11,12], Alonso-Mallo and Cano [13], and Calvo and Palencia [86] devise alternative corrective procedures applied to algebraically stable SDIRKs to retrieve full temporal convergence rates.

Fewer results exist on semilinear, quasilinear and nonlinear problems. Verwer [437] solves Burgers’ equation numerically by using SDIRK-NCS23 and SDIRK-NC34 and he empirically observes global convergence rates of $\text{Min}(p, q + 2 + 2/3)$ and $\text{Min}(p, q + 1 + 1/4)$ with homogeneous and inhomogeneous Dirichlet conditions, respectively. For a nonlinear problem with nonsmooth coefficients, this convergence rate was reduced to $\text{Min}(p, q + 1)$. Lubich and Ostermann [292] conclude that, with the possible exception of 3-D problems using Neumann boundary conditions, results for quasilinear problems $u'(t) = A(u(t))u(t) + f(t)$ were the same as those for linear nonautonomous problems. For semilinear problems $u'(t) = A(t)u(t) +$
g(t, u(t)), Lubich and Ostermann [290] find that homogeneous Dirichlet conditions yields \(\min(p, q + 1 + 3/4)\) on the semilinear problem but \(\min(p, q + 1 + 5/4)\) on the linear problem. Nonlinear equations exhibit boundary order reduction most severely. Ostermann and Thalhammer [329] study strongly \(A(\alpha)\)-stable methods applied to parabolic nonlinear equations and obtain convergence rates of \(\min(p, q+1)\) independent of whether the step-size was fixed or not. Calvo et al. [81] provide a method for avoiding order reduction on nonlinear problems when using IMEX additive Runge-Kutta methods. The implicit portion of these third- and fourth-order IMEX methods are SDIRKs padded with a column of zeros \((a_{i1} = 0, i = 1, 2, \ldots, s)\).

### 2.13 Efficiency

Quantifying the efficiency of DIRK-type methods for purposes of comparing methods is far more difficult than it is for explicit Runge-Kutta methods [256]. Because the methods are ideally never stability bound, one may focus on accuracy efficiency alone, i.e., the work required per unit of physical time at fixed integration error. To do this, however, one must quantify both accuracy and work. If there is no order reduction, accuracy of methods may be compared by using the leading order truncation error, \(A^{(p+1)}\). In the presence of order reduction, the proper analogue of \(A^{(p+1)}\) is not clear. Matters are made even more difficult on stiff problems because one is confronted with different accuracies for the differential variables and the emerging algebraic variables. Measuring work of a method may be affected by many things that are intrinsic to the overall method, including the magnitude of \(\gamma\), the spacing of the abscissae, the quality of the stage-value prediction, the stage-order and the many stability characteristics of the method. It is also affected by issues not intrinsic to the method, including iteration control and the implementation strategy. In certain cases, confluent methods may reduce work because the iteration on the second stage, having the repeated abscissae, begins with a high-quality starting guess. Succeeding at measuring work would also benefit a variable-order implementation. Capitalizing on a quantitative measure of relative efficiency, several qualitative guidelines seem reasonable. Small values of \(\gamma\), high stage-order, \(L\)-stability, internal stability coupled with high-quality stage-value predictors and iteration control are likely to enhance efficiency. Some work savings are also possible when using stiffly-accurate EDIRK, ESDIRK and QESDIRK methods by stage-derivative reuse [324].

### 2.14 Solvability

Before proceeding to solve for the stage- and step-values, \(U^k\) and \(U^{[n+1]}\), it would be prudent to understand under what conditions the coupled nonlinear stage and step equations possess a unique solution [76, 77, 80, 129, 193, 269, 470]. If the equations do possess a unique solution, the method is termed feasible [410]. For inner product norms and differential equations having a one-sided Lipschitz constant \(\nu\), the nonlinear stage equations of DIRK or SDIRK method, having positive \(a_{ii}\), possess a unique solution for step-sizes given by \([193, 302]\ \nu \Delta t < \min \left(\frac{1}{a_{ii}}\right), i = 1, 2, \ldots, s\). For EDIRKs, ESDIRKs and QESDIRKs, where \(a_{11} = 0\) and \(a_{ij}\) is not invertible,
the previous result is likely to still apply, but with $i = 2, 3, \cdots, s$ [193]. To make this relation practical, one must estimate the value of $\nu$ for their problem. We are not aware of any code that explicitly uses this criterion as part of a step-size algorithm. Calvo et al. [80] consider this same topic by using the maximum norm but offer no comparable result for DIRK-type methods. Kraaijevanger [266] also considers the maximum norm and discusses solvability for both unconditionally and conditionally contractive methods.

2.15 Implementation

Depending on the size of the equation system, one solves the implicit algebraic equations in different ways. For small systems of equations, one generally uses direct solvers. In this case, using the definition $F_j = F(U_j, t^{[n]} + c_j \Delta t)$, one must solve

$$U_i = U^{[n]} + X_i + (\Delta t)a_{ii}F_i, \quad X_i = (\Delta t) \sum_{j=1}^{i-1} a_{ij}F_j, \quad 1 \leq i \leq s \quad (81)$$

where $X_i$ is explicitly computed from existing data. Combining (81) with an appropriate starting guess, a modified Newton iteration [180, 210, 371, 372, 378] provides $U_i$ and $F_i$. This is accomplished by solving

$$(\mathbf{I} - (\Delta t)a_{ii}\mathbf{J}) (U_{i,k+1} - U_{i,k}) = -\left(U_{i,k} - U^{[n]}\right) + X_i + (\Delta t)a_{ii}F_i \quad (82)$$

where the subscript $k$ denotes the value on the $k$th iteration, $(\partial F/\partial U) = \mathbf{J}$ is the Jacobian, $\mathbf{I}$ is the identity matrix, $(\mathbf{I} - (\Delta t)a_{ii}\mathbf{J}) = \mathbf{N}$ is the (Newton) iteration matrix, and $(U_{i,k+1} - U_{i,k}) = \mathbf{d}_{i,k}$ is the displacement. The RHS of (82) is called the residual, $\mathbf{r}_{i,k}$ where

$$\mathbf{Nd}_{i,k} = \mathbf{r}_{i,k}. \quad (83)$$

Solving for the displacement vector is generally done using Gaussian elimination where the iteration matrix is factorized into a permutation matrix $\mathbf{P}$, for pivoting and a lower, $\mathbf{L}$, and upper, $\mathbf{U}$, triangular matrix. Shampine remarks that although the iteration matrix for stiff problems is often ill-conditioned, the linear systems associated with it are not [377]. On the $k$th-iteration, one has

$$\mathbf{PLUd}_{i,k} = \mathbf{r}_{i,k}, \quad U_{i,k+1} = U_{i,k} + \mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{P}^{-1}\mathbf{r}_{i,k}. \quad (84)$$

Jacobians are generally not reevaluated at each stage but rather after many steps unless the problem is extremely stiff. The iteration is terminated when either the normalized $d_{i,k} = d_{i,k}/(|U_{i,k}| + \eta/\epsilon)$ (displacement test) or the normalized $r_{i,k} = r_{i,k}/(|U_{i,k}| + \eta/\epsilon)$ (residual test) are sufficiently small [210, 324, 413, 414],

$$\tau_{\text{resid}} = c_\epsilon \epsilon \leq \|d_{i,k}||, \quad \text{or} \quad \tau_{\text{displ}} = c_\epsilon \epsilon \leq \|r_{i,k}||, \quad c_\epsilon \approx 0.005 - 0.1, \quad (85)$$

where $\epsilon$ is the user-specified integration relative error tolerance, $\eta$ is the user-specified integration absolute error tolerance, $c_\epsilon$ is the tolerance ratio, and $\eta$ is an absolute
error tolerance that need not be equal for each equation [45, 206, 343, 378]. Nørsett and Thomsen [313] investigate the proper choice for the tolerance ratio for SDIRKs. If the selected ratio is too small, then unnecessary iterations are performed; if it is too large, then local and global error will suffer. They select \( c_\varepsilon = 2\|(b - \hat{b})^T A^{-1}\|^{-1} \) where they consider embedded methods having order \( p + 1 \). Displacements (and residuals) may be evaluated in either an \( L_\infty \) or an \( L_2 \) norm,

\[
\|d_{i,k}\|_\infty = \max_i |d_{i,k}|, \quad \|d_{i,k}\|_2 = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (d_{i,k})^2}.
\]  

(Hindmarsh [206, 377] chooses the displacement test for LSODE [343] because the residual magnifies the error in the stiff ODE case. Two slightly different convergence rates may be defined as

\[
\rho_m = (\|d_{i,m}\|/\|d_{i,1}\|)^{1/m}, \quad \alpha_k = \|d_{i,k}\|/\|d_{i,k-1}\|, \quad \alpha = \max_k \alpha_k
\]

In PVODE [206], the convergence test is \( R\|d_{i,k}\|_2 < 0.1\varepsilon \) where \( R \) is a constant given in the reference [206]. DASSL [45] requires that \( \frac{\epsilon_0}{1 - \rho_m} \|d_{i,m}\|_2 < \epsilon/3 \). Hairer and Wanner [193] use \( \frac{\alpha_k}{1 - \alpha_k} \|d_{i,k}\|_2 \leq c_\varepsilon \varepsilon \). Houbak et al. [210, 313] suggest a combination of displacement and residual tests. Shampine [372] insists that accepting an iteration based on a displacement test must be accompanied by a convergence rate \( \alpha \) of less than unity. Hence, at least two iterations must be computed. Shampine [372] advocates \( \alpha \) as the most accurate indicator of convergence rate. Divergence is defined as a convergence rate of \( \alpha_k > 2 \) by Hindmarsh [206], \( \alpha_k > 1 \) by Shampine [372], and \( \rho_m > 0.9 \) by Brenan et al. [45]. For stiffly-accurate, first-stage explicit DIRK-type methods, Olsson and Söderlind [324] recommend stage derivative reuse; the values of \( F_s = F^{[n+1]} \) at the end of one step are used immediately as \( F_1 \) in the following step.

In mildly stiff cases where either the time step or \( a_{ii} \) are changing rapidly but the Jacobian is still sufficiently accurate, computational savings are possible. The scaled iteration matrix is defined

\[
N_s = -\frac{1}{a_{ii}(\Delta t)} N = \left[ J - \frac{1}{a_{ii}(\Delta t)} I \right]
\]

for the purposes of changing either \( a_{ii} \) or \( \Delta t \) cheaply. Equation (82) is modified accordingly. Following Enright [141], decomposing \( N_s \) into a lower triangular matrix, its inverse and an upper Hessenberg matrix, \( N_s = LHL^{-1} \), one may update \( N_s \) with updated \( (a_{ii})_2 \) or \( \Delta t_2 \) via

\[
J - \frac{1}{(a_{ii})_2(\Delta t)_2} I = L \left[ H + \frac{(a_{ii})_2(\Delta t)_2 - (a_{ii})_1(\Delta t)_1}{(a_{ii})_1(a_{ii})_2(\Delta t)_1(\Delta t)_2} I \right] L^{-1}.
\]

An alternative procedure has been given by Krogh [206, 275, 462]. Instead of solving for the displacement as suggested above, a constant \( c \) is computed and a modified procedure is solved:

\[
d_{k}^{(i)} = cU^{-1}L^{-1}P^{-1}r_{k}^{(i)}, \quad c = \frac{2(a_{ii})_1(\Delta t)_1}{(a_{ii})_1(\Delta t)_1 + (a_{ii})_2(\Delta t)_2}.
\]
Either of these procedures may be useful in the implementation of DIRK, EDIRK, or QESDIRK methods where nonzero diagonal elements may have different values. Al-Rabeh [16] constructs \( p(p + 1) \) pairs with \( a_{ii} = \gamma \) left as a free parameter. As the step size is changed, \( \gamma \) is changed, within reason, such that the product contained within the iteration matrix \( \gamma(\Delta t) \) remains fixed.

In the event that the system of ODEs becomes nonstiff intermittently during the computation, one may wish to perform stiffness switching [65,314,338]. The idea is to switch between a Newton-type iteration and a fixed-point, or Picard iteration, or even an explicit Runge-Kutta method. To do this, one first estimates the value of the Lipschitz constant for the problem at hand as \( ||J||_2 \) or, less precisely but more easily, as \( ||J||_\infty \). By selecting \( (\Delta t) \) to satisfy some error tolerance, stiffness is then defined as \( (\Delta t)||J||_2 > 1 \) [129,286,373,376,414]. Given that fixed-point iteration converges for \( (\Delta t)\gamma ||J||_2 < 1 \), for values larger than unity, a Newton iteration is used; for values less than unity, a fixed point iteration is used. Van der Houwen and Sommeijer [214] propose an intermediate solution, called approximate factorization iteration, for cases where \( (\Delta t)||J||_\infty \) is small. They switch between iteration procedures based on the value of \( a_{ii}(\Delta t)||J||_\infty \). Fixed-point iteration is used for \( a_{ii}(\Delta t)||J||_\infty < a \), approximate factorization is used for \( a \leq a_{ii}(\Delta t)||J||_\infty < b \) and Newton iteration is used for \( b \leq a_{ii}(\Delta t)||J||_\infty \), with \( a = 1/2 \) and \( b = 3 \). A complicating factor in this approach is that the estimation of the stiffness may be inaccurate. Higham and Trefethen [198] contend that the true characterization of stiffness comes not from the spectra of the Jacobian but its pseudospectra and the two may be different.

One further issue relevant to implementation using direct methods is the computation of the Jacobian. One may use a numerically generated Jacobian by using finite-difference techniques or use an analytically derived one [378]. It is a commonly held belief that an analytical Jacobian should be used whenever possible. Petzold comments that the numerical Jacobian is the weakest part of the DASSL code [45]. For certain problems, a numerical Jacobian can never be expected to work. A relatively recent development towards analytical Jacobians is automatic differentiation tools applied directly to the source code. These tools are available for FORTRAN 77 [40], FORTRAN 90 [341], and C [41]. When an analytical Jacobian is not feasible and a numerical one must be considered, one must compute

\[
\frac{\partial F_i}{\partial U_j} \approx \frac{F(U_i + \Delta_j e_j) - F(U_i)}{\Delta_j},
\]

where \( e_j = \{0, 0, \cdots , 1, \cdots , 0, 0\}^T \) is composed of only zeros except a single 1 in the \( j \)-th position. Its is apparent from the context in which it is used. The difficulties involved are choosing an appropriate \( \Delta_j \) and computing the Jacobian expeditiously. Most codes choose \( \Delta_j \) based on some variation of \( \Delta_j = \sqrt{u} \) Max \((|U_j|, \eta)\), where \( u \) is related to machine precision. It is the smallest number such that \( 1 + u > 1 \) on the computer of interest. To avoid division by zero, \( \eta \) is some absolute error tolerance. LSODE [343] and DASSL [45] offer slight variations on this idea.

Salane [349] offers a more sophisticated approach to compute a good \( \Delta_j \). First, he defines \( U^{(ref)} \) as a U-vector constructed using representative values of the elements.
Then the initial choice for $\Delta_j$ is given by

$$\Delta_j = \phi_j |U_j^{(\text{ref})}|, \quad \phi_j = u^{1/2}. \quad (92)$$

He then defines terms related to the scale $\Sigma_j$, $\sigma_j$ and the difference $\delta_j$; $\Sigma_j = \max_i(|F(U_i)|, |F(U_i + \Delta_j e_j)|)$, $\sigma_j = \min_i(|F(U_i)|, |F(U_i + \Delta_j e_j)|)$ and $\delta_j = \max_i(|F(U_i + \Delta_j e_j)| - |F(U_i)|)$. The goal is to adjust values of the $\phi_j$ at every Jacobian evaluation while not letting any value of $\phi_j$ be greater than a maximum, $\phi_{\text{max}} \approx u^{1/10}$, or less than a minimum, $\phi_{\text{min}} \approx u^{3/4}$. Further, the $j$th-column of the Jacobian is $j_j$.

Three FORTRAN subroutines are offered to compute the Jacobian. Any of these subroutines would be called repeatedly within the course of a typical integration. In the pseudocode contained in figure 1, it is assumed that the value of $F(U_i)$ has already been computed. Salane [348–350] and Ypma [463] discuss approaches to reduce the cost of computing numerical Jacobians.

Figure 1. Pseudocode to establish optimal numerical perturbation $\Delta_j$ for numerical Jacobians based on Salane [349].

For large systems of equations, a direct solve using the iteration matrix is not feasible. This is a common situation in the solution of stiff, spatially discretized PDEs. Chan and Jackson [104] and Jackson [245] review this issue, discussing sparse linear equation solvers and preconditioned Krylov subspace methods. The latter are used in the KRYSI (Krylov semi-implicit) code [207] and Bjurstrøm’s version of the GODESS (generic ODE solving system) code [42, 428]. We remark that the resulting algebraic system of equations resulting from backward differentiation.
formulae (BDF) methods is essentially identical to those resulting from DIRK-type methods. Rather than the modified Newton method that is used for small systems of equations, one usually considers inexact Newton methods for larger systems of coupled equations [207, 254, 344]. The implementation of this inexact Newton method in PVODEs is described by Hindmarsh [206]. Jothiprasad et al. [248] compare nonlinear multigrid with two variants of an inexact Newton scheme while using an ESDIRK method to solve the compressible Navier-Stokes equations. Isono and Zingg [240] contrast an approximate factorization algorithm with a Newton-Krylov algorithm in the study of unsteady, compressible flows using the same ESDIRK. In a further study with the ESDIRK from ARK4(3)6L[2]SA [255], Bijl and Carpenter [37, 95] combine a multigrid solution strategy followed by a Newton-Krylov method to achieve a 30 percent speed-up.

Variable-order implicit Runge-Kutta implementations are possible but are far less common than similar multistep implementations. Cash and Liem [102] and Cameron [92] do this with SDIRKs, Burrage et al. [54] use SIRKs, Butcher et al. [71] consider DESI methods, and Hairer and Wanner [194] use the Radau IIA family of methods. Hairer and Wanner state that the objective of such implementations is to minimize the local error per unit step. The required step-size is estimated for each candidate method and then the cost of this step-size is estimated. One then selects the method with the lowest cost to step-size ratio. As the optimal $\gamma$ changes for SDIRKs and ESDIRKs as order and the number of stages changes, an embedded approach like the ones Cash and Liem and Burrage et al. use seems less than ideal. With increased efficiency as the goal, a variable-order strategy would best be incorporated into the iteration control strategy. Possibly the reason why so few variable-order implicit Runge-Kutta codes exist is the difficulty in executing this well. Butcher [63] describes an order selection strategy for variable-order methods.

In certain contexts, implicit methods may be implemented to mimic IMEX methods. Brown and Hindmarsh [46] effectively consider solving (82) by treating the RHS as if all terms were to be integrated implicitly but constructing the Jacobian by using only the stiff part of the RHS.

### 2.16 Step-Size Control

Local integration error for Runge-Kutta methods is usually controlled by first creating a local error estimate via an embedded method. This error estimate is then fed to an error controller, that adjusts the time step in order to maintain some user-specified relative error tolerance $\epsilon$. Ideally, this step-size estimate is used as part of an iteration control strategy. Although this review focuses on largely cost-free embedded methods where $p = \hat{p} + 1$, some papers use $p = \hat{p} - 1$ [7, 16, 17, 209]. Others [17, 382] include the possibility of using extra function evaluations to increase the order of their embedded method. The reliability of this error estimate may be facilitated by using L-stable main and embedded methods [179] as well as higher stage-order methods. Given noisy data, Krogh [274] describes a step-size selection strategy that uses a least-squares fit of past data. Ilie et al. [223] approach optimal time step locations during the integration of both initial and boundary value
problems by considering a grid deformation map, \( t = \Theta(\tau) \). In this formulation, each time step location in the deformed space is equispaced relative to adjacent step locations but variable in the undeformed space. The goal is to determine an optimal grid deformation map, through a variational approach, which either 1) minimizes the work required to maintain a specified local error tolerance or 2) minimizes the local error given a fixed amount of work (steps).

Following Söderlind [402–405], we consider several error-control strategies of the form

\[
(\Delta t)^{[n+1]} = \kappa(\Delta t)^{[n]} \left\{ \frac{\epsilon}{||\delta^{[n+1]}||} \right\}^a \left\{ \frac{||\delta^{[n]}||}{\epsilon} \right\}^\beta \left\{ \frac{||\delta^{[n-1]}||}{\epsilon} \right\}^\gamma \times \\
\times \left\{ \frac{(\Delta t)^{[n]}}{(\Delta t)^{[n-1]}} \right\}^a \left\{ \frac{(\Delta t)^{[n-1]}}{(\Delta t)^{[n-2]}} \right\}^b
\]

for \( p(\hat{\tau}) \)-pairs (\( \hat{\tau} = p - 1 \)). In (93), \( \kappa \approx 0.95 \) is the safety factor, \( (\Delta t)^{[n-1]} = \Delta t^{[n-1]} - \Delta t^{[n-1]} \) is the step size, and \( \delta^{[n+1]} \) is the vector of most recent local error estimates of the integration from \( \Delta t^{[n-1]} \) to \( \Delta t^{[n]} \) associated with the computation of \( U^{[n+1]} \). To analyse this controller, Söderlind defines two polynomials \( P(q) = aq^2 - bq + \gamma \) and \( Q(q) = q^2 - aq - b \) where it is required that \( \text{deg} \ P(q) \leq \text{deg} \ Q(q) \). If common factors are contained in both \( P(q) \) and \( Q(q) \) then they are removed. From this, the characteristic polynomial or characteristic equation of the controller, having no common factors between \( P(q) \) and \( Q(q) \), is

\[
\Phi(q) = (q - 1)Q(q) + \hat{q}P(q) = q^3 + (\hat{q}a - 1)q^2 + (-\hat{q}\beta + a - b)q + (\hat{q}\gamma + b).
\]

For stability, all roots of \( \Phi \) must have a magnitude of less than or equal to one. Controllers based on this structure possess three distinct orders: the order of the dynamics, \( p_D \); the order of adaptivity, \( p_A \); and the order of the step-size filter, \( p_F \). Söderlind names particular controllers as \( H_{PDPAF} \). The degree of the polynomial \( \Phi \) is the order of the dynamics, while the order of adaptivity determines the rate at which the local error is adapted to the error tolerance. The filter order is related to averaging of the potentially noisy data and, when greater than zero, the controller acts as a low-pass step-size filter by removing high frequency data. As small roots to the characteristic equation are sometimes desirable, one may enforce \( (\hat{q}a - 1) = 0 \), \( (-\hat{q}\beta + a - b) = 0 \) and \( (\hat{q}\gamma + b) = 0 \) to obtain \( \Phi = (q - 0)^3 \). This implies \( \alpha = (a + 1)/\hat{q} \), \( \beta = (a - b)/\hat{q} \) and \( \gamma = -b/\hat{q} \) and that all roots vanish. Controllers satisfying these equations are denoted \( H_{0PDPAF} \). Methods possessing third-order dynamics are of particular interest. Within this class, one may design \( H312 \) \([\beta = -2\alpha, \gamma = \alpha] \), \( H321 \) \([b = (1 - a), \gamma = -(\alpha + \beta)] \) and \( H330 \) \([a = 2, b = -1] \) controllers, but \( H_{PDPAF} \) controllers having \( p_D < p_A + p_F \) do not appear to exist. Table 8 lists coefficients for various controllers. A more exhaustive list is given by Söderlind [404] as well as further design details. As the characteristic polynomial root positions play an important role in controller behavior, we note the \( p_D \)-roots of several controllers: \( H211 \) \( \{1/2 \pm I/2\} \), \( PPID \) \( \{0.4838 \pm 0.3275I, 0.7325\} \), \( H321 \).
Table 8. Error controller coefficients and the order of the dynamics, adaptivity, and filtering.

<table>
<thead>
<tr>
<th>Controller</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\gamma$</th>
<th>$a$</th>
<th>$b$</th>
<th>$p_D$</th>
<th>$p_A$</th>
<th>$p_F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I = H_0110$</td>
<td>$\frac{1}{p^{2}+1}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>H211</td>
<td>$\frac{1}{p}$</td>
<td>$\frac{3}{p^2}$</td>
<td>0</td>
<td>$\frac{7}{8}$</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>H321</td>
<td>$\frac{1}{p}$</td>
<td>$\frac{3}{p^2}$</td>
<td>0</td>
<td>$\frac{7}{8}$</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>PC = $H_0220$</td>
<td>$\frac{1}{p}$</td>
<td>$\frac{3}{p^2}$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>PID</td>
<td>$\frac{1}{p}$</td>
<td>$\frac{3}{p^2}$</td>
<td>$\frac{1}{p^3}$</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>H312</td>
<td>$\frac{1}{p}$</td>
<td>$\frac{3}{p^2}$</td>
<td>$\frac{1}{p^3}$</td>
<td>$\frac{7}{8}$</td>
<td>$\frac{1}{8}$</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>H312 general</td>
<td>$\frac{1}{p}$</td>
<td>$\frac{3}{p^2}$</td>
<td>$\frac{1}{p^3}$</td>
<td>$\frac{7}{8}$</td>
<td>$\frac{1}{8}$</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>PPID</td>
<td>$\frac{1}{p}$</td>
<td>$\frac{3}{p^2}$</td>
<td>$\frac{1}{p^3}$</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>H321</td>
<td>$\frac{1}{p}$</td>
<td>$\frac{3}{p^2}$</td>
<td>$\frac{1}{p^3}$</td>
<td>$\frac{7}{8}$</td>
<td>$\frac{1}{8}$</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>H331</td>
<td>$\frac{1}{p}$</td>
<td>$\frac{3}{p^2}$</td>
<td>$\frac{1}{p^3}$</td>
<td>$\frac{7}{8}$</td>
<td>$\frac{1}{8}$</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>H330 general</td>
<td>$\frac{1}{p}$</td>
<td>$\frac{3}{p^2}$</td>
<td>$\frac{1}{p^3}$</td>
<td>2</td>
<td>$-1$</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

$\{1/3, 1/2, 2/3\}$, PID $\{-0.1784, 1/2, 0.6228\}$ and H312 $\{0, 0, 1/2\}$. Alternatively, one may write an $H321_{\text{general}}$ controller having characteristic equation roots given by $q_1$, $q_2$ and $q_3$, where all roots have magnitudes preferably less than unity. Either three roots are real or one is real and the others form a complex conjugate pair

$$
\begin{align*}
\alpha &= (5 - 3q_1 - 3q_2 - 3q_3 + q_1q_2 + q_1q_3 + q_2q_3 + q_1q_2q_3) / (4\hat{p}) \\
\beta &= 2(-1 + q_1)(-1 + q_2)(-1 + q_3) / (4\hat{p}) \\
\gamma &= - (\alpha + \beta) \\
\alpha &= (1 + q_1)(1 + q_2)(1 + q_3) / 4 \\
\beta &= 1 - \alpha.
\end{align*}
$$

The step-size transfer map, $H_{\hat{\phi}[321]}(q)$, and the error transfer map, $R_{\hat{\phi}[321]}(q)$, are given by [404]

$$
\begin{align*}
-\hat{p}H_{\hat{\phi}[321]}(q) &= \frac{\hat{p}P(q)}{(q-1)Q(q) + \hat{p}P(q)} \\
&= \frac{(1 + q)(-3 + q_2 + q_3 + q_2q_3 + q_1(1 + q_2 + q_3 - 3q_2q_3))}{4(q - q_1)(q - q_2)(q - q_3)} \\
&+ \frac{(1 + q)q(5 + q_2(-3 + q_3) - 3q_3 + q_1(-3 + q_2 + q_3 + q_2q_3))}{4(q - q_1)(q - q_2)(q - q_3)} \\
R_{\hat{\phi}[321]}(q) &= \frac{(q-1)Q(q)}{(q-1)Q(q) + \hat{p}P(q)} \\
&= \frac{(q-1)^2(3 + 4q - q_1 - q_2 - q_3 - q_1q_2 - q_1q_3 - q_2q_3 - q_1q_2q_3) - 4(q - q_1)(q - q_2)(q - q_3)}{4(q - q_1)(q - q_2)(q - q_3)}.
\end{align*}
$$

48
One may also consider the control map of the controller \([404]\)

\[
C(q) = \frac{P(q)}{(q - 1)Q(q)}.
\]  

(98)

As with the H321\(_{\text{general}}\) controller, one may also write an H312\(_{\text{general}}\) as

\[
\alpha = -(q_1 - 1)(q_2 - 1)(q_3 - 1)/(4\hat{p})
\]

\[
\beta = -2\alpha
\]

\[
\gamma = \alpha
\]

\[
a = (3(-1 + q_3) + q_2(3 + q_3) + q_1(3 + q_2 + q_3 - q_2q_3))/4
\]

\[
b = (-1 + q_2 + q_3 - q_2q_3 - q_1(-1 + q_2 + q_3 + 3q_2q_3))/4
\]

and

\[
-\hat{p}H_{\hat{\phi}[312]}(q) = \frac{-(1 + q)^2(q_1 - 1)(q_2 - 1)(q_3 - 1)}{4(q - q_1)(q - q_2)(q - q_3)},
\]

(100)

\[
R_{\hat{\phi}[312]}(q) = \frac{(q - 1)(4q + (-1 + q_2)(-1 + q_3) + q_1(-1 + q_2 + q_3 + 3q_2q_3)}{4(q - q_1)(q - q_2)(q - q_3)}
\]

\[
+ \frac{q(q - 1)(3 + q_1(-3 + q_2(-1 + q_3) - q_3) - 3q_3 - q_2(3 + q_3))}{4(q - q_1)(q - q_2)(q - q_3)}.
\]

(101)

In the event that one needs higher-order adaptivity or filtering, one may consider a more general controller than (93) by multiplying the RHS by

\[
\left\{ \left\lceil \frac{||\delta^{(n-2)}||}{\epsilon} \right\rceil \right\}^\delta \left\{ \left\lceil \frac{\epsilon}{||\delta^{(n-3)}||} \right\rceil \right\}^\epsilon \left\{ \left\lceil \frac{(\Delta t)^{(n-2)}}{\epsilon} \right\rceil \right\}^\epsilon \left\{ \left\lceil \frac{(\Delta t)^{(n-3)}}{\epsilon} \right\rceil \right\}^\epsilon \left\{ \left\lceil \frac{(\Delta t)^{(n-4)}}{\epsilon} \right\rceil \right\}^\epsilon.
\]

(102)

If this is done, one might best include the step-size ratio dependence of \(\alpha, \beta, \gamma, \delta\) and \(\epsilon\) that occurs with controllers using \(\gamma, \delta, \epsilon, \cdots\) otherwise the delivered controller behavior in cases where the step-size ratio deviates significantly from unity may be different from the design intent. In the event of order reduction, \(\hat{p}\) no longer reflects the actual order of the embedded method. Controller coefficients might best be considered so that the controller functions well when given either the formal or actual value of \(\hat{p}\). Order conditions for the adaptivity and filter are

\[
p_A = 2 \quad a = 1 - b - c - d
\]

\[
p_A = 3 \quad b = -1 - 2c - 3d,
\]

\[
p_A = 4 \quad c = 1 - 3d
\]

\[
p_A = 5 \quad d = -1
\]

\[
p_F = 1 \quad \alpha = -\beta - \gamma - \delta - \epsilon
\]

\[
p_F = 2 \quad \beta = -2\gamma - 3\delta - 4\epsilon
\]

\[
p_F = 3 \quad \gamma = -3\delta - 6\epsilon
\]

\[
p_F = 4 \quad \delta = -4\epsilon.
\]

(103)
Using these along with \( P(q) = \alpha q^4 - \beta q^3 + \gamma q^2 - \gamma q + \varepsilon \) and \( Q(q) = q^4 - \alpha q^3 - b q^2 - c q - d \) (with common factors between \( P(q) \) and \( Q(q) \) removed), one may construct the characteristic polynomial of the controller and generate a controller having desirable roots.

In the simplest case, the local error estimate is given by \( \delta^{[n+1]} = U^{[n+1]} - \hat{U}^{[n+1]} \). There are two ways in which this might be improved. Normalization of the errors from each equation makes particular sense when elements of the integration vector have different orders of magnitude. To avoid division by zero, one adds a small positive absolute error tolerance, \( \eta \), to the denominator \( 45, 206, 343, 378 \). Hence, the two equivalent expressions

\[
\delta^{[n+1]} = \frac{U^{[n+1]} - \hat{U}^{[n+1]}}{|U^{[n+1]}| + \frac{\eta}{\varepsilon}}, \quad \delta^{[n+1]} \varepsilon = \frac{U^{[n+1]} - \hat{U}^{[n+1]}}{|U^{[n+1]}| + \eta}.
\]

A scalar measure of \( \delta^{[n+1]} \) is often obtained by using either an \( L_\infty \) or an \( L_2 \) norm,

\[
||\delta^{[n+1]}||_\infty = \max_i \left|\delta_i^{[n+1]}\right|, \quad ||\delta^{[n+1]}||_2 = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\delta_i^{[n+1]}\right)^2},
\]

where \( N \) is the number of equations being integrated. When stiff equations are being integrated, some error estimators are prone to gross overestimation of the local error. This overestimation is related to the stability function of the embedded method as \( z \to -\infty \). When \( \hat{R}(-\infty) \) is nonzero for stiff scaled eigenvalues, Shampine and Baca \( 209, 379 \) recommend multiplying the traditional estimate by inverse powers of the (Newton) iteration matrix. Defining \( m \) by

\[
\left( \frac{R(z) - \hat{R}(z)}{\exp(z) - R(z)} \right) = O(z^m), \quad \Re(z) < 0, \quad z \to -\infty
\]

one computes the error estimate by

\[
\delta^{[n+1]} = N^{-m} \left( \frac{U^{[n+1]} - \hat{U}^{[n+1]}}{|U^{[n+1]}| + \frac{\eta}{\varepsilon}} \right),
\]

where \( N^{-1} = U^{-1} L^{-1} P^{-1} \) from (84). If multiple equations are being solved, an error controller for each equation may be required, including a tailored value of \( \eta \). The motivation for doing this is that each value of \( \delta^{[n+1]} \), \( \delta^{[n]} \) and \( \delta^{[n-1]} \) in (93) might otherwise be a value from a different equation, resulting in an erratic error history. Generally, one also places limits on how quickly the step-size is allowed to vary. Increases or decreases of a factor of less than five in any given step are prudent \( 378 \). Increases greater than a factor of two may be at odds with robust stage-value predictors, error control and iteration control. Hindmarsh \( 206 \) allows LSODE to increase the step-size up to a factor of 10 000 on the first step but limits this increase to 10 afterwards. VODE limits reductions in step-size to \( 1/10 \). In situations where the future step-size selected by the controller is nearly equal to the
current step-size, some codes make no change to the step-size until some threshold of, often, +20 percent is exceeded. Söderlind [403] strongly discourages introducing such a dead-zone because, when the controller is eventually invoked, the strong actions may cause error overshoots and step rejections.

Although never applied to implicit Runge-Kutta methods, the $p(p-2)$-pair approach of Tsitouras and Papakostas [431] is appealing. They consider a modified I-controller,

$$(\Delta t)^{[n+1]}_1 = \kappa(\Delta t)^{[n]} \left\{ \frac{\epsilon}{f_2(\Delta t)||\delta^{[n+1]}||} \right\}^{\frac{1}{p-1}},$$ (108)

where $f_2 \approx 10$ is experimentally optimized. More sophisticated incarnations of this strategy are certainly possible and would allow one to design main scheme methods that were less compromised by demands of the embedded method. An alternative approach is to use two-step embedded methods [176] to enable higher order and better stability. Beyond control of local error, one might wish to control global error. The literature contains limited guidance on global error estimation of implicit Runge-Kutta methods applied to stiff problems [124, 144, 307, 361, 363, 375, 417].

Because machine precision is finite, the preceding approaches must be applied respecting this limit. If $|U^{[n+1]}|$ becomes very small, then an absolute error tolerance is added to the error estimate. Further, there are limits on how strict an error tolerance may be set [370]. Related to this limit is the selection of an initial step size [45, 165, 167, 193, 206, 313, 343, 445]. If the chosen step-size is too large, then the error will be excessive or the iteration may not converge. If the chosen step-size is too small, then an inefficiency is added or the result is swamped by rounding error. Rounding error may be minimized by using compensated summations [69, 189, 199, 346].

Traditionally, the embedded method is distinct from the main method because there are two sets of scheme weights, $b_i$ and $\hat{b}_i$. A disadvantage to this approach while solving stiff ODEs is that the error estimate has the stability function of an explicit method. One could follow the strategy of Shampine and Baca, (107), and multiply the resulting error estimate times the appropriate inverse power of the iteration matrix. However, to improve on this process, Al-Rabeh [17] and others [8, 89, 281, 450, 451] consider what one may call internal error-control as opposed to the traditional external error-control. Internal error control is often performed within the context of DAEs and is accomplished by imposing all embedded method constraints on one of the stages, often the penultimate stage, to avoid fixing $\gamma$ and to permit higher order methods. The resulting error estimate is likely relatively reliable if the internal stability of the penultimate stage is controlled. When combined with the stiffly-accurate assumption, unfortunately, the method becomes confluent. A potential drawback of this approach is when significant order reduction occurs, each method reduces to the same order because they share the same stage order. As the stiffness primarily affects the order of the stiff modes [157], the two solutions will probably not collide. It is an interesting question whether there is any benefit to generating both an internal and an external error estimate. Several proposed
methods in this paper offer both in the event the construct is helpful, possibly using the difference as a measure of stiffness or order reduction.

2.17 Iteration Control

Traditionally for DIRK-type methods, the step-size is selected by using an error estimate in collaboration with an error controller such as a P(redictive)C(ontrol)-controller. However, one must also consider how the step-size selection affects the convergence rate of the iterative solver. Iteration control attempts to coordinate the two requirements of integration accuracy and iteration efficiency to yield an optimal time step choice throughout the computation. Gustafsson and Söderlind [180] propose a comprehensive approach to iteration control when using direct solvers. Salane [350] addresses some of these issues, but less comprehensively. Iteration control strategies used in the LSODE [343], PVODE [206], and DASSL [45] software packages may also be consulted. Unfortunately, little guidance exists in the literature on iteration control associated with indirect solvers [206] where, arguably, most real-world CPU-hours are consumed.

In the case of a modified Newton iteration, Gustafsson and Söderlind contrast the value of the iteration matrix being used to that which would exist when using a current step-size and fresh Jacobian. As the iteration matrix becomes more dated, they decompose the expected convergence rate of the iteration into that caused by changes in step-size from the value used to construct the iteration matrix and that due to an increasingly inaccurate Jacobian. As one may observe the actual convergence rate, $\alpha$, and readily compute the convergence rate component due to step-size changes, $\alpha_{lu}$

$$\alpha_k = \frac{|d^{(i)}_k|}{|d^{(i)}_{k-1}|}, \quad \alpha = \max_k \alpha_k, \quad \alpha_{lu} = \frac{(\Delta t) - (\Delta t)_{lu}}{(\Delta t)_{lu}} \tag{109}$$

via a linearization, then $\alpha_{jac} = \alpha - \alpha_{lu}$. The subscript $lu$ denotes the value used to construct the most recent iteration matrix. For quantities related to the Jacobian, the subscript $jac$ is used. Obviously, reliable estimates for the scaled displacement are essential to proper functioning of the iteration control. One then contrasts these estimates to target values for these quantities, $\alpha_{ref}$, $\alpha_{LU}$ and $\alpha_{JAC}$. The goal now is to keep $\alpha \approx \alpha_{ref}$, $\alpha_{lu} \approx \alpha_{LU}$ and $\alpha_{jac} \approx \alpha_{JAC}$. If $\alpha_{jac} > \alpha_{JAC}$, then the convergence is likely to either fail or be inefficient due to an old Jacobian. Note that with this procedure, one need not place a priori fixed age limits on the Jacobian or iteration matrices as is commonly done. Gustafsson and Söderlind recommend starting with the values $\alpha_{ref} = 0.2$, $\alpha_{LU} = 0.2$ and $\alpha_{JAC} = 0.2$. Söderlind [403] recommends $\alpha_{ref} \leq 0.37$ on lower precision calculations and reduced values as the precision requirement becomes more strict. With these values, they provide pseudocode for selecting the time step in conjunction with the error controller, criteria for updating the Jacobian, and when to update the iteration matrix, for both converging and diverging iterations. Several improvements to this procedure have been devised by Olsson and Söderlind [323] and de Swart [423]. An abbreviated version of the latter appears in figure 2. It is invoked once the step has either been successfully completed
if (.not. excessive iterate growth) then 
\((\Delta t)[n+1] = \frac{(\Delta t)[n]\alpha_{\text{ref}}}{\max\{\alpha, f_{\text{max}}^{\text{ref}}\}}\)

if (iteration converged) then 
\(\text{compute } (\Delta t)[n+1]_{\text{EC}}\)
if ((fresh Jacobian) .and. (\(\alpha > \alpha_{\text{ref}}\)) then 
\((\Delta t)[n+1] = \min\{ f_{\text{max}}(\Delta t)[n], \max\{ f_{\text{min}}(\Delta t)[n], \min\{ (\Delta t)[n+1]_{\text{EC}}, (\Delta t)[n+1]_{\alpha}\}\}\}\)
else 
\((\Delta t)[n+1] = \min\{ f_{\text{max}}(\Delta t)[n], \max\{ f_{\text{min}}(\Delta t)[n], (\Delta t)[n+1]_{\alpha}\}\}\)
endif 
if ((.not. exact) .and. (\(\alpha_{\text{Jac}} > \alpha_{\text{JAC}}\)) then 
if (fresh Jacobian) then 
\((\Delta t)[n] = \frac{(\Delta t)[n]}{f_{\text{rigid}}}\)
else 
update Jacobian \(\text{update Jacobian}\)
endif 
else 
elseif (excessive iterate growth) then 
\((\Delta t)[n] = \frac{(\Delta t)[n]}{f_{\text{rigid}}}\)
else (iteration diverged) then 
\((\Delta t)[n] = \min\{ f_{\text{max}}(\Delta t)[n], \max\{ f_{\text{min}}(\Delta t)[n], (\Delta t)[n+1]_{\alpha}\}\}\)
if (.not. fresh Jacobian) then 
update Jacobian \(\text{update Jacobian}\)
endif 
elseif (slow convergence) then 
if (fresh Jacobian) then 
if (\(\alpha > \zeta_{\alpha_{\text{ref}}}\)) then 
\((\Delta t)[n] = \min\{ f_{\text{max}}(\Delta t)[n], \max\{ f_{\text{min}}(\Delta t)[n], (\Delta t)[n+1]_{\alpha}\}\}\)
else 
\((\Delta t)[n] = \frac{(\Delta t)[n]}{f_{\text{rigid}}}\)
endif 
else 
update Jacobian matrix 
\((\Delta t)[n] = (\Delta t)[n]\)
endif 
endif 
if ((updated Jacobian) .or. (\(\alpha_{\text{lu}} > \alpha_{\text{LU}}\)) then 
\text{update iteration matrix}\)
if (redostep) \((\Delta t)_{\text{lu}} = (\Delta t)[n]\)
if (.not. redostep) \((\Delta t)_{\text{lu}} = (\Delta t)[n+1]\)
endif

Figure 2. Pseudocode for iteration control based on the work of de Swart et al. [423].
or has failed and must be recomputed beginning at $t^n$. The terms $(\Delta t)_n^{n+1}$ and $(\Delta t)_EC^{n+1}$ are, respectively, the time steps predicted to return the convergence rate to its optimal value, $\alpha_{\text{ref}}$, and the error-control time step required to bring the integration error, $\delta^{n+1}$, to the relative error tolerance, $\epsilon$. Several comments are in order. First, de Swart recommends the values $\alpha_{\text{ref}} = 0.15$, $\alpha_{LU} = 0.2$, $\alpha_{JAC} = 0.1$, $f_{\text{min}} = 0.2$, $f_{\text{max}} = 2.0$, $f_{\text{rigid}} = 2.0$ and $\xi = 1.2$. These seven numbers serve to bound changes. Next, one must supplement this pseudocode with criteria by which convergence, divergence, slow convergence, or excessive iterate growth may be determined. The last three situations require the step to be recomputed. De Swart provides such criteria, and the first two have already been discussed here. Slow convergence describes a situation in which the convergence rate is too slow to satisfy the iteration tolerance within the remaining iterations or that the maximum number of iterations has been attained. This former criterion is given by de Swart and Hairer and Wanner [193] based on the displacement

$$\frac{\alpha_k^{k_{\text{max}} - k}}{1 - \alpha_k} ||d_k^{(i)}|| > c_\epsilon \epsilon,$$

where $k_{\text{max}}$ is the maximum allowed iterations and values for $c_\epsilon$ are given in (85). Excessive iterate growth occurs when the iterate has reached some large multiple of its original absolute value during iteration. Two obvious generalizations may be of value. The first is that in cases of $a_{ii}$ varying, one might redefine $\alpha_{LU}$ with

$$\alpha_{LU} = \left| \frac{a_{ii}(\Delta t)_{n} - (a_{ii})_{LU}(\Delta t)_{LU}}{(a_{ii})_{LU}(\Delta t)_{LU}} \right|.$$  

This condition may occur when using a DIRK, QSDIRK, E DIRK, QESDIRK or during a variable-order implementation. Secondly, if the procedure given by Krogh [206,275,462] in (90) is used, then $\alpha_{LU}$ might be raised substantially (but this would make it difficult to assess $\alpha_{JAC}$). A variable-order implementation would require that this strategy be expanded.

Limited experience with this procedure on Navier-Stokes calculations involving extremely stiff chemical mechanisms illuminated three potential shortcomings. An IMEX additive Runge-Kutta [255] has been used with a PID-controller on each equation of each spatial grid point, and the iteration control was performed essentially at each grid point. Step-sizes are chosen as the minimum $\Delta t$ called out by any grid point by the error controller. Any point could fail to converge and force all points to redo the failed step. The first problem is that passing the displacement test seemed to be an insufficiently robust measure of whether the integration is proceeding well and needed to be supplemented with increased values of the minimum iteration number. A second is that $\alpha_{JAC}$ is not aggressive enough in updating the Jacobian. As a consequence, Jacobian updates needed to be hardwired to occur sufficiently often – a distinctly undesirable situation. Lastly, values of $\alpha_k$ near but less than unity, which corresponded to values of the displacement much less than the iteration tolerance, might best be filtered out of the computation of $\alpha$.

Volcker et al. [439] apply similar iteration control strategies to integrations performed using ESDIRK methods towards reservoir simulations.
2.18 Stage-Value Predictors

Solution of the stage and step equations, (82), is invariably done using an iterative procedure. Because of this, the efficiency of this iterative procedure depends strongly on the quality of the starting guess. A good starting guess, \( U_{i,0}^{[n+1]} \), for stage \( i \) toward the computation of the \( U \)-vector at step \( n+1 \) minimizes both the chances of a failed iteration and the effort required for convergence. A variety of possible stage-value, or starting, predictors may be used with DIRK-type methods ranging from simple approaches to rather sophisticated ones. Strategies often depend on whether the \( A \)-matrix is invertible. For EDIRK, ESDIRK, and QESDIRK methods, Olsson and Söderlind\cite{201,205,323} advise against stage-value predictors that use information from the first stage.

Implicit in each of these strategies is that the underlying method is not modified in any way to facilitate stage-value prediction. However, abscissae position may also be manipulated to facilitate convergence. For example, if the method was multiply confluent, iteration would be rapid on the second repeated abscissa. As this may compromise the quality of the method, one may prefer to optimize the method and accompany it with a good stage-value predictor.

Stage-value predictors are considered for the stages \( U_i^{[n]} \) that give rise to the computation of \( U_i^{[n+1]} \). To do this, past information will be used that possibly includes \( U_i^{[n-3]} \), \( U_i^{[n-2]} \), \( U_i^{[n-1]} \), \( U_i^{[n]} \), and \( U_i^{[n+1]} \). Variable step-sizes are given by \( t[n] - t[n-1] = \Delta t \) and

\[
r_1 = \frac{t[n+1] - t[n]}{t[n] - t[n-1]} \tag{112}
\]

for single-step predictors, \( t[n-1] - t[n-2] = \Delta t \) and

\[
r_1 = \frac{t[n] - t[n-1]}{t[n-1] - t[n-2]}, \quad r_2 = \frac{t[n+1] - t[n]}{t[n-1] - t[n-2]} \tag{113}
\]

for two-step predictors and \( t[n-2] - t[n-3] = \Delta t \) and

\[
r_1 = \frac{t[n-1] - t[n-2]}{t[n-2] - t[n-3]}, \quad r_2 = \frac{t[n] - t[n-1]}{t[n-2] - t[n-3]}, \quad r_3 = \frac{t[n+1] - t[n]}{t[n-2] - t[n-3]} \tag{114}
\]

for three-step predictors. The simplest stage predictors are given by \cite{188}

\[
U_{i,0}^{[n]} = U_i^{[n]}, \quad U_{i,0}^{[n]} = U_i^{[n]} + (\Delta t)c_i F_i^{[n]}, \tag{115}
\]

The first of these is referred to as the trivial guess or the trivial predictor. One could modify it to \( U_{i,0}^{[n]} = U_i^{[n-1]} \). Less primitive than the two proceeding methods, dense output may be used to extrapolate function values from the previous step to generate starting guesses for the stage values \cite{193,255,313,351}

\[
U(t[n] + \theta \Delta t) = U_i^{[n]} + (\Delta t) \sum_{i=1}^{s} b_i(\theta) F_i^{[n-1]}, \tag{116}
\]
where θ is selected to be one of the scheme abscissae, \( c_i \). Both stability and accuracy degrade quickly as θ is increased; consequently, a limit of \( \theta \leq 3 \) is often imposed [173, 175, 352]. This step change restriction is not inconsistent with prudent error and iteration control strategies [423]. These methods, like those that follow, cannot be used on the first step where the trivial guess must be used. Some authors advocate hermite interpolation for stage-value predictors [7, 281].

More sophisticated approaches to stage-value prediction are now considered. Laburta [284] considers stage-value predictors that require as many as two effective extra function evaluations (\( F_{n-1}^{[n-1]} \) and \( F_{s+2}^{[n-1]} \)) may be used to increase the order of the predictor. Hairer et al. [188] generalize this procedure to appear as

\[
U_{i,0}^{[n]} = U^{[n-1]} + (\Delta t) \sum_{j=1}^{s} \beta_{ij} F_j^{[n-1]} \\
+ (\Delta t) \left( \beta_{i,0} F_{n-1}^{[n-1]} + \beta_{i,s+1} F_s^{[n]} + \beta_{i,s+2} F_{s+2}^{[n-1]} \right)
\]

(117)

\[
U_{s+2}^{[n-1]} = U^{[n-1]} + (\Delta t) \sum_{j=1}^{s} \mu_{ij} F_j^{[n-1]} \\
+ (\Delta t) \left( \mu_0 F_{n-1}^{[n-1]} + \mu_{s+1} F_s^{[n]} \right)
\]

(118)

where extra function evaluations (\( F_{n-1}^{[n-1]} \), \( F_n^{[n]} \) and \( F_{s+2}^{[n-1]} \)) may be used to increase the order of the predictor. Hairer et al. [188] generalize this procedure to appear as

\[
U_{i,0}^{[n]} = U^{[n-1]} + (\Delta t) \sum_{j=1}^{s} \beta_{ij} F_j^{[n-1]} + (\Delta t) \sum_{j=1}^{m} \nu_{ij} F_{s+j}^{[n-1]} \\
U_{s+j}^{[n-1]} = U^{[n-1]} + (\Delta t) \sum_{k=1}^{s} \mu_{jk} A F_k^{[n-1]} + (\Delta t) \sum_{k=1}^{j-1} \mu_{jk} B F_k^{[n-1]}
\]

(119)

(120)

where \( \mu_{ij} = (\mu_{ij}^A, \mu_{ij}^B) \) and \( \xi_i = \sum_{j=1}^{s} \mu_{ij}^A + \sum_{j=1}^{i-1} \mu_{ij}^B \). It is useful to define \( c = (e + rc) \) and \( B = e \otimes b^T \). Matrices \( \beta_{ij} \) and \( \mu_{ij}^A \) are of dimension \( s \times s \), \( \nu_{ij} \) is of dimension \( s \times m \) and \( \mu_{ij}^B \) is of dimension \( s \times (m-1) \). It may be seen that Laburta’s method includes up to three additional function evaluations (\( m \leq 3 \)). Effectively, however, it has two additional function evaluations as \( F^{[n]} \) from the previous step may be used for \( F_{n-1}^{[n-1]} \) in the current step. Order conditions for these methods, \( t_1^{(p)} \), up to order \( p = 4 \), may be constructed with six of the eight order conditions already given by Hairer et al. [188] for step-size ratios of unity. For bushy trees, \( t_1^{(p)} \), \( \kappa = 1, 2, 3, 4, \)

\[
\sum_{j=1}^{s} \beta_{ij} c_{j}^{\kappa-1} + \sum_{j=1}^{m} \nu_{ij} c_{j}^{\kappa-1} = \sum_{j=1}^{s} B_{ij} c_{j}^{\kappa-1} + r \sum_{j=1}^{s} a_{ij} c_{j}^{\kappa-1}
\]

(121)
For trees, \( t_2^{(3)} \) and \( t_3^{(4)} \) (\( \kappa = 2, 3 \))

\[
\sum_{i,j,k=1}^s \beta_{ijk} a_{jk} c_{k}^{\kappa-1} + \sum_{j=1}^m \sum_{k=1}^s \nu_{i,j,k} A^{A} c_{k}^{\kappa-1} + \sum_{j=1}^m \sum_{k=1}^s \nu_{i,j,k} B^{B} c_{k}^{\kappa-1} = \\
\sum_{j,k=1}^s B_{ij} a_{jk} c_{k}^{\kappa-1} + r \sum_{j,k=1}^s a_{ij} \left( B_{jk} c_{k}^{\kappa-1} + r a_{jk} c_{k}^{\kappa-1} \right) \tag{122}
\]

for \( t_2^{(4)} \),

\[
\sum_{i,j,k=1}^s \beta_{ijk} c_{jk} a_{jk} c_{k} + \sum_{j=1}^m \sum_{k=1}^s \nu_{ij,j,k} A^{A} c_{k} + \sum_{j=1}^m \sum_{k=1}^s \nu_{ij,j,k} B^{B} c_{k} = \\
\sum_{j,k=1}^s B_{ij} c_{jk} a_{jk} c_{k} + r \sum_{j,k=1}^s a_{ij} \left( B_{jk} c_{k} + r a_{jk} c_{k} \right) \tag{123}
\]

and for \( t_4^{(4)} \),

\[
\sum_{i,j,k,l=1}^s \beta_{ijk} a_{jk} c_{k} c_{l} + \sum_{j=1}^m \sum_{k=1}^s \nu_{ij,j,k} A^{A} c_{l} + \sum_{j=1}^m \sum_{k=1}^s \nu_{ij,j,k} B^{B} c_{l} + \\
\sum_{j=1}^m \sum_{k,l=1}^s \nu_{ij,j,k} B^{B} c_{l} = r \sum_{j,k,l=1}^s a_{ij} \left( B_{jk} a_{kl} c_{l} + r a_{jk} \left( B_{kl} c_{l} + r a_{kl} c_{l} \right) \right) \\
+ \sum_{j,k,l=1}^s B_{ij} a_{jk} a_{kl} c_{l}. \tag{124}
\]

These conditions can be simplified by using row simplifying assumptions, \( C(\eta) \). No assessment of predictor stability is presented.

As these stage-value predictors require extra function evaluations, one might seek different strategies that require no extra function evaluations. Many efforts towards effective and cost-free stage-value predictors are based on considering a multistep, extended Runge-Kutta coefficient array. One- \([201]\), two- \([78,79,174]\), and three-step stage-value predictors may be considered in the form

\[
U_{i,0}^{[n]} = \eta_i U_{i-1}^{[n]} + \sum_{j=1}^s B_{ij} U_{j}^{[n]} \tag{125}
\]

\[
U_{i,0}^{[n]} = \eta_i U_{i-1}^{[n-2]} + \sum_{j=1}^s B_{ij} U_{j}^{[n]} + \sum_{j=1}^s C_{ij} U_{j}^{[n-1]} \tag{126}
\]

\[
U_{i,0}^{[n]} = \eta_i U_{i-1}^{[n-3]} + \sum_{j=1}^s B_{ij} U_{j}^{[n]} + \sum_{j=1}^s C_{ij} U_{j}^{[n-1]} + \sum_{j=1}^s D_{ij} U_{j}^{[n-2]}, \tag{127}
\]

where matrices \( B_{ij} \), \( C_{ij} \), and \( D_{ij} \) are of dimension \( s \times s \) and \( \eta_i \) is a vector of length \( s \).

As written, this formulation requires the invertibility of the \( A \)-matrix but is easily
Butcher arrays characterized by \( \tilde{\Phi} \) outperform lower order ones. Pinto et al. [174,175] conclude that at stringent tolerances, predictor order matters where the step-size ratio, \( r = (\Delta t)^{n+1}/(\Delta t)^{n} \), is either \( r < 0.1, 0.1 \leq r \leq 2.0, \) or \( 2.0 \leq r \). As \( r \) increases, order is reduced and inverse order is increased. González-Pinto et al. [174,175] conclude that at stringent tolerances, predictor order matters but at more lax tolerances, stability properties matter more. Both González-Pinto et al. [174] and Calvo and Portillo [87] find that higher order starters do not always outperform lower order ones.

General one-, two- and three-step predictors are represented by using extended Butcher arrays characterized by \( \tilde{A}, \tilde{b} \) and \( \tilde{c} \) as

\[
\begin{array}{c|c|c|c|c|c}
\hline
& A & O & O & O & O \\
\hline
\tilde{c} & \tilde{A} & \tilde{b} & \tilde{r}_{1}\tilde{A} & \tilde{r}_{2}\tilde{A} & \tilde{r}_{3}\tilde{A} \\
\hline
b^T & r_{1}b^T & r_{2}b^T & r_{3}b^T \\
\hline
\end{array}
\]

associated with (125), (126) and (127), where \( \tilde{B} = e \otimes b^T, \tilde{e}^T = \{e, e, \cdots, e\}, \tilde{A}e = \tilde{c}, O \) is a square matrix of zeros, and \( B e = e \) for methods that are at least first-order accurate. Note that the components of \( \tilde{c} \) are given by \( c, \tilde{c} = e + r_{1}c, \) \( \tilde{c} = e(1 + r_{1}) + r_{2}c \) and \( c = e(1 + r_{1} + r_{2}) + r_{3}c \)

\[
\tilde{c} = \{c, c\}^T, \quad \tilde{\tilde{c}} = \{c, c, c\}^T, \quad \tilde{c} = \{c, c, c, c\}^T. \tag{129}
\]

Directly related to these, \( \text{diag}(c) = C, \text{diag}(c) = \tilde{C} \) and \( \text{diag}(c) = \tilde{C} \). We also define \( \tilde{\beta} \), the last row of \( \tilde{A} \), for one-, two- and three-step predictors as

\[
\tilde{\beta} = \{B, r_{1}A\}^T, \quad \tilde{\tilde{\beta}} = \{B, r_{1}B, r_{2}A\}^T, \quad \beta = \{B, r_{1}B, r_{2}B, r_{3}A\}^T. \tag{130}
\]

The order of accuracy of the stage-value predictor is given by

\[
||U_{i}^{[n]} - U_{i,0}^{[n]}|| = O((\Delta t)^{\tilde{p}}), \tag{131}
\]

where \( \tilde{p} \) is the largest integer for which the relation holds across all \( i \). One may now write the \( j \)-order conditions for the predictors at order \( p \leq 4 \), following the notation in (4), \( \tilde{\Phi}^{(p)}_{j} \), as

\[
\begin{align*}
\tilde{\Phi}_{1}^{(1)} & = \beta^T \tilde{\tilde{c}} \\
\tilde{\Phi}_{2}^{(1)} & = 2\beta^T \tilde{\tilde{c}} \\
\tilde{\Phi}_{1}^{(3)} & = 3\beta^T \tilde{c}^2, \quad \tilde{\Phi}_{2}^{(3)} = 6\beta^T \tilde{A}c \\
\tilde{\Phi}_{1}^{(4)} & = 4\beta^T \tilde{c}^3, \quad \tilde{\Phi}_{2}^{(4)} = 8\beta^T \tilde{C}A\tilde{c} \\
\tilde{\Phi}_{3}^{(4)} & = 12\beta^T \tilde{A}c^2, \quad \tilde{\Phi}_{4}^{(4)} = 24\beta^T \tilde{A}A\tilde{c}.
\end{align*}
\tag{132-136}
\]

58
Note that $\tilde{\Phi}_1^{(p)} = p\tilde{\beta}^T\tilde{c}^{p-1}$ and that the predictor order conditions are defined slightly different than those for the main method, e.g., (4).

These order conditions may be recast by using generalized simplifying assumptions,

\[
\begin{align*}
\tilde{B}(p) & : \quad \tilde{\beta}^T\tilde{c}^{k-1} - 1/k, \quad k = 1, 2, \ldots, p, \\
\tilde{C}(\eta) & : \quad \tilde{A}\tilde{c}^{k-1} - \tilde{c}^k/k, \quad k = 1, 2, \ldots, \eta,
\end{align*}
\]

where $C(\eta) + B(p)$ implies $\tilde{B}(\eta) + \tilde{C}(\eta)$ for $p \geq \eta \geq 1$. Related to $C(\eta)$ is $\tilde{q}^{(k)} = \tilde{A}\tilde{c}^{k-1} - \tilde{c}^k/k$, which allows the non-bushy-tree order conditions to be written

\[
\begin{align*}
\tilde{\Phi}_2^{(3)} & = 6\tilde{\beta}^T\tilde{q}^{(2)} + \tilde{\phi}^{(3,1)} \\
\tilde{\Phi}_2^{(4)} & = 8\tilde{\beta}^T\tilde{C}\tilde{q}^{(2)} + \tilde{\phi}^{(4,1)} \\
\tilde{\Phi}_3^{(4)} & = 12\tilde{\beta}^T\tilde{A}\tilde{q}^{(3)} + \tilde{\phi}^{(4,1)} \\
\tilde{\Phi}_4^{(4)} & = 24\tilde{\beta}^T\tilde{A}\tilde{q}^{(2)} + 12\tilde{\beta}^T\tilde{q}^{(3)} + \tilde{\phi}^{(4,1)}.
\end{align*}
\]

Alternatively, if simplifying assumptions $B(p)$ and $C(q)$ are invoked with $p \geq q \geq 1$, then one may write $\tilde{\Phi}_1^{(q)} = q\tilde{c}^q$ and

\[
\begin{align*}
\tilde{\Phi}_1^{(q+1)} & = (q + 1)\tilde{\beta}^T\tilde{c}^q \\
\tilde{\Phi}_1^{(q+2)} & = (q + 2)\tilde{\beta}^T\tilde{c}^{q+1} \\
\tilde{\Phi}_2^{(q+2)} & = (q + 1)(q + 2)\tilde{\beta}^T\tilde{A}\tilde{c}^q \\
\tilde{\Phi}_1^{(q+3)} & = (q + 3)\tilde{\beta}^T\tilde{c}^{q+2} \\
\tilde{\Phi}_2^{(q+3)} & = (q + 1)(q + 3)\tilde{\beta}^T\tilde{C}\tilde{A}\tilde{c}^{q+1} \\
\tilde{\Phi}_4^{(q+3)} & = (q + 1)(q + 2)(q + 3)\tilde{\beta}^T\tilde{A}\tilde{A}\tilde{c}^q.
\end{align*}
\]

To be more specific, the bushy tree order conditions $\tilde{\Phi}_1^{(p)} = p\tilde{\beta}^T\tilde{c}^{p-1}$, where $\tilde{c}^0 = \tilde{c}$ for the basic method, and the one-, two- and three-step predictors are

\[
\begin{align*}
\Phi_1^{(p)} & = p(A\tilde{c}^{p-1}) \\
\tilde{\Phi}_1^{(p)} & = p(B\tilde{c}^{p-1} + r_1A\tilde{c}^{p-1}) \\
\Phi_1^{(p)} & = p(B\tilde{c}^{p-1} + r_1B\tilde{c}^{p-1} + r_2A\tilde{c}^{p-1}) \\
\Phi_1^{(p)} & = p(B\tilde{c}^{p-1} + r_1B\tilde{c}^{p-1} + r_2B\tilde{c}^{p-1} + r_3A\tilde{c}^{p-1}).
\end{align*}
\]

Predictor order conditions based on the term $\tilde{\Phi} = (l + 1)(k + l + 2)\tilde{\beta}^T\tilde{C}^{k}\tilde{A}\tilde{c}^l$ are
given by

\[
\Phi = (l + 1)(k + l + 2)AC^kAc^l
\]

\[
\Phi = (l + 1)(k + l + 2)[BC^kAc^l + r_1AC^k(BC^l + r_1Ac^l)]
\]

\[
\bar{\Phi} = (l + 1)(k + l + 2)[BC^kAc^l + r_1BC^k(BC^l + r_1Ac^l)]
+ r_2AC^k(BC^l + r_1Bc^l + r_2Ac^l)
\]

\[
\bar{\Phi} = (l + 1)(k + l + 2)[BC^kAc^l + r_1BC^k(BC^l + r_1Ac^l)]
+ r_2BC^k(BC^l + r_1Bc^l + r_2Ac^l)
+ r_3AC^k(BC^l + r_1Bc^l + r_2Bc^l + r_3Ac^l)].
\]

These equations provide expressions for \(\Phi_2^{(3)}\), \(\Phi_2^{(4)}\) and \(\Phi_3^{(q+3)}\) or, alternatively, \(\Phi_2^{(q+2)}\), \(\Phi_2^{(q+3)}\) and \(\Phi_3^{(q+3)}\). If row simplifying assumptions are invoked, \(C(l + 1)\), these expressions are simplified to

\[
(l + 1)(k + l + 2)\tilde{\beta}^T\tilde{C}^k\tilde{A}\tilde{c}^l = (l + 1)(k + l + 2)\tilde{\beta}^T\tilde{C}^k\tilde{q}^{(l+1)} + \tilde{\Phi}_1^{(k+l+2)}.
\]

One can also write the expressions for \(\Phi = (l + 1)(l + 2)(k + l + 3)\tilde{\beta}^T\tilde{C}^k\tilde{A}\tilde{c}^l\) as

\[
\Phi = (l + 1)(l + 2)(k + l + 3)AC^kAAc^l
\]

\[
\Phi = (l + 1)(l + 2)(k + l + 3)BC^k[AAc^l]
+ r_1AC^k[(BA + r_1AB)c^l + r_1^2AAc^l]
\]

\[
\bar{\Phi} = (l + 1)(l + 2)(k + l + 3)BC^k[AAc^l]
+ r_1BC^k[(BA + r_1AB)c^l + r_1^2AAc^l]
+ r_2AC^k[(BA + r_1BB + r_2AB)c^l + (r_1^2BA + r_1r_2AB)c^l + r_2^2AAc^l]
\]

\[
\Phi = (l + 1)(l + 2)(k + l + 3)BC^k[AAc^l]
+ r_1BC^k[(BA + r_1AB)c^l + r_1^2AAc^l]
+ r_2BC^k[(BA + r_1BB + r_2AB)c^l + (r_1^2BA + r_1r_2AB)c^l + r_2^2AAc^l]
+ r_3AC^k[(BA + (r_1 + r_2)BB + r_3AB)c^l + (r_1^2BA + r_1r_2BB + r_1r_3AB)c^l]
+ r_3AC^k[(r_2^2BA + r_2r_3AB)c^l + r_2^2AAc^l].
\]

This provides an expression for \(\Phi_4^{(q)}\) when \(k = 0, l = 1\), or \(\Phi_4^{(q+3)}\) when \(k = 0, l = q\).

If row simplifying assumptions are invoked, one has

\[
(l + 1)(l + 2)(k + l + 3)\tilde{\beta}^T\tilde{C}^k\tilde{A}\tilde{c}^l = (l + 1)(l + 2)(k + l + 3)\tilde{\beta}^T\tilde{C}^k\tilde{q}^{(l+1)}
+ (l + 2)(k + l + 3)\tilde{\beta}^T\tilde{C}^k\tilde{q}^{(l+2)}
+ \tilde{\Phi}_1^{(k+l+3)}.
\]

Lastly, it is useful to write \(\tilde{\beta}^T\tilde{A} = [BA + r_1AB, r_1^2AA]^T\) for single step predictors and

\[
\tilde{\beta}^T\tilde{A} = \begin{bmatrix} BA + r_1BB + r_2AB \\ r_2^2BA + r_1r_2AB \\ r_2^2AA \end{bmatrix},
\]

\[
\tilde{\beta}^T\tilde{A} = \begin{bmatrix} BA + (r_1 + r_2)BB + r_3AB \\ r_1^2BA + r_1r_2BB + r_1r_3AB \\ r_2^2BA + r_2r_3AB \\ r_3^2AA \end{bmatrix}.
\]
for the two- and three-step predictors. The present expressions for \( \Phi \) may be related with the expressions for \( \Phi \) given by Calvo et al. [78] as, \( \Phi = \gamma A \Phi \), \( \Phi = \gamma (B, r_1 A) \Phi \) and \( \Phi = \gamma (B, r_1 B, r_2 A) \Phi \), where \( \gamma \) is the tree density. González-Pinto et al. [174] offer a variable-order strategy for two-step predictors, whereby a family of highly stable predictors are used in increasing orders so that the accuracy of the preceding method may be checked.

To find the predictor matrices, one solves

\[
Be = e - \eta, \quad B\Phi_j^{(p)} = \bar{\Phi}_j^{(p)} \tag{165}
\]

\[
Be + Ce = e - \eta, \quad C\Phi_j^{(p)} + B\bar{\Phi}_j^{(p)} = \bar{\Phi}_j^{(p)} \tag{166}
\]

\[
Be + Ce + De = e - \eta, \quad D\Phi_j^{(p)} + C\bar{\Phi}_j^{(p)} + B\bar{\Phi}_j^{(p)} = \bar{\Phi}_j^{(p)} \tag{167}
\]

for one-, two- and three-step predictors. Following Higuera and Roldán [201], one may consider predictors for five-stage SDIRK4 and define the square matrices

\[
V = \{e, \Phi_1^{(1)}, \Phi_1^{(2)}, \Phi_1^{(3)}, \Phi_1^{(3)} \} \tag{168}
\]

\[
V^* = \{(e - \eta), \Phi_1^{(1)}, \Phi_1^{(2)}, \Phi_1^{(3)}, \Phi_1^{(3)} \} \tag{169}
\]

\[
\mathfrak{V} = \{e, \bar{\Phi}_1^{(1)}, \bar{\Phi}_1^{(2)}, \bar{\Phi}_1^{(3)}, \bar{\Phi}_1^{(3)} \}, \quad \mathfrak{V}^* = \{(e - \eta), \bar{\Phi}_1^{(1)}, \bar{\Phi}_1^{(2)}, \bar{\Phi}_1^{(3)}, \bar{\Phi}_1^{(3)} \} \tag{170}
\]

\[
V = \{e, \bar{\Phi}_1^{(1)}, \bar{\Phi}_1^{(2)}, \bar{\Phi}_1^{(3)}, \bar{\Phi}_1^{(3)} \}, \quad V^* = \{(e - \eta), \bar{\Phi}_1^{(1)}, \bar{\Phi}_1^{(2)}, \bar{\Phi}_1^{(3)}, \bar{\Phi}_1^{(3)} \} \tag{171}
\]

where the dimension of the matrices is related to the number of nonconfluent stages. Now, one solves

\[
BV = V^*, \quad CV + BV = \mathfrak{V}^*, \quad DV + CV + B\mathfrak{V} = V^*. \tag{172}
\]

For cases where \( V \) is invertible (this requires that \( A^{-1} \) exists), one has

\[
B = V^*V^{-1}, \quad C = -BVVV^{-1} + \mathfrak{V}^*V^{-1}, \quad D = -CVV^{-1} - B\mathfrak{V}V^{-1} + V^*V^{-1}. \tag{173}
\]

Limiting discussion to only single-step predictors with invertible \( A \) matrices (SDIRKs and DIRKs with \( a_i \neq 0 \)), Higuera and Roldán [201] construct predictors where \( \eta = 0 \) is used [323]. Using this same single-step predictor structure, one may generate an interpolating polynomial by using \( U_i^{[n-1]} \) and \( U_i^{[n]} \), \( i = 1, 2, \ldots, s \) to determine \( U_i^{[n+1]} \), \( i = 1, 2, \ldots, s \), which enables one to compute \( \eta \) and \( B \) without solving the preceding order conditions. The predictor is then used in extrapolation like in the dense-output stage-value predictors. In this case, the predictors are given by [347]

\[
\eta_i = \prod_{k=1}^{s} \left( \frac{1 + c_ir - c_k}{-c_k} \right), \quad B_{ij} = \frac{1 + c_ir}{c_j} \prod_{k=1,k\neq j}^{s} \left( \frac{1 + c_ir - c_k}{c_j - c_k} \right) \tag{174}
\]

where caution should be exercised on vanishing denominators. González-Pinto et al. [175] present a variation on these single-stage predictors for high stage-order methods. They consider two different stage-value predictions of the same stage by
using (174) and the analogous formula where \( \eta_i = 0 \). The stage-value prediction is then equal to the latter prediction, plus the product of the inverse iteration matrix times the difference between the two predicted values in order to introduce a stabilizing contribution.

The predictor stability functions at stage \( i \), \( R^{(i)}_{\text{pred}}(z) \), for (125), (126) and (127) are then given by [173–175]

\[
R^{(i)}_{\text{pred}}(z) = \eta_i + \sum_{j=1}^{s} B_{ij} R^{(j)}_{\text{int}}(z) \
\]

\[
R^{(i)}_{\text{pred}}(z) = \eta_i + \left( \sum_{j=1}^{s} B_{ij} R^{(j)}_{\text{int}}(r_1 z) \right) R(z) + \sum_{j=1}^{s} C_{ij} R^{(j)}_{\text{int}}(z) \
\]

\[
R^{(i)}_{\text{pred}}(z) = \eta_i + \left( \sum_{j=1}^{s} B_{ij} R^{(j)}_{\text{int}}(r_2 z) \right) R(r_1 z) R(z) + \left( \sum_{j=1}^{s} C_{ij} R^{(j)}_{\text{int}}(r_1 z) \right) R(z) \
+ \sum_{j=1}^{s} D_{ij} R^{(j)}_{\text{int}}(z),
\]

where \( R^{(j)}_{\text{int}}(z) \) is the internal stability function, (53), at stage \( j \). Applied to stiff problems, \( R^{(j)}_{\text{int}}(-\infty) = 0 \) implies that the starting algorithm cushions the very stiff components [174]. Three simple things may be done to enhance predictor stability: set \( \eta_i = 0 \), choose an L-stable underlying method, \( R(-\infty) = 0 \), and design a method so that \( R^{(j)}_{\text{int}}(-\infty) = 0 \).

When the \( A \)-matrix is singular because the first stage is explicit, as it is with EDIRKs, ESDIRKs and QESDIRKs, this single-step structure must be modified [202,204]. Modified stage-value predictors may now be considered in the form

\[
U^{[n]}_{i+1,0} = \tilde{\eta}_i U^{[n-1]} + \sum_{j=1}^{s-1} \tilde{B}_{ij} U^{[n]}_{j+1} \
\]

\[
U^{[n]}_{i+1,0} = \tilde{\eta}_i U^{[n-2]} + \sum_{j=1}^{s-1} \tilde{B}_{ij} U^{[n]}_{j+1} + \sum_{j=1}^{s-1} \tilde{C}_{ij} U^{[n-1]}_{j+1} \
\]

\[
U^{[n]}_{i+1,0} = \tilde{\eta}_i U^{[n-3]} + \sum_{j=1}^{s-1} \tilde{B}_{ij} U^{[n]}_{j+1} + \sum_{j=1}^{s-1} \tilde{C}_{ij} U^{[n-1]}_{j+1} + \sum_{j=1}^{s-1} \tilde{D}_{ij} U^{[n-2]}_{j+1}.
\]

where \( i = 1, 2, \cdots, (s - 1) \), the matrices \( \tilde{B}_{ij}, \tilde{C}_{ij}, \) and \( \tilde{D}_{ij} \) are of dimension \( (s - 1) \times (s - 1) \), and \( \tilde{\eta} \) is a vector of length \( (s - 1) \). The conventional Butcher array is then decomposed according to (3). Also, \( \tilde{b}, \tilde{c} \) and \( \tilde{e} \) (composed of ones) are vectors of length \( (s - 1) \), and \( \tilde{B} = \tilde{e} \otimes \tilde{b} \). The components of \( \tilde{\eta} \) are \( \tilde{c}, \tilde{c} = \tilde{e} + r_1 \tilde{c}, \tilde{c} = \tilde{e}(1 + r_1) + r_2 \tilde{c} \) and \( \tilde{c} = \tilde{e}(1 + r_1 + r_2) + r_3 \tilde{c} \). Order conditions for these methods...
are solved by using

\[ \mathbf{B}\mathbf{e} = \mathbf{e} - \eta, \quad \mathbf{B}\Phi_j^{(p)} = \Phi_j^{(p)} \]  
(181)

\[ \mathbf{B}\mathbf{e} + \mathbf{C}\mathbf{e} = \mathbf{e} - \eta, \quad \mathbf{C}\Phi_j^{(p)} + \mathbf{B}\Phi_j^{(p)} = \Phi_j^{(p)} \]  
(182)

\[ \mathbf{B}\mathbf{e} + \mathbf{C}\mathbf{e} + \mathbf{D}\mathbf{e} = \mathbf{e} - \eta, \quad \mathbf{D}\Phi_j^{(p)} + \mathbf{C}\Phi_j^{(p)} + \mathbf{B}\Phi_j^{(p)} = \Phi_j^{(p)}. \]  
(183)

Higueras and Roldán [202, 204] use these relations in the context of single-step predictors for ODEs and index-2 DAEs. Order conditions for first-stage explicit methods are slight modifications of those with invertible \( A \)-matrices. The difference terms \( d = (1 + r_1) \) and \( d = (1 + r_1 + r_2) \) will be useful. Bushy tree order conditions for the basic method and for one-, two- and three-step predictors are

\[ \hat{\Phi}_1^{(p)} = p(\hat{A}\hat{c}^{p-1}) \]  
(184)

\[ \hat{\Phi}_1^{(p)} = p(\hat{B}\hat{c}^{p-1} + r_1(\hat{A}\hat{c}^{p-1} + a)) \]  
(185)

\[ \hat{\Phi}_1^{(p)} = p(\hat{B}\hat{c}^{p-1} + r_1\hat{B}\hat{c}^{p-1} + r_2(\hat{A}\hat{c}^{p-1} + a\hat{c}^{p-1})) \]  
(186)

\[ \hat{\Phi}_1^{(p)} = p(\hat{B}\hat{c}^{p-1} + r_1\hat{B}\hat{c}^{p-1} + r_2\hat{B}\hat{c}^{p-1} + r_3(\hat{A}\hat{c}^{p-1} + a\hat{c}^{p-1})). \]  
(187)

Order conditions derived from the term \( \hat{\Phi} = (l + 1)(k + l + 2)\hat{B}\hat{C}^k\hat{A}\hat{c}^l \) are given by

\[ \hat{\Phi} = (l + 1)(k + l + 2)\hat{A}\hat{C}^k\hat{A}\hat{c}^l \]  
(188)

\[ \hat{\Phi} = (l + 1)(k + l + 2)[\hat{B}\hat{C}^k\hat{A}\hat{c}^l + r_1\hat{A}\hat{C}^k(\hat{B}\hat{c}^l + r_1\hat{A}\hat{c}^l + r_1a)] \]  
(189)

\[ \hat{\Phi} = (l + 1)(k + l + 2)[\hat{B}\hat{C}^k\hat{A}\hat{c}^l + r_1\hat{B}\hat{C}^k(\hat{B}\hat{c}^l + r_1\hat{A}\hat{c}^l + r_1a) \]
\[ + r_2\hat{A}\hat{c}^k(\hat{B}\hat{c}^l + r_1\hat{B}\hat{c}^l + r_2\hat{A}\hat{c}^l + r_2a\hat{c}^l)] \]  
(190)

\[ \hat{\Phi} = (l + 1)(k + l + 2)[\hat{B}\hat{C}^k\hat{A}\hat{c}^l + r_1\hat{B}\hat{C}^k(\hat{B}\hat{c}^l + r_1\hat{A}\hat{c}^l + r_1a) \]
\[ + r_2\hat{B}\hat{C}^k(\hat{B}\hat{c}^l + r_1\hat{B}\hat{c}^l + r_2\hat{A}\hat{c}^l + r_2a\hat{c}^l) \]
\[ + r_3\hat{A}\hat{C}^k(\hat{B}\hat{c}^l + r_1\hat{B}\hat{c}^l + r_2\hat{B}\hat{c}^l + r_3\hat{A}\hat{c}^l + r_3a\hat{c}^l)]. \]  
(191)

Expressions for \( \hat{\Phi}_2^{(3)} \), \( \hat{\Phi}_2^{(4)} \) and \( \hat{\Phi}_3^{(4)} \) or, alternatively, \( \hat{\Phi}_2^{(q+2)} \), \( \hat{\Phi}_2^{(q+3)} \) and \( \hat{\Phi}_3^{(q+3)} \) may be derived by using these relations. One can also write the expressions derived from
These equations provide an expression for \( \tilde{\Phi} \) when \( k = 0 \), \( l = 1 \), or \( \tilde{\Phi} \) when \( k = 0 \), \( l = q \). To solve for the predictor matrices, one may consider a nonconfluent five-stage, stage-order two ESDIRK and define the square matrices

\[
\begin{align*}
\mathcal{V} & = \{ \mathbf{e}, \tilde{\Phi}_1^{(1)}, \tilde{\Phi}_1^{(2)}, \tilde{\Phi}_1^{(3)} \} \\
\hat{\mathcal{V}} & = \{ \mathbf{e}, \hat{\Phi}_1^{(1)}, \hat{\Phi}_1^{(2)}, \hat{\Phi}_1^{(3)} \}, \quad \hat{\mathcal{V}}^* = \{ (\tilde{\mathbf{e}} - \hat{\eta}), \hat{\Phi}_1^{(1)}, \hat{\Phi}_1^{(2)}, \hat{\Phi}_1^{(3)} \} \\
\widehat{\mathcal{V}} & = \{ \mathbf{e}, \hat{\Phi}_1^{(1)}, \hat{\Phi}_1^{(2)}, \hat{\Phi}_1^{(3)} \}, \quad \widehat{\mathcal{V}}^* = \{ (\tilde{\mathbf{e}} - \hat{\eta}), \hat{\Phi}_1^{(1)}, \hat{\Phi}_1^{(2)}, \hat{\Phi}_1^{(3)} \} \\
\hat{\mathcal{V}} & = \{ \mathbf{e}, \hat{\Phi}_1^{(1)}, \hat{\Phi}_1^{(2)}, \hat{\Phi}_1^{(3)} \}, \quad \hat{\mathcal{V}}^* = \{ (\tilde{\mathbf{e}} - \hat{\eta}), \hat{\Phi}_1^{(1)}, \hat{\Phi}_1^{(2)}, \hat{\Phi}_1^{(3)} \},
\end{align*}
\]

where the dimension of the matrices is related to the number of nonconfluent stages. Now, one must solve

\[
\begin{align*}
\hat{B}\mathcal{V} &= \mathcal{V}^*, & \hat{C}\mathcal{V} + \hat{B}\mathcal{V} &= \mathcal{W}^*, & \hat{D}\mathcal{V} + \hat{C}\mathcal{V} + \hat{B}\mathcal{W} &= \hat{V}^*.
\end{align*}
\]

In cases where \( \hat{\mathcal{V}} \) is invertible (this requires that \( \hat{\mathbf{A}}^{-1} \) exists), one has \( \hat{B} = \hat{V}^*\hat{V}^{-1} \) and

\[
\begin{align*}
\hat{C} &= -\hat{B}\hat{V}\hat{V}^{-1} + \hat{W}^*\hat{V}^{-1}, & \hat{D} &= -\hat{C}\hat{V}\hat{V}^{-1} - \hat{B}\hat{W}\hat{V}^{-1} + \hat{V}^*\hat{V}^{-1}.
\end{align*}
\]

Higueras and Roldán [201] supplement their single-stage predictor, (125), with a strategy described by Hairer and Wanner [193] and used in the SDIRK4 code: they use early stage values to predict later stage values. Originally this strategy was based on the \( \Phi_j^{[p]} \) vectors, but Higueras and Roldán also extend it to include \( \Phi_j^{[p]} \) vectors. Before putting these order conditions to use, one may define \( Z_i^{[n]} = U_i^{[n]} - U^{[n]} \)
associated with stage $i$ at $t^n + c_i \Delta t$. The starting guess for stage $i$ is then $U_{i,0}^{[n]} = U_i^{[n]} + Z_{i,0}^{[n]}$. Hairer and Wanner propose to estimate $Z_{i,0}^{[n]}$ by using $Z_{i,0}^{[n]} = \sum_{j=1}^{i-1} \alpha_j Z_j^{[n]}$ where the values $\alpha_j$ may be precomputed for each stage. Depending on how many stages of the current step one has completed, one obtains different orders of accuracy.

Considering a five-stage SDIRK such as SDIRK4, $Z_2^{[n]}$ may be computed to first-order accuracy by using $\alpha_1 = c_2 / c_1$. If the first two (three) stages of an SDIRK have been computed then $Z_3^{[n]}$ ($Z_4^{[n]}$) may be computed to second-order accuracy by using

$$
\begin{pmatrix}
 c_1 & c_2 \\
 (Ac)_1 & (Ac)_2 \\
 \varphi_1^{(3)} & \varphi_2^{(3)} & \varphi_3^{(3)}
\end{pmatrix}
\begin{pmatrix}
 \alpha_1 \\
 \alpha_2 \\
 \alpha_3
\end{pmatrix}
= 
\begin{pmatrix}
 c_3 \\
 (Ac)_3 \\
 \varphi_3^{(3)}
\end{pmatrix},
\tag{202}
$$

where $\varphi^{(3)}$ is some combination of the two third-order trees, $Ac^2$ and $A^2c$. Note that these terms included are simply proportional to the $\Phi_j^{[p]}$ given above. On the fifth-stage, a third-order approximation may be used

$$
\begin{pmatrix}
 c_1 & c_2 & c_3 & c_4 \\
 (Ac)_1 & (Ac)_2 & (Ac)_3 & (Ac)_4 \\
 (Ac^2)_1 & (Ac^2)_2 & (Ac^2)_3 & (Ac^2)_4 \\
 (A^2c)_1 & (A^2c)_2 & (A^2c)_3 & (A^2c)_4
\end{pmatrix}
\begin{pmatrix}
 \alpha_1 \\
 \alpha_2 \\
 \alpha_3 \\
 \alpha_4
\end{pmatrix}
= 
\begin{pmatrix}
 c_5 \\
 (Ac)_5 \\
 (Ac^2)_5 \\
 (A^2c)_5
\end{pmatrix},
\tag{204}
$$

Hairer and Wanner use dense output for the first stage of SDIRK4, $\alpha_1 = c_2 / c_1$ for stage two, and (202), (203), and (204) for latter stages. Higuera and Roldán [201] found best results for this stage-order one method by using (125) and (165) to predict the first two stages and (203) and (204) for the fourth- and fifth stages. The third stage could be predicted with either (202) or (205) and (206). For EDIRKs, ESDIRKs and QESDIRKs, one must increase the subscripts by one. If row-simplifying assumption $C(\eta)$ is being used, the selection of order conditions must respect potentially redundant conditions. This same strategy may also use stage information from the latter stages of the previous step. For instance, to compute

$$
U_3^{[n]} = \alpha_1 U_3^{[n]} + \alpha_2 U_4^{[n]} + \alpha_3 U_5^{[n]} + \alpha_4 U_1^{[n+1]} + \alpha_5 U_2^{[n+1]}
\tag{205}
$$

one solves

$$
\begin{pmatrix}
 1 & 1 & 1 \\
 c_3 & c_4 & c_5 \\
 (Ac)_3 & (Ac)_4 & (Ac)_5
\end{pmatrix}
\begin{pmatrix}
 \Phi_1^{[2]} \\
 1 \\
 \Phi_1^{[3]}
\end{pmatrix}
= 
\begin{pmatrix}
 1 \\
 c_3 \\
 \Phi_1^{[3]}
\end{pmatrix}. \tag{206}
$$

Notice that when stages from two steps are involved, the summation is over $U_i^{[n]}$ and $U_i^{[n]}$ rather than $Z_i^{[n]}$. This strategy may be extended to include additional prior steps by using order conditions based on two- and three-step predictors.
2.19 Discontinuities

In the course of integrating the ODE

\[
\frac{dU}{dt} = F(U(t), t)
\]

the RHS function, \( F(U(t), t) \), or one of its elementary differentials may become discontinuous [81, 96, 138, 142, 166, 297, 325, 380, 427]. Examples of situations where this may occur include the time-dependent temperature in a room when a thermostat is employed or the time dependent velocity of an object which collides with another object. In computer codes used to solve time-dependent partial differential equations, temporal discontinuities may arise by changes in boundary conditions or changes within a submodel caused by conditional statements. If Runge-Kutta methods attempt to integrate over this discontinuity, both local and global errors will likely increase unless proper care is taken by the algorithm. A function \( F(U(t), t) \) has a discontinuity of order \( q - 1 \) if at least one of the partial derivatives with respect to time is discontinuous of order \( q - 1 \), but all partial derivatives with respect to time of order \( < q - 1 \) are continuous. Let \( F_L(U(t), t) \) and \( F_R(U(t), t) \) be the functions to the left and right, respectively, of the discontinuity and let a discontinuity be located at \( t_d \). The size of the \( q \)th-order discontinuity is given by

\[
\frac{d^{(q-1)}F_L}{dt^{(q-1)}} \bigg|_{t_d^+} - \frac{d^{(q-1)}F_R}{dt^{(q-1)}} \bigg|_{t_d^+} = K_q,
\]

where the functions are evaluated immediately to the right of the discontinuity, \( t_d^+ \). Only bounded discontinuities will be considered, i.e., \( K_q \) is finite. If at some value of \( t = t_s \), \( F(U(t), t) \) becomes infinite, then a singularity has been encountered. Discussions of singular initial value problems may be found in the literature [127, 147, 249, 260, 261, 418].

Some discontinuities may be anticipated while others may not. Those that are anticipated are often accompanied by switching functions, \( g(U(t), t) \). This function, \( g(U(t), t) = 0 \), defines the discontinuity hypersurface. Shampine and Thompson [380] consider well-posedness in this situation. If the integration is well-posed on the interval \([a, t_d]\) with \( U(a) = A \), then a small perturbation to the initial condition; \( U(a) = A + \alpha \) will give rise to a small perturbation in \( U_L(t) \) by the amount \( \delta U_L \). This then induces a small change in the location of the discontinuity; \( \delta t \). Taking the first terms of a Taylor series expansion about \( U(t) \) and \( t_d \), where \( g(U(t_d), t_d) = 0 \),

\[
g(U_L(t_d + \delta t) + \delta U_L, t_d + \delta t) = g(U_L(t_d), t_d) + \frac{\partial g}{\partial U} \bigg|_{t_d} \delta U_L
\]

\[
+ \left( \frac{\partial g}{\partial t} \bigg|_{t_d} + \frac{\partial g}{\partial U} \bigg|_{t_d} \frac{\partial U}{\partial t} \bigg|_{t_d} \right) \delta t + \cdots
\]

\[
= 0
\]

Hence,

\[
\delta t = - \frac{\partial g}{\partial U} \bigg|_{t_d} \left( \frac{\partial g}{\partial t} \bigg|_{t_d} + \frac{\partial g}{\partial U} \bigg|_{t_d} \frac{\partial U}{\partial t} \bigg|_{t_d} \right)^{-1} \delta U_L.
\]
Provided that the denominator of the above expression does not vanish, the problem is well-posed with $\delta U_L$ proportional to $\delta t$. Another aspect of well-posedness is the transversality condition. Mannshardt [297] defines a transversality condition associated with $g(U(t), t)$. This condition requires that $g(U(t), t)$ change sign across the discontinuity, or

$$\left.\frac{dg(U(t), t)}{dt}\right|_{t_d^-} \geq G \quad ; \quad \left.\frac{dg(U(t), t)}{dt}\right|_{t_d^+} \geq G$$

(211)

for some constant $G > 0$. If this condition were not to hold, solutions could slide along the switching hypersurface or along intersections of multiple switching hypersurfaces. Lastly, multiple simultaneous events could lead to an ill-posed problem. Events must be sufficiently isolated so that the numerical method is able to distinguish between adjacent events, and ensures that the events occur in the proper order. Associated with questions of well-posedness are those of convergence [297, 380].

Managing discontinuous RHS functions is typically done using a three step process. First, the event must be detected. This is often accomplished by observing the output of an error estimator. If there is sufficient reason to believe that a discontinuity has just been encountered, the second step is to locate the event. Depending on the objective, the event location can be determined as precisely as needed. Lastly, once the discontinuity has been located to the required precision based on the user-specified local error tolerance, the third step is to cross the discontinuity. As Runge-Kutta methods are self-starting, restarting the integration after the discontinuity has been crossed requires no special treatment.

In cases where an event function is available, these tasks become more straightforward. Carver [96] differentiates the event function to write a differential equation with its initial condition

$$\frac{dg}{dt} = g'; \quad g_0 = g(U(t_0), t_0)$$

(212)

which is integrated along with the original system of ODEs. Typically, an event would be detected during an integration by looking for a sign change in the event function. Carver defines $s_1 = g_n * g_{n+1}$ and $s_2 = g_n' * g_{n+1}'$. With these two quantities, he is able to determine whether a discontinuity is present between $t^{[n]}$ and $t^{[n+1]}$ and how the integration should proceed in the following three cases:

- $s_1 > 0$ and $s_2 > 0$: No discontinuity is present between $t^{[n]}$ and $t^{[n+1]}$.
- $s_1 < 0$ and $s_2 > 0$: There is at least one discontinuity in the interval. One event location can be computed by solving $g(U(t), t) = 0$ for $t$.
- $s_1 > 0$ and $s_2 < 0$: There are at least two discontinuities in the interval. The step-size should be reduced until less than two discontinuities exist on the interval.

At this point, the event has been detected and approximately located but not crossed.
Event detection, location and crossing often occur in the general situation where no event function is available. Several general criteria have been presented in the literature to detect the presence of a discontinuity in the course of integrating first-order ODEs. They share the objectives of being inexpensive and relying on functionality already present in the integration algorithm. Regardless of which criterion is chosen, discontinuities having large values of \( q \) but small values of \( K \) may go undetected. Further, since local error estimates are used in event detection, lax local error tolerances may also contribute to a discontinuity going undetected. Rather than going undetected, caution should be made to ensure that a discontinuity is not confused with rapidly varying RHS caused by numerical stiffness. If the integration method lacks sufficient stability, discontinuity detection may be routinely activated by stiffness.

Assume that the integration has successfully stepped from \( t^{[n-1]} \) to \( t^{[n]} \) and is now attempting to step from \( t^{[n]} \) to \( t^{[n+1]} \) using an initial step-size of \( (∆ t)^{[n]}_1 \). Assume that the step is rejected by the error controller. A rejected step is typically when event detection is initiated. Gear and Østerby [166] recompute a new step-size to be used on the failed step, \( (∆ t)^{[n]}_2 \). A discontinuity is suspected if either criterion is met:

- (G\( \overline{O} \)1) \( 2(∆ t)^{[n]}_2 < (∆ t)^{[n]}_1 \) then a discontinuity is suspected of being in the interval \( \{t^{[n]}, t^{[n]} + (∆ t)^{[n]}_1\} \).

- (G\( \overline{O} \)2) There are two successive failed steps, i.e., both \( (∆ t)^{[n]}_1 \) and \( (∆ t)^{[n]}_2 \) result in failed steps. In this case, a discontinuity is suspected of being in the interval \( \{t^{[n]}, t^{[n]} + (∆ t)^{[n]}_2\} \).

Enright et al. [142] suspect a discontinuity if:

- (E1) After a step from \( t^{[n]} \) to \( t^{[n+1]} \) has been rejected \( j \)-times, it is accepted on the \((j + 1)\)-th attempt with the step-size \( (∆ t)^{[n]}_{j+1} \). A new step-size is estimated from \( t^{[n+1]} \) to \( t^{[n+2]} \) as \( (∆ t)^{[n]}_{j+1} \). If \( (∆ t)^{[n+1]} \geq 2(∆ t)^{[n]}_{j+1} \), then a discontinuity is suspected of being in the interval \( \{t^{[n+1]}, t^{[n+1]} + (∆ t)^{[n]}_{j+1}\} \).

Lastly, Calvo et al. [81] suspect a discontinuity if either:

- Same as G\( \overline{O} \)1

- In the last three steps, the code has at least two step rejections.

In some cases, event detection should be disabled. Two such cases are immediately following the crossing of a discontinuity and at the beginning of the integration when approximate (nonequilibrium) initial conditions have been applied.

Event location may be approached in several different ways [81, 142, 166]. Each of the proposed strategies are quite involved and will not be reviewed here. These papers discuss how the discontinuity is located to an accuracy sufficient to preserve the local error tolerance prescribed for the integration when the event is crossed.

Once the discontinuity has been crossed and the local error has been preserved, the integration is simply resumed with the event detection logic deactivated for the first step after the crossing.
2.20 Software

Early efforts at ODE software using DIRK-type methods, apart from methods embedded within an application code, include DIRK, a code by R. Alexander based on his 1977 paper [5]. The code is based primarily on stiffly-accurate, L-stable methods. Gaffney [161] uses DIRK to compare DIRK-type methods with SIRK, BDF, cyclic composite multistep, and blended linear multistep methods. Nørsett and Thom-ensen [315] create the SIMPLE code along with the SPARKS subroutines [211]. SIM-PLE uses the methods given in papers by the authors [312,313]. It is a descendent of two previous codes, SIRSPN [308] and SIRKUSS [283,308] by Nørsett. Later, Hindmarsh and Nørsett [207] modified SIMPLE with a Newton-Krylov method, naming the new code KRYSI. Similar to DIRK, Hairer and Wanner [193] created the code SDIRK4 based on two five-stage, fourth-order, L-stable, stiffly-accurate SDIRK methods. It is freely available from the first author’s website. Compere [109] has repackaged SDIRK4 with a C++ interface. Lastly, Olsson [321, 322] has included the ESDIRK methods of Kværno [281] into an object-oriented solver named GODDESS. A Newton-Krylov version of GODDESS has been made by Thomsen and Bjurstrøm [42,428] for the solution of large systems of ODEs.

3 Early Methods

Probably the earliest DIRK-type methods, apart from the first-order accurate, L-stable implicit Euler method, are the implicit midpoint and trapezoidal (Lobatto IIIA) rules, both of which are second-order accurate and A-stable [190]. In the same order, their tableaus are given by

\[
\begin{array}{c|ccc}
1 & 1 & & \\
& 1 & & \\
\end{array} \quad \begin{array}{c|ccc}
1 & 1 & & \\
\frac{1}{2} & \frac{1}{2} & & \\
\frac{1}{2} & \frac{1}{2} & & \\
\end{array} \quad \begin{array}{c|ccc}
0 & 0 & 0 & \\
1 & \frac{1}{2} & \frac{1}{2} & \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \\
\end{array}
\] (213)

In 1955, Hammer and Hollingsworth [195] gave a two-stage, third-order accurate E(S)DIRK scheme which they stated was “exactly” the trapezoidal rule (Radau I), and in 1964, Butcher [60] gave the three-stage, fourth-order EDIRK (Lobatto III)

\[
\begin{array}{c|ccc}
0 & 0 & 0 & \\
\frac{2}{3} & \frac{1}{3} & \frac{1}{3} & \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \\
\end{array} \quad \begin{array}{c|ccc}
0 & 0 & 0 & \\
\frac{1}{2} & \frac{1}{4} & \frac{1}{4} & \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \\
\end{array}
\] (214)

Neither scheme is A-stable, but both (the implicit-midpoint and trapezoidal rules too) have stage-order two. Also neglecting stability properties, in 1966, Ceschin and Kunzmann [103] designed two stiffly-accurate, third- and fourth-order EDIRKs
by using the $C(2)$ assumption

\[
\begin{bmatrix}
0 & 0 & 0 & 0 \\
1/3 & 1/3 & 1/3 & 0 \\
1/6 & 1/6 & 1/6 & 1/6 \\
1/6 & 1/6 & 1/6 & 1/6 \\
\end{bmatrix}
\quad \quad \quad \quad
\begin{bmatrix}
0 & 0 & 0 & 0 \\
1/3 & 1/3 & 1/3 & 0 \\
1/6 & 1/6 & 1/6 & 1/6 \\
1/6 & 1/6 & 1/6 & 1/6 \\
\end{bmatrix}
\]

(215)

where $E(y) = -(1/72)y^4$ and $E(y) = -(7/13824)y^6$, respectively (40). In a more incidental way, Ehle [136] (1969) and Chipman [106] (1971) mention several DIRK-type methods in their respective theses. In his 1971 thesis, Alt [19, 21, 22] derives third- and fourth-order, stage-order two, stiffly-accurate EDIRKs in three and four stages

\[
\begin{bmatrix}
0 & 0 & 0 & 0 \\
3/2 & 3/2 & 0 \\
1/18 & 1/18 & 1/18 & 1/18 \\
\end{bmatrix}
\quad \quad \quad \quad
\begin{bmatrix}
0 & 0 & 0 & 0 \\
3/2 & 3/2 & 0 \\
1/18 & 1/18 & 1/18 & 1/18 \\
\end{bmatrix}
\]

(216)

The third-order method is characterized by $A^{(4)} = 0.1550$, $E(y) = (13/72)y^4$, $R_{\text{int}}(-\infty) = \{1, -1, -11/15\}$, and hence is A-stable. For the fourth-order method, $A^{(5)} = 0.3934$, $E(y) = (181259/129600)y^6$ and $R_{\text{int}}(-\infty) = \{1, -1, -268/285, -379/750\}$ so that it is also A-stable. In unpublished lecture notes of 1973, Miller [299] constructs two stiffly-accurate, L-stable, DIRK methods

\[
\begin{bmatrix}
1/3 & 1/3 & 1/3 & 1/3 \\
1/6 & 1/6 & 1/6 & 1/6 \\
\end{bmatrix}
\quad \quad \quad \quad
\begin{bmatrix}
1 & 1/2 & 1/2 & 1/2 \\
1/3 & 1/3 & 1/3 & 1/3 \\
\end{bmatrix}
\]

(217)

More recently, the two-stage method was given again by Baker [28, 29]. Their leading-order errors are, respectively, $A^{(3)} = 0.1550$ and $A^{(4)} = 0.04944$. Nørsett [308] chose to focus on SDIRK methods in his 1974 thesis. Unlike his contemporaries, he was also concerned with error estimation through an embedded method requiring an additional stage. Two of the three basic SDIRKs that he presented were discussed previously in (50). The third method is the general solution to the four-stage, fourth-order SDIRK configurable as either an A- or L-stable method depending on the choice of $\gamma$.

Kurdi [278] investigated stage-order two, EDIRKs in his 1974 thesis. Both he and Crouzeix [121] constructed the three-stage, third-order, A-stable, stiffly-accurate ES-EDIRK. In addition, he designed two fourth-order, five-stage, stiffly-accurate meth-
They have, from left to right, $A^{(5)}(5) = 0.07498$ and $A^{(5)}(5) = 0.06615$ while their internal stability is characterized by $R_{\text{int}}(-\infty) = \{1, -1, -3/4, -23/32, 0\}$ and $R_{\text{int}}(-\infty) = \{1, -1, -3/4, -19/24, 0\}$. Crouzeix [120, 121, 123] has derived many DIRK-type methods. His 1975 thesis offers two SDIRK methods, (50), that also appear independently in Nørsett’s 1974 thesis. These methods are third- and fourth-order accurate. In 1977, Alexander [5] provides a review of many early efforts, as well as additional methods which will be discussed later. In 1979, Cash [97] extends Alexander’s paper to include embedded methods.

4 Two- and Three-stage Methods ($S_1 = 2$)

4.1 Second-Order Methods

4.1.1 ESDIRK

Three-stage, stiffly-accurate, stage-order two ESDIRKs with external error-control take the form

\[
\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\gamma & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & (1 - b_2 - \gamma) & b_2 & \gamma & 0 & 0 & 0 \\
\frac{b_2}{b_1} & (1 - b_2 - \gamma) & b_2 & \gamma & 0 & 0 & 0 \\
\frac{b_2}{b_1} & (1 - b_2 - \gamma) & b_2 & \gamma & 0 & 0 & 0
\end{array}
\]

Internal error-control for a 2(1) pair is impossible because the stage order is two. Four degrees of freedom (DOF) are available in the main method before $C(2, 2)$ ($a_{21} = \gamma$) and $r_1^{(1)}$ ($b_1 = 1 - b_2 - \gamma$) are imposed and three in the embedded method before $\tilde{r}_1^{(1)}$ ($\tilde{b}_1 = 1 - \tilde{b}_2 - \tilde{b}_3$) is applied. A second-order, L-stable method is constructed by solving the four equations: $r_1^{(1,2)} = p_2 = q_2^{(2)} = 0$, to find that $b_2 = (1 - 2\gamma)/(4\gamma)$, $A^{(3)} = (1 - 6\gamma + 6\gamma^2)/\sqrt{18}$ and $E_4 = (2\gamma - 1)^2(4\gamma - 1)/4$. A third-order L-stable method is impossible. For L-stability, $p_2 = 0$ and, hence, $\gamma$ must be a root of the polynomial $2\gamma^2 - 4\gamma + 1 = 0$ or $\gamma = (2 \pm \sqrt{2})/2$. Each value also ensures I-stability because $E_4$ remains non-negative as long as $\gamma \geq 1/4$. To attain higher order dissipation errors, one may obtain $E_4 = 0$ at $\gamma = 1/4, 1/2$. Because it is desirable to have a value of $\gamma$ as small as possible and abscissae within the integration interval, one selects $\gamma = (2 - \sqrt{2})/2 \approx 0.2929$, resulting in $A^{(3)} = 0.05719$. That, with the embedded method listed below, constitutes ESDIRK2(1)3L[2]SA. Bank et
al. [31], and more recently Hosea and Shampine [209] and Butcher and Chen [72], give this method. Internal stability values for the three stages are $R^{(i)}_{\text{int}}(-\infty) = \{1, -1, 0\}$. Contractivity radii may be calculated to be $r_{F_2} = 2.780$ and $r_{F_\infty} = 2.414$.

First-order, external error-control is accomplished by solving $\hat{\tau}_1^{(1)} = 0$, $\hat{\rho}_3 = 0$ and $\hat{R}(-\infty) = \hat{\rho}_2/\gamma^2 = \hat{R}(-\infty)$. Selecting $\hat{R}(-\infty) = \gamma$, which prevents the main and embedded methods from colliding and keeps $\hat{R}(-\infty)$ reasonably small, gives

$$
\hat{b}_2 = \frac{\gamma(-2 + 7\gamma - 5\gamma^2 + 4\gamma^3)}{2(2\gamma - 1)}, \quad \hat{b}_3 = \frac{-2\gamma^2(1 - \gamma + \gamma^2)}{(2\gamma - 1)}
$$

(220)

with $\hat{E}_4 = -\gamma^4(\gamma - 1)(\gamma + 1) > 0$, $\hat{E}_2 = -1 + 4\gamma - 2\gamma^2 + 2\gamma^3 > 0$ and $B^{(3)} = 3.105$, $C^{(3)} = 0.8284$ and $E^{(3)} = 2.276$ when $\gamma = (2 - \sqrt{2})/2$. The embedded method is then A-stable with $r_{F_2} = 1.563$ and $r_{F_\infty} = 0.7071$. Hosea and Shampine [209] also offer a method with $\gamma = 1/4$, which is not L-stable but is stiffly-accurate. In both of their methods, which they call TR-BDF2 and TRX2, a third-order embedded method with less desirable stability characteristics is appended to the main method rather than a second-order one. Based on knowing only the article title and abstract, Wen et al. [448] also study ESDIRK methods having two implicit stages.

### 4.1.2 SDIRK

Two-stage, stiffly-accurate, L-stable SDIRK methods allow one DOF

$$
\begin{array}{c|ccc}
\gamma & \gamma & 0 \\
1 & (1 - \gamma) & \gamma \\
\hline
b_i & (1 - \gamma) & \gamma \\
\hline
b_i & (1 - b_2) & b_2
\end{array}
$$

(221)

after $\tau_1^{(1)}$ is enforced ($b_1 = 1 - \gamma$). Solving the two order conditions necessary to achieve second order, $\tau_1^{(1,2)}$, yields two solutions: $\gamma = (2 \pm \sqrt{2})/2$. Both are L-stable. The $\gamma = (2 - \sqrt{2})/2$ solution is the far better choice because $\gamma$ is less than unity. Moreover, the $\gamma = (2 - \sqrt{2})/2$ solution is much more accurate: $A^{(3)} = 0.04168$ versus $A^{(3)} = 1.41605$. For $\gamma = (2 - \sqrt{2})/2$, one finds $D = 1.000$, $r_{F_2} = 8.243$ and $r_{F_\infty} = 2.414$. This method has been given by Alexander [5], Crouzeix and Raviart [123], and Crouzeix [121] with $\gamma$-bounds given in relation to A-stability measures. Luo [294] uses this method for unconstrained optimization. Though not stiffly-accurate, Fränken and Ochs [158] derive passive, L-stable, SDIRKs with $p = s = 2$. 

72
4.2 Third-Order Methods

4.2.1 ESDIRK

Dispensing with L-stability via the the stiffly-accurate assumption, stage-order two ESDIRK methods are written as

\[
\begin{array}{ccc}
0 & 0 & 0 \\
2\gamma & \gamma & \gamma \\
c_3 & (c_3 - a_{32}) & a_{32} \\
b_i & (1 - b_2 - b_3) & b_2 & b_3 \\
\end{array}
\]

(222)

instead of (219), which permits \((s^2 + s + 2)/2 = 7\) degrees of freedom (DOF). The general solution is a one-parameter family of methods, first given by Kurdi [278] and Crouzeix [121] for \(c_3 = 1\) and later by Cooper and Sayfy [116], Crouzeix and Raviart [123], and Shintani [384]. For A-stability, \(\gamma\) must be a root of \(p_3 = 6\gamma^2 - 6\gamma + 1 = 0\) and one must ensure that the method is also I-stable. These conditions are only possible for \(\gamma = (3 + \sqrt{3})/6 \approx 0.7887\), which implies that \(c_2 \approx 1.577\). In terms of \(c_3\), the rest of the solution is \(a_{32} = c_3(c_3 - 2\gamma)/4\gamma\), \(b_2 = (-2 + 3c_3)/(12(c_3 - 2\gamma)\gamma)\) and \(b_3 = (1 - 3\gamma)/(3c_3(c_3 - 2\gamma))\). The error is given by \(A^{(4)} = \sqrt{\mu_0 + \mu_1 c_3 + \mu_2 c_3^2}/(36\sqrt{3})\) where \(\mu_0 = 79+38\sqrt{3}, \mu_1 = -10(15+7\sqrt{3})\) and \(\mu_2 = 60(2+\sqrt{3})\). Internal stability is given by \(R^{(3)}_{\text{int}}(-\infty) = (\nu_0 + \nu_1 c_3^2 + \nu_2 c_3^3)/(2+\sqrt{3})\) where \(\nu_0 = 2+\sqrt{3}, \nu_1 = -2(3+\sqrt{3})\) and \(\nu_2 = 3\). If one wishes \(R^{(3)}_{\text{int}}(-\infty) = 0\) then \(c_3 = (3 + \sqrt{3})/3 - \sqrt{(2 + \sqrt{3})}/3 \approx 0.4620\) with \(A^{(4)} = 0.1316\). Minimum error occurs at \(c_3 = (15 + 7\sqrt{3})/(12(2 + \sqrt{3}))\) where \(A^{(4)} = 0.1270, R(-\infty) = -0.7321\) and \(R^{(3)}_{\text{int}}(-\infty) = -0.2410\). If \(c_3 = 1\), the A-stable method becomes stiffly-accurate with \(A^{(4)} = 0.1584\) and \(R^{(3)}_{\text{int}}(-\infty) = R(-\infty) = -0.7321\). It is impossible to append a second-order, A-stable embedded method onto this scheme. Allowing different diagonal elements, Cameron [88, 89] offers a third-order, stiffly-accurate EDIRK. This provides little gain over the stiffly-accurate ESDIRK. Abdalkhani [3] also offers a third-order, stiffly-accurate EDIRK.

4.2.2 SDIRK

Third-order SDIRKs using two stages are not stiffly-accurate and take the form

\[
\begin{array}{ccc}
\gamma & \gamma & 0 \\
c_2 & (c_2 - \gamma) & \gamma \\
b_i & (1 - b_2) & b_2 \\
\end{array}
\]

(223)

There are only two solutions to the order conditions of this method: \(\gamma = (3 + \sqrt{3})/6\). The larger of the two provides I-stability and was first given by Norsett [308], Crouzeix [120] and Scherer [357]. The Butcher array has been written in (50). Leading-order error is \(A^{(4)} = 0.1270\). Androulakis et al. [23] call these coefficients optimal. Fortunately, it is both A-stable and algebraically stable with \(R(-\infty) = -0.7321\). It is also a passive Runge-Kutta method [158]. A second-order embedded method is impossible with this method. Slightly related, Calahan [75] derives a second-order method in 1968 with this form and \(b_2 = 1/4\), but in the context of Rosenbrock methods.
4.3 Fourth-Order Methods

4.3.1 ESDIRK

Pursuing the maximum order possible while retaining a stage-order of two produces two methods of fourth-order where \(|R(z)| \to \infty\) as \(z \to -\infty\) because \(\deg P(z) = \deg Q(z) + 1\). Hence, despite their high-order, these methods are not useful for stiff equations. The methods are determined by using \(C(2)\) and \(D(1)\). The first method has \(\gamma = 1/2, c_2 = 1, c_3 = 1/2, a_{32} = -1/8, b_2 = 1/6, b_3 = 2/3\) and \(A^{(5)} = 0.01740\) while the second has \(\gamma = 1/6, c_2 = 1/3, c_3 = 5/6, a_{32} = 5/8, b_2 = 1/2, b_3 = 2/5\) and \(A^{(5)} = 0.001933\). These methods have been given by Ozawa [333] in the context of functionally fitted methods. Reconsidering these as EDIRKs to free up one DOF for enforcing \(p_3 = 0\) causes the Butcher coefficients to become imaginary.

5 Three- and Four-stage Methods (\(S_1 = 3\))

5.1 Third-Order Methods

5.1.1 ESDIRK

Stiffly accurate ESDIRK methods in four stages have \((s^2 - s + 2)/2 = 7\) available DOF. Stage-order two methods appear in Butcher array form as

\[
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
2\gamma & \gamma & \gamma & 0 & 0 \\
c_3 & (c_3 - a_{32} - \gamma) & a_{32} & \gamma & 0 \\
1 & (1 - b_2 - b_3 - \gamma) & b_2 & b_3 & \gamma \\
\hline
b_i & (1 - b_2 - b_3 - b_4) & b_2 & b_3 & b_4
\end{array}
\] \quad (224)

To enforce third-order overall accuracy for ODEs, a stage-order of two, and L-stability, one must satisfy the following six equations: \(\tau_1^{(1,2,3)} = p_3 = q_{2,3}^{(2)} = 0\), leaving a one-parameter family of methods in \(c_3\). The general solution is

\[
a_{32} = \frac{c_3(c_3 - 2\gamma)}{4\gamma}, \quad b_2 = \frac{-2 + 3c_3 + 6\gamma(1 - c_3)}{12\gamma(c_3 - 2\gamma)}, \quad b_3 = \frac{1 - 6\gamma + 6\gamma^2}{3c_3(c_3 - 2\gamma)}, \quad \text{where } c_3 - 2\gamma \neq 0 \text{ and } c_3 \neq 0, E_4 = (1 - 12\gamma + 36\gamma^2 - 24\gamma^3)/12 \quad \text{and } E_6 = (3\gamma - 1)(6\gamma - 1)(-1 + 9\gamma - 18\gamma^2 + 12\gamma^3)/36.
\]

Solutions do not exist for the special cases of \(c_3 - 2\gamma = 0\) and \(c_3 = 0\). The general solution results in an internal stability function and leading-order error given by

\[
R_{int}^{(3)}(-\infty) = \frac{c_3^2 - 4c_3\gamma + 2\gamma^2}{2\gamma^2}, \quad A^{(4)} = \sqrt{\frac{\mu_0 - 20c_3\mu_1 + 40c_3^2\mu_2}{1296}}, \quad \text{where } \mu_0 = 27 - 408\gamma + 2332\gamma^2 - 6504\gamma^3 + 9864\gamma^4 - 7776\gamma^5 + 2592\gamma^6, \mu_1 = (1 - 6\gamma + 6\gamma^2)^2 \text{ and } \mu_2 = 3 - 38\gamma + 162\gamma^2 - 264\gamma^3 + 144\gamma^4.
\]

L-stability requires that \(\gamma\) is a root of the polynomial \(p_3 = 6\gamma^3 - 18\gamma^2 + 9\gamma - 1 = 0\). One may also show that

\[
E_{int}^{(2)}(y) = 0, \quad E_{int}^{(3)}(y) = \frac{c_3(4\gamma - c_3)(c_3 - 2\gamma)^2 y^4}{4} \quad (227)
\]

74
so that internal I-stability is achieved on stage 3 for $c_3 \geq 0$ and $4\gamma \geq c_3$. Demanding I-stability of the step requires that $E_4, E_6 > 0$ and precludes two of the roots, leaving $\gamma = 0.43586652150845899941601945$. This implies that for $\gamma \approx 0.4358$ and $0 \leq c_3 \leq 1.743$, the third stage is I-stable. This basic scheme has been considered by many researchers [7, 8, 88, 89, 281, 396, 398, 426, 450, 451], though probably first by Cooper and Sayfy [116] and later by Alexander and Coyle [8] in the context of DAEs with internal error-control and a single-parameter family of methods. Second-order, external error-control is accomplished by solving $\hat{\gamma}^{(1,2)}_1 = \hat{p}_4 = 0$, $\hat{R}(\infty) = -\hat{p}_3/\gamma^3 = \gamma/2$. For a A-stability, one has

One may also elect to increase an element of the stage-order vector, taking it from $\{3,2,2,3\}$ to $\{3,2,3,3\}$ by selecting $c_3 = (3 \pm \sqrt{3})\gamma$. To keep $c_3$ less than unity, one selects $c_3 = (3 - \sqrt{3})\gamma$ to find $A^{(4)} = 0.03693$ and $R^{(3)}_{\text{int}}(\infty) = -0.7321$. Internal error-control via an A- or L-stable, internal, three-stage, second-order method requires that $c_3 = 1$ and $\bar{E}_4 = (2\gamma - 1)^2(4\gamma - 1)/4 \geq 0$. Hence, $\gamma \geq 1/4$ for A-stability of the internally embedded method and, from before, $\gamma = (2 \pm \sqrt{2})/2$ for L-stability. The embedded method is therefore A-stable but not L-stable. This scheme has been given by Kværnø [281, 282] and will be called ESDIRK3(2I)4L2[2]SA. Error and stability are given by $A^{(4)} = 0.04907$, $\hat{R}(\infty) = R^{(3)}_{\text{int}}(\infty) = (2\gamma^2 - 4\gamma + 1)/2\gamma^2 \approx -0.9567$, $\hat{A}^{(3)} = 0.1120$, $B^{(4)} = 1.476$, $C^{(4)} = 1.912$ and $E^{(4)} = 0.4380$. Optimal internal stability is achieved when $R^{(3)}_{\text{int}}(\infty) = 0$ by setting $c_3 = \gamma(2 \pm \sqrt{2})$. Keeping the abscissa within the current step, the $c_3 = \gamma(2 - \sqrt{2})$ solution is selected for $A^{(4)} = 0.04705$. An alternative approach to internal error-control is given by Williams et al. [450, 451] and uses the trapezoidal rule, (213), for the first two stages; that is, $\gamma = 1/2$. The 3(2) method is A-stable and has $A^{(4)} = 0.1058$ and $R^{(3)}_{\text{int}}(\infty) = -0.5$. Cameron [89] designs two A-stable 3(2) pairs by using the third- and fourth-stages for his two solutions. Thomsen [426] elects to forfeit L-stability by setting $\gamma = 5/12 \approx 0.4167$, but achieves a 3(2) pair with $R^{(3)}_{\text{int}}(\infty) = \{1, -1, -31/49, 17/125\}$ and $A^{(4)} = 0.03291$. Cameron [88] proposes a method with internal error-control and $\gamma = 17/50 = 0.34$. Skvortsov’s [396] stiffly-accurate 3(2) pair has $\gamma \approx 0.1590$. Reduced storage via a van der Houwen strategy [256] requires that $a_{31} = b_1$, giving the unacceptably large value of $c_3 \approx 1.825$. Corresponding error and internal stability values are $A^{(4)} = 0.1080$ and $R^{(3)}_{\text{int}}(\infty) = 1.394$. Conditional contractivity requires that all $b_i > 0$, a criterion that cannot be met by this single-parameter family of methods. Second-order, external error-control is accomplished by solving $\hat{\gamma}^{(1,2)}_1 = \hat{p}_4 = 0$, $\hat{R}(\infty) = -\hat{p}_3/\gamma^3 = \gamma/2$. For a A-stability, one has
\[
\hat{b}_2 = \frac{c_3(-1 + 6\gamma - 24\gamma^3 + 12\gamma^4 - 6\gamma^5)}{4\gamma(2\gamma - c_3)(1 - 6\gamma + 6\gamma^2)} + \frac{(3 - 27\gamma + 68\gamma^2 - 55\gamma^3 + 21\gamma^4 - 6\gamma^5)}{2(2\gamma - c_3)(1 - 6\gamma + 6\gamma^2)}
\]
\[
\hat{b}_3 = \frac{-\gamma(-2 + 21\gamma - 68\gamma^2 + 79\gamma^3 - 33\gamma^4 + 12\gamma^5)}{c_3(c_3 - 2\gamma)(1 - 6\gamma + 6\gamma^2)}
\]
\[
\hat{b}_4 = \frac{-3\gamma^2(-1 + 4\gamma - 2\gamma^2 + \gamma^3)}{(1 - 6\gamma + 6\gamma^2)}
\]

with \(\hat{E}_4 = \left(-1 + 12\gamma - 48\gamma^2 + 72\gamma^3 - 28\gamma^4 + 12\gamma^5\right)/4 > 0\) and \(\hat{E}_6 = -\gamma^6(\gamma - 2)/(\gamma + 2)/4 > 0\). Selecting this embedded method, L-stability and \(c_3 = 3/5\), one may derive ESDIRK3(2I)4L[2]SA where stage 3 is strongly A-stable \([255]\). The embedded method is characterized by \(\hat{A}_{(3)} = 0.02552, B_{(4)} = 2.907, C_{(4)} = 1.641\) and \(E_{(4)} = 1.435\). If \(c_3 = 1\), one may compare the nonstiffly-accurate, stage-order one external value, \(\hat{U}_{(n+1)}\), with the stiffly-accurate, stage-order two internal value of \(\hat{U}_{(n+1)}\). This comparison may provide some measure of the stiffness. In this case, ESDIRK3(2I)4L[2]SA, one has \(B_{(4)} = 3.831, C_{(4)} = 1.912\) and \(E_{(4)} = 1.923\) for the externally embedded method. Alexander \([7]\) compares a 3(2) with a 3(4) method and finds that, due to the conservative nature of the error estimate of the 3(2) pair, the 3(4) pair is more efficient in practice.

### 5.1.2 QESDIRK

One can use a stiffly-accurate QESDIRK, rather than an ESDIRK, by solving virtually the identical order conditions as the increased SOV method but try to mimic stage-order three behavior at relatively lax tolerances by reducing the error in \(q_{(3)}\) with \(SOV = \{3, 2, 3, 3\}\). As the previous ESDIRK has the unpleasantly large value of \(c_2 \approx 0.8717\), this QESDIRK method provides an opportunity to investigate the utility of differing nonzero diagonal elements \([227]\). The Butcher array for such a method is

\[
\begin{array}{c|cccccc}
\text{0} & 0 & 0 & 0 & 0 & 0 \\
c_2 & c_2/2 & c_2/2 & 0 & 0 \\
c_3 & (c_3 - a_{32} - \gamma) & a_{32} & \gamma & 0 \\
1 & (1 - b_2 - b_3 - \gamma) & b_2 & b_3 & \gamma \\
b_i & (1 - b_2 - b_3 - b_4) & b_2 & b_3 & b_4 \\
\end{array}
\]

(228)

If \(c_2 = 2\gamma\), then the ESDIRK is retrieved. Table 9 lists three possible L-stable QESDIRKs for the smaller of the two roots giving \(c_3\). Only the case where \(c_2 = \gamma\) seems worthy of any further consideration; hence, we give the two nontrivial abscissae: \(c_2 = 4266051685502/7784516477473\) and \(c_3 = 3420827508294/7669858543565\).
Table 9. Properties of various stiffly-accurate, L-stable, four-stage, third-order QES-DIRKs.

<table>
<thead>
<tr>
<th>$\gamma$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$A^{(4)}$</th>
<th>$R^{(3)}_{\text{int}}(-\infty)$</th>
<th>$\eta^{(3)}_2$</th>
<th>$D$</th>
<th>$\lambda_{\text{Min}}^M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.43586652150845899941601945</td>
<td>2$\gamma$</td>
<td>0.5527</td>
<td>0.03692</td>
<td>-0.7321</td>
<td>0.1104</td>
<td>1.00</td>
<td>-0.8591</td>
</tr>
<tr>
<td>0.5480175482786778827013568</td>
<td>$\gamma$</td>
<td>0.4460</td>
<td>0.06742</td>
<td>-0.7792</td>
<td>0.02743</td>
<td>4.14</td>
<td>-26.67</td>
</tr>
<tr>
<td>0.61099606113865971779911435</td>
<td>2$\gamma$/3</td>
<td>0.3579</td>
<td>0.1029</td>
<td>-0.8284</td>
<td>0.01126</td>
<td>13.14</td>
<td>-302.3</td>
</tr>
<tr>
<td>0.69150898244044165522135159</td>
<td>$\gamma$/3</td>
<td>0.2170</td>
<td>0.1723</td>
<td>-0.9015</td>
<td>0.002041</td>
<td>102.0</td>
<td>-21850</td>
</tr>
</tbody>
</table>

5.1.3 SDIRK

Three-stage, stiffly-accurate, SDIRK methods allow $(s^2 - s + 2)/2 = 4$ DOF and take the form

$$
\begin{array}{c|ccc}
\gamma & \gamma & 0 & 0 \\
\gamma & (c_2 - \gamma) & \gamma & 0 \\
1 & (1 - b_2 - \gamma) & b_2 & \gamma \\
1 & (1 - b_2 - \gamma) & b_2 & \gamma \\
\end{array}
$$

Solving the four order conditions necessary to achieve third-order accuracy yields a three-method family of schemes given by

$$
\begin{align*}
& b_1 = \frac{-1 + 4\gamma}{4\beta}, \quad b_2 = \frac{-3\alpha^2}{4\beta}, \quad c_2 = \frac{2 - 9\gamma + 6\gamma^2}{3\alpha}, \\
& \beta = (1 - 4\gamma + 2\gamma^2) \quad \text{and} \quad \alpha = (-1 + 6\gamma - 9\gamma^2 + 3\gamma^3). 
\end{align*}
$$

From the three values of $\gamma$, $\gamma = 0.43586652150845899941601945$, only one provides L-stability, with $A^{(4)} = 0.02970$, $D = 1.209$, $b_{\text{Min}}^1 = -0.6444$ and $\lambda_{\text{Min}}^M = -1.353$. This method is given by Alexander [5] and Crouzeix and Raviart [123]. Androulakis et al. [23] point out that minimal $A^{(4)}$ corresponds to a different value of $\gamma$, but the resulting scheme is not A-stable. This same scheme is also given by Butcher [69]. Nørsett and Thomsen [312] use the three-stage SDIRK structure without the stiffly-accurate assumption and create a 2(3) pair, NT I with $\gamma = 5/6$, that has an algebraically stable third-order method and an A-stable second-order method. Method NT I has $A^{(3)} = 0.02795$, $A^{(4)} = 0.1244$, $R(-\infty) = -17/25$, and $R(-\infty) = -91/125$. Coroian [117,118] offers several second- and third-order methods having a three-stage SDIRK structure. Ababneh and Ahmad [2] construct a three-stage, third-order SDIRK with $\gamma = 5/4$. Similar to his ES33b method, Skvortsov [398] accepts $L(\alpha)$-stability with the goal to reduce $A^{(4)}$ with his method S33b where $\alpha = 75.6$ degrees.
5.2 Fourth-Order Methods

5.2.1 ESDIRK

Stage-order two, fourth-order ESDIRK methods in four stages

\[
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
2\gamma & \gamma & \gamma & 0 & 0 \\
c_3 & (c_3 - a_{32} - \gamma) & a_{32} & \gamma & 0 \\
c_4 & (c_4 - a_{42} - a_{43} - \gamma) & a_{42} & a_{43} & \gamma \\
b_i & (1 - b_2 - b_3 - \gamma) & b_2 & b_3 & b_4 \\
\end{array}
\]

(231)

that are at least A-stable constitute a two-parameter family of non-stiffly-accurate methods in, e.g., \(c_3\) and \(c_4\) where one must solve \(\tau_1^{1,2,3,4} = p_4 = q_{2,3,4}^{(2)} = \tau_3^{(4)} = 0\). Solving \(p_4 = (1 - 12\gamma + 36\gamma^2 - 24\gamma^3) = 0\) and requiring I-stability gives \(\gamma = (3 + 2\sqrt{3}\cos\left(\frac{\pi}{18}\right)) / 6 \approx 1.069\). Taking the derivatives of \(A^{(5)}\) with respect to \(c_3\) and \(c_4\), the minimum error occurs in the immediate vicinity of \(c_3 = 5/7\) and \(c_4 = 4/7\), but the error is still a rather unimpressive \(A^{(5)} = 0.2328\). At this same point, \(R_{\text{int}}(-\infty) = \{-1, -1, -0.1135, -0.3543\}, R(-\infty) = -0.6304, b > 0\) and the four eigenvalues of \(M\) are \(\{1.353, -0.06234, 0.02493, -0.003735\}\). Cooper and Sayfy [116], Crouzeix and Raviart [123] and Shintani [384] also derive methods of this class.

5.2.2 SDIRK

The three-stage, fourth-order, SDIRK family of methods

\[
\begin{array}{c|cccc}
\gamma & \gamma & 0 & 0 \\
c_2 & (c_2 - \gamma) & \gamma & 0 \\
c_3 & (c_3 - a_{32} - a_{32}) & a_{32} & \gamma \\
b_i & (1 - b_2 - b_3 - \gamma) & b_2 & b_3 \\
\end{array}
\]

(232)

has three members, only one of which is A-stable; \(\gamma = (3 + 2\sqrt{3}\cos\left(\frac{\pi}{18}\right)) / 6 \approx 1.069\). Both Nørsett [308] and Crouzeix [120] discovered this method, (50), where \(A^{(5)} = 0.2570, R(-\infty) = -0.6304,\) and the eigenvalues of \(M\) are \(\{1.553, 0.06234, 0.02493, -0.003735\}\). Cooper and Sayfy [116], Crouzeix and Raviart [123] and Shintani [384] also derive methods of this class.

5.3 Fifth-Order Methods

5.3.1 ESDIRK

Al-Rabeh [18] has designed fifth-order ESDIRK methods in four-stages by using assumptions \(C(2), D(1)\) and the structure given in (231). Unfortunately, \(\deg P(z) = 1 + \deg Q(z)\), making the methods highly inappropriate for stiff problems.
methods exist within this class corresponding to the roots of \(-1+15\gamma-60\gamma^2+60\gamma^3=0\): \(\gamma = 0.659027622374092215178380771\), \(\gamma = 0.231933685553030572496784561\) and \(\gamma = 0.109039009072877212324834668\) where \(c_2 = 2\gamma\), \(c_3 = (3-25\gamma+40\gamma^2)/(5(2\gamma-1)(6\gamma-1))\) and \(c_4 = 1-\gamma\). To complete the solution, one enforces \(B(4)\) and solves for \(b\), solves \(C(2,i)\), \(i = 3, 4\) for \(a_{32}\) and \(a_{42}\) and then solves \(D(1,1)\) for \(a_{43}\).

The intermediate root possesses internal stability in the far LHP and has \(\gamma = 0\).

Stiffly accurate, stage-order two ESDIRK methods in five stages

\[
\begin{array}{ccccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
2\gamma & \gamma & \gamma & 0 & 0 & 0 \\
c_3 & (c_3-a_{32}-\gamma) & a_{32} & \gamma & 0 & 0 \\
c_4 & (c_4-a_{42}-a_{43}-\gamma) & a_{42} & a_{43} & \gamma & 0 \\
1 & (1-b_2-b_3-b_4-\gamma) & b_2 & b_3 & b_4 & \gamma \\
\frac{b_i}{(1-b_2-b_3-b_4-\gamma)} & b_2 & b_3 & b_4 & \gamma \\
\frac{b_i}{(1-b_2-b_3-b_4-b_5)} & b_2 & b_3 & b_4 & b_5
\end{array}
\]

allow 11 DOF, or 9 after \(C(2,2)\) and \(\tau_1^{(1)}\) have been applied. Although it is possible to achieve fourth-order accuracy in a stage-order two, stiffly-accurate, L-stable ESDIRK, the resulting \(\gamma\) and, consequently, \(c_2\), are rather large. This may adversely affect the convergence of the iterative method used to solve for the stage values. It also forces \(c_2 = 2\gamma\) out of the integration interval. To design a third-order method from this class of methods, seven conditions, \(\tau_1^{(1,2,3)} = p_4 = d_2^{(2,3,4)} = 0\), must be satisfied. This leaves four DOF for purposes other than fourth-order accuracy, e.g., \(\gamma\), \(c_3\), \(c_4\) and \(a_{43}\). From the general solution, one finds \(E_8 = \gamma^8\), \(E_6 = (1-12\gamma+36\gamma^2-12\gamma^3)(-1+12\gamma-36\gamma^2+36\gamma^3)/36\) and \(E_4 = (1-16\gamma+72\gamma^2-96\gamma^3+24\gamma^4)/12\). We do not consider the many special cases to this solution. Third-order, L-stable methods may be found for specific values of \(\gamma\) determined from the polynomials \(E_6^2 - 4E_8E_4 = 0\) and \(E_4 = 0\). The smallest of the four real roots of the first polynomial constitutes the minimum value of \(\gamma\) for an L-stable, stiffly-accurate method. The largest value that \(\gamma\) may take for an L-stable, stiffly-accurate method is the largest value of \(\gamma\) that keeps \(E_4\) non-negative. To 25 digits one may determine, as given by Hairer and Wanner, [193]

6 Four- and Five-stage Methods (\(S_1 = 4\))

6.1 Third-Order Methods

6.1.1 ESDIRK

Stiffly accurate, stage-order two ESDIRK methods in five stages
0.2236478009341764510696898 ≤ γ ≤ 0.5728160624821348554080014. Internal stability of the third- and fourth-stages are given by

\begin{align}
R^{(3)}_{\text{int}}(-\infty) &= \frac{c_3^2 - 4c_3\gamma + 2\gamma^2}{2\gamma^2} \\
R^{(4)}_{\text{int}}(-\infty) &= \frac{\gamma(c_4^2 - 4c_4\gamma + 2\gamma^2) - (a_{43}c_3(c_3 - 2\gamma))}{2\gamma^3}
\end{align}

(234) (235)

The first vanishes when \( c_3 = \gamma(2 \pm \sqrt{2}) \), and the second vanishes when \( a_{43} = \gamma(c_4^2 - 4c_4\gamma + 2\gamma^2) \). Several different approaches can lead to useful methods. With external error-control, one could take advantage of the low truncation error of the second stage relative to the second stage of the 3(2) from the previous section. Enforcing \( q^{(3)}_{3,4} = 0 \) and \( R^{(4)}_{\text{int}}(-\infty) = 0 \), while minimizing \( \gamma \) with \( \gamma = 9/40 = 0.225 \), leads to three methods with a SOV of \{3, 2, 3, 3, 3\}, \( c_3 = (3 - \sqrt{3})\gamma \) and \( R^{(3)}_{\text{int}}(-\infty) = -0.7321 \) and three with \( c_3 = (3 + \sqrt{3})\gamma \) and \( R^{(3)}_{\text{int}}(-\infty) = 2.7321 \). Amongst the solutions using \( c_3 = (3 - \sqrt{3})\gamma \), the two better solutions have \( c_4 = \frac{41531592441}{25872126399072} \) and \( A^{(4)} = 0.01283 \) and \( A_4 = \frac{110884995777}{118536574556} \) and \( A^{(4)} = 0.002399 \). Order conditions for the externally embedded scheme are \( \tau^{(1,2)}_1 = \hat{p}_5 = \hat{p}_4 = 0 \) and \( \tau^{(3)}_1 = 1/600 \). Instead of maximizing the elements of the stage-order vector, one could choose to simply have \( R^{(3)}_{\text{int}}(-\infty) = R^{(4)}_{\text{int}}(-\infty) = 0 \) and find the solution having the minimum \( A^{(4)} \). This occurs for \( \gamma = 9/40 = 0.225 \) at \( c_3 = 9(2 + \sqrt{2})/40 \) and approximately \( c_3 = 3/5 \) where \( A^{(4)} = 0.0007769 \); using the embedded method just described, \( A^{(3)} = 0.002357 \), \( B^{(4)} = 1.966 \), \( C^{(4)} = 1.095 \) and \( E^{(4)} = 2.134 \). The actual minima of the system is \( A^{(4)} = 0.0007666 \) at \( \gamma = 9/40 \) where the method is I-stable on all internal stages and, hence, A-stable on stage 2 and L-stable on stages 3, 4 and 5. This scheme, ESDIRK3(2)5L[2]SA, appears in Table 10. Note that \( a_{i1} = a_{i2}, i = 1, 2, \ldots, s \). Another option is to use L-stable internal error-control by setting \( c_4 = 1 \) and \( \hat{R}(-\infty) = R^{(4)}_{\text{int}}(-\infty) = 0 \). The resulting four-stage, second-order internally embedded method is I-stable only if both \( \hat{E}_6 = \gamma^6 \) and \( \hat{E}_4 = (-1 + 12\gamma - 48\gamma^2 + 72\gamma^3 - 24\gamma^4)/4 \) are non-negative. The former simply requires \( \gamma \geq 0 \), while the latter additionally requires \( 0.1804253064293985641345831 \leq \gamma \leq 2.185600973550400826291400 \). Hence, the acceptable values of \( \gamma \) for the main method are a subset of those that are acceptable for an L-stable, internally embedded method. By selecting \( \gamma = 9/40 = 0.225 \), both main and internal embedded methods are L-stable. For the final degree of freedom, \( c_3 \), enforcing \( R^{(3)}_{\text{int}}(-\infty) = 0 \) gives \( c_3 = 9(2 + \sqrt{2})/40 \) from which one selects \( c_3 = 9(2 - \sqrt{2})/40 \) to find \( A^{(4)} = 0.06446 \). Relaxing the stability constraint on stage three, one sets \( c_3 = 4/5 \), which gives a better method with \( A^{(4)} = 0.01149 \) and \( R^{(3)}_{\text{int}}(-\infty) = 17/81 \). As stages 2, 3 and 4 are I-stable, stage 2 is strictly A-stable, stage 3 is strongly A-stable and stage 4 is L-stable. Order conditions for the external controller are the same as those given previously, except the last equation that is changed to \( \tau^{(3)}_1 = 1/180 \). For the internally embedded method, \( B^{(5)} = 1.388 \), \( C^{(5)} = 1.613 \) and \( E^{(5)} = 0.2678 \). The externally embedded method has \( B^{(5)} = 1.216 \) and \( E^{(5)} = 0.4877 \). The scheme, ESDIRK3(2)5L[2]SA, is presented in Table 11. If one insists on keeping \( \gamma = 9/40 \),
b_4 < 0 and the conditional contractivity radius is zero. Reduced storage methods may be obtained by setting b_1 = a_{41}, along with R_{int}^{(3)}(\infty) = 0, R_{int}^{(4)}(\infty) = 0. The resulting method has c_4 = \frac{3381838234979}{7256498923456} and A^{(4)} = 0.01226. Embedded method order conditions are the same as those for the increased SOV scheme described previously.

### 6.1.2 QESDIRK

Stiffly accurate, stage-order two QESDIRK methods in five stages are given by

\[
\begin{align*}
\begin{array}{c|cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{9}{20} & \frac{9}{40} & \frac{9}{40} & 0 & 0 & 0 \\
\frac{9(2 + \sqrt{2})}{40} & \frac{9(1 + \sqrt{2})}{40} & \frac{9(1 + \sqrt{2})}{40} & 0 & 0 & 0 \\
\frac{3}{5} & \frac{22 + 15\sqrt{2}}{80(1 + \sqrt{2})} & \frac{22 + 15\sqrt{2}}{80(1 + \sqrt{2})} & -\frac{7}{40(1 + \sqrt{2})} & \frac{9}{45} & 0 \\
1 & \frac{2398 + 1205\sqrt{2}}{2835(4 + 3\sqrt{2})} & \frac{2398 + 1205\sqrt{2}}{2835(4 + 3\sqrt{2})} & -\frac{2374(1 + 2\sqrt{2})}{2835(5 + 3\sqrt{2})} & \frac{5827}{7566} & 0 \\
\end{array}
\end{align*}
\]

\[
\begin{align*}
\begin{array}{c|cccccccc}
b_i & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{2398 + 1205\sqrt{2}}{2835(4 + 3\sqrt{2})} & \frac{2398 + 1205\sqrt{2}}{2835(4 + 3\sqrt{2})} & -\frac{2374(1 + 2\sqrt{2})}{2835(5 + 3\sqrt{2})} & \frac{5827}{7566} & 0 & 0 \\
\frac{2398 + 1205\sqrt{2}}{2835(4 + 3\sqrt{2})} & \frac{2398 + 1205\sqrt{2}}{2835(4 + 3\sqrt{2})} & -\frac{2374(1 + 2\sqrt{2})}{2835(5 + 3\sqrt{2})} & \frac{5827}{7566} & 0 & 0 \\
\end{array}
\end{align*}
\]

<table>
<thead>
<tr>
<th>b_i</th>
<th>4559445343783</th>
<th>4559445343783</th>
<th>-7197361914863</th>
<th>34398249</th>
<th>49563</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>237134161420893</td>
<td>237134161420893</td>
<td>25574783784720</td>
<td>42052120</td>
<td>235060</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\begin{array}{c|cccccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
c_2 & c_2/2 & c_2/2 & 0 & 0 & 0 \\
c_3 & (c_3 - a_{32} - \gamma) & a_{32} & \gamma & 0 & 0 \\
c_4 & (c_4 - a_{42} - a_{43} - \gamma) & a_{42} & a_{43} & \gamma & 0 \\
1 & (1 - b_2 - b_3 - b_4 - \gamma) & b_2 & b_3 & b_4 & b_5 \\
b_1 & (1 - b_2 - b_3 - b_4 - \gamma) & b_2 & b_3 & b_4 & \gamma \\
b_3 & (1 - b_2 - b_3 - b_4 - \gamma) & b_2 & b_3 & b_4 & b_5 \\
\end{array}
\end{align*}
\]

The first matter is to establish bounds on \gamma for L-stability by computing E_4 and E_6^2 - 4E_4E_6. The bounds are given in Table 12. One now enforces the following conditions: \tau_1^{(1,2,3)} = p_4 = q_2^{(2,3,4)} = q_3^{(3,4)} = 0 to arrive at a general solution with both \gamma and c_4 unspecified and an SOV = \{3, 2, 3, 3, 3\}. As the goal is to reduce q_2^{(3)} as much as possible, \gamma is chosen as its minimal value, and c_4 is selected to enforce R_{int}^{(4)}(\infty) = 0. The results are shown in Table 13, where it appears that only the ESDIRK is useful since any decrease in c_2 results in large increases in the leading-order error and a measure of nonlinear instability. By changing the value of \gamma to a value listed in Table 14, one could also pursue second-order damping in a QESDIRK method. As QESDIRKs do not appear promising,
Table 11. ESDIRK3(2I)5L\\[2\]SA.

| \(b\) | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| \(\frac{3}{20}\) | 0 | 0 | 0 | 0 | 0 |
| \(\frac{4}{5}\) | 0 | 0 | 0 | 0 | 0 |
| 1 | \(-5827\) | \(-5827\) | \(-5827\) | \(-5827\) | \(-5827\) |
| 1 | \(-5827\) | \(-5827\) | \(-5827\) | \(-5827\) | \(-5827\) |

Table 12. Bounds on \(\gamma\) for L-stability of five-stage, third-order QESDIRKs.

| \(\gamma\) | \(c_2\) |
| \(0.223647800934176451069690\) | \(2\gamma\) |
| \(0.26308566496196092682671\) | \(\gamma\) |
| \(0.282563032306194734249668\) | \(2\gamma/3\) |
| \(0.305947487558948408902962\) | \(\gamma/3\) |

one may create second-order damped, stiffly-accurate, ESDIRK with \(q^{(3)}_{3,4} = 0\) and \(R^{(4)}_{\text{int}} = 0\). The resulting method, ESDIRK3(2I)5L\\[2\]SA_SOD has \(c_3 = (3 - \sqrt{3})\gamma\), \(c_4 = 1538744296651/12233015540721\), \(A^{(4)} = 0.01806\), \(D = 2.268\) and a minimum value for an eigenvalue of the algebraic stability matrix of \(-9.358\).

6.1.3 SDIRK

Four-stage, stiffly-accurate, L-stable SDIRK methods contain \((s^2 - s + 2)/2 = 7\) DOF

\[
\begin{array}{ccc|ccc}
\gamma & \gamma & 0 & 0 & 0 \\
(\gamma - c_2) & \gamma & 0 & 0 \\
(\gamma - c_3 - a_{32}) & a_{32} & \gamma & 0 \\
1 & (1 - b_2 - b_3 - \gamma) & b_2 & b_3 & \gamma \\
(\gamma - b_i) & (1 - b_2 - b_3 - b_4) & b_2 & b_3 & b_4 \\
(\gamma - b_i) & (1 - b_2 - b_3 - b_4) & b_2 & b_3 & b_4 \\
\end{array}
\]

(237)

and are L-stable for \(0.2236478009341764510696898 \leq \gamma \leq 0.5728160624821348554080014\). Solving the four order conditions necessary to achieve third order yields a three-
Table 13. Properties of various stiffly-accurate, L-stable, five-stage, third-order QESDIRKs.

<table>
<thead>
<tr>
<th>γ</th>
<th>c_2</th>
<th>c_3</th>
<th>c_4</th>
<th>A^{(4)}</th>
<th>R_{int}^{(3)}(-\infty)</th>
<th>q_2^{(3)}</th>
<th>D</th>
<th>λ_{min}^M</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2250</td>
<td>2γ</td>
<td>0.2853</td>
<td>0.09355</td>
<td>0.002399</td>
<td>−0.7321</td>
<td>0.01519</td>
<td>1.997</td>
<td>−11.71</td>
</tr>
<tr>
<td>0.2250</td>
<td>2γ</td>
<td>0.2853</td>
<td>0.5162</td>
<td>0.01283</td>
<td>−0.7321</td>
<td>0.01519</td>
<td>1.000</td>
<td>−1.056</td>
</tr>
<tr>
<td>0.2250</td>
<td>2γ</td>
<td>0.2853</td>
<td>1.415</td>
<td>0.04927</td>
<td>−0.7321</td>
<td>0.01519</td>
<td>4.039</td>
<td>−7.474</td>
</tr>
<tr>
<td>0.2650</td>
<td>γ</td>
<td>0.2157</td>
<td>0.08115</td>
<td>0.01247</td>
<td>−0.7792</td>
<td>0.003102</td>
<td>7.056</td>
<td>−95.15</td>
</tr>
<tr>
<td>0.2650</td>
<td>γ</td>
<td>0.2157</td>
<td>0.4836</td>
<td>0.03059</td>
<td>−0.7792</td>
<td>0.003102</td>
<td>7.697</td>
<td>−105.9</td>
</tr>
<tr>
<td>0.2650</td>
<td>γ</td>
<td>0.2157</td>
<td>1.423</td>
<td>0.05463</td>
<td>−0.7792</td>
<td>0.003102</td>
<td>24.00</td>
<td>−243.6</td>
</tr>
<tr>
<td>0.2850</td>
<td>2γ/3</td>
<td>0.1669</td>
<td>0.06796</td>
<td>0.01945</td>
<td>−0.8284</td>
<td>0.001143</td>
<td>21.16</td>
<td>−760.0</td>
</tr>
<tr>
<td>0.2850</td>
<td>2γ/3</td>
<td>0.1669</td>
<td>0.4665</td>
<td>0.04279</td>
<td>−0.8284</td>
<td>0.001143</td>
<td>28.96</td>
<td>−1541.0</td>
</tr>
<tr>
<td>0.2850</td>
<td>2γ/3</td>
<td>0.1669</td>
<td>1.461</td>
<td>0.06576</td>
<td>−0.8284</td>
<td>0.001143</td>
<td>79.79</td>
<td>−2447.0</td>
</tr>
<tr>
<td>0.3100</td>
<td>γ/3</td>
<td>0.09730</td>
<td>0.04363</td>
<td>0.03054</td>
<td>−0.9015</td>
<td>0.0001839</td>
<td>165.3</td>
<td>−47810.0</td>
</tr>
<tr>
<td>0.3100</td>
<td>γ/3</td>
<td>0.09730</td>
<td>0.4483</td>
<td>0.06207</td>
<td>−0.9015</td>
<td>0.0001839</td>
<td>259.1</td>
<td>−127900.0</td>
</tr>
<tr>
<td>0.3100</td>
<td>γ/3</td>
<td>0.09730</td>
<td>1.523</td>
<td>0.08661</td>
<td>−0.9015</td>
<td>0.0001839</td>
<td>659.8</td>
<td>−146900.0</td>
</tr>
</tbody>
</table>

Table 14. Values for γ second-order damped, five-stage, third-order, L-stable QESDIRKs.

<table>
<thead>
<tr>
<th>γ</th>
<th>c_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.30253457818265077121644</td>
<td>2γ</td>
</tr>
<tr>
<td>0.35362860281293140867039</td>
<td>γ</td>
</tr>
<tr>
<td>0.3776348398366128533677</td>
<td>2γ/3</td>
</tr>
<tr>
<td>0.40527435197627562691063</td>
<td>γ/3</td>
</tr>
</tbody>
</table>

parameter family of methods

\[
b_2 = \frac{-(2 + 3c_3 + 9\gamma - 12c_3\gamma - 6\gamma^2 + 6c_3\gamma^2)}{(6(c_2 - c_3)(c_2 - \gamma))}
\]  
\[
b_3 = \frac{-(2 + 3c_2 + 9\gamma - 12c_2\gamma - 6\gamma^2 + 6c_2\gamma^2)}{(6(c_2 - c_3)(c_3 - \gamma))}
\]  
\[
a_{32} = \frac{-(c_2 - c_3)(c_3 - \gamma)(-1 + 9\gamma - 18\gamma^2 + 6\gamma^3)}{(c_2 - \gamma)(-2 + 3c_2 + 9\gamma - 12c_2\gamma - 6\gamma^2 + 6c_2\gamma^2)}.
\]

Keeping γ as small as possible, γ = 9/40 is chosen. These methods are a two-parameter family given in terms of c_2 and c_3. Setting \(\frac{\partial A^{(4)}}{\partial c_2} = \frac{\partial A^{(4)}}{\partial c_3} = 0\), one finds a nearby solution, SDIRK3M, having c_2 = 7/13, c_3 = 11/15 and A^{(4)} = 0.003328, where the absolute minimum error of the system is A^{(4)} = 0.003321. The other coefficients are a_{21} = 163/520, a_{31} = −6481433/8838675, a_{32} = 87795409/70709400, b_1 = 4032/9943, b_2 = 6929/15485, b_3 = −723/9272 and λ_{min}^M = −0.1128. An
embedded method should satisfy $\tau_1^{(1,2)} = \hat{p}_4 = 0$ and $\tau_1^{(3,4)} \neq 0$. A coarse grid numerical search was unable to find any algebraically stable four-stage, third-order, stiffly-accurate, L-stable SDIRK methods. However, a conditionally contractive method having $r_{F_2} = 2.279$ and $A^{(4)} = 0.01167$ is found at $c_2 = 3/4$, $c_3 = 2/5$, $a_{21} = 21/40$, $a_{31} = 49637/167100$, $a_{32} = -40789/334200$, $b_1 = 118/441$, $b_2 = 149/4410$ and $b_3 = 557/1176$. Kverne [280] derives a 3(2) pair designed for index-1 DAEs including a dense-output method. She sets $\gamma = 1/4$. Ten years later, Cameron et al. [91] derive the identical method, but with a different embedded scheme. Nørsett and Thomsen [312] generate a 3(4) pair, NT II, by using a four-stage, nonstiffly-accurate SDIRK with $\gamma = 5/6$ and an A-stable main method. It has $A^{(4)} = 0.1212$, $\hat{A}^{(5)} = 0.5121$, $R(-\infty) = -91/125$, $\hat{R}(-\infty) = -341/625$ and the main method is nearly algebraically stable, but both are A-stable. Al-Rabeh [17] also constructs a 3(4) pair (his expressions i) and iii) with typos on $b_1$ with $\gamma = 0.43586652150854989941601945$, $A^{(4)} = 0.03244$, $\hat{A}^{(5)} = 0.01479$, $R(-\infty) = 0$ (L-stable) and $\hat{R}(-\infty) = 0.7175$ (A-stable). Cameron [89] derives two 2(3) pairs where the embedded methods are stiffly-accurate, and both methods are L-stable, one with $\gamma = 11/40$ and the other with $\gamma = 0.35322850954885$. Coroian [119] derives several A- and L-stable third-order SDIRKs in four-stages.

6.2 Fourth-Order Methods

6.2.1 ESDIRK

Constraining the stiffly-accurate method to have a stage-order of two, an overall order of four and to be L-stable, one enforces nine conditions: $\tau_1^{(1,2,3,4)} = p_4 = q_{2,3,4}^{(2)} = 0$, leaving a two-parameter family of methods in, e.g., $c_3$ and $c_4$ following the structure of (233). The general solution is given by

\[
\begin{align*}
b_2 &= \frac{3 - 12\gamma + 4c_4(-1 + 3\gamma) - 2c_3(2 - 6\gamma + c_4(-3 + 6\gamma))}{24\gamma(2\gamma - c_3)(2\gamma - c_4)}, \\
b_3 &= \frac{\phi_2 - 4c_4\phi_1}{12c_3(c_3 - c_4)(c_3 - 2\gamma)}, \\
b_4 &= \frac{\phi_2 - 4c_3\phi_1}{12c_4(c_4 - c_3)(c_4 - 2\gamma)}, \\
a_{32} &= \frac{c_3(c_3 - 2\gamma)}{4\gamma}, \\
a_{42} &= \frac{(c_4(4\gamma - 2\gamma)(-4c_3^2\phi_1 - 2\gamma\phi_2 + c_3\phi_3 + 2c_4\phi_4))}{(4\gamma(2\gamma - c_3)(4c_3\phi_1 - \phi_2))}, \\
a_{43} &= \frac{(c_4(4\gamma - 2\gamma)(4c_3\phi_1 - \phi_2))}{(c_3(c_3 - 2\gamma)(4c_3\phi_1 - \phi_2))}, \\
\end{align*}
\]

with $\phi_1 = (1 - 6\gamma + 6\gamma^2)$, $\phi_2 = (3 - 20\gamma + 24\gamma^2)$, $\phi_3 = (5 - 36\gamma + 48\gamma^2)$, $\phi_4 = (-1 + 12\gamma - 36\gamma^2 + 24\gamma^3)$, $c_3 \neq 2\gamma$, $c_4 \neq 2\gamma$, $c_3 \neq c_4$, $4c_3\phi_1 \neq \phi_2$, $c_3 \neq 0$, $c_4 \neq \frac{-3 + 20\gamma - 24\gamma^2}{4(1 - 6\gamma + 6\gamma^2)}$, $c_4 \neq 0$, $E_8 = (4\gamma - 1)(1 - 12\gamma + 24\gamma^2)(1 - 16\gamma + 72\gamma^2 - 96\gamma^3 + 48\gamma^4)/576$, $E_6 = (1 - 24\gamma + 204\gamma^2 - 768\gamma^3 + 1224\gamma^4 - 576\gamma^5)/72$ and $R_{\text{int}}^{(3)}(-\infty) = (c_3^2 - 4c_3\gamma + 2\gamma^2)/(2\gamma^2)$. 84
and, one is assured an A-stable, internally embedded method. From the previous section, L-stability requires that \( \gamma \approx 0.43586652150845899941601945 \). Minimally, two additional conditions must be enforced with the two remaining DOF: \( c_4 = 1 \) and \( \hat{r}_1^{(3)} = 0 \) \((C(3, 4))\). The method, first given by Kvernø [281], has \( c_3 = \frac{2(2-9\gamma+12\gamma^2)}{1-6\gamma+12\gamma^2} \), \( R_{\text{int}}^{(3)}(-\infty) = -0.5415, \ R_{\text{int}}^{(4)}(-\infty) = -0.5525, \ A^{(5)} = 0.04506, B^{(5)} = 2.588, C^{(5)} = 3.059 \) and \( E^{(5)} = 0.5157 \). It will be referred to as ESDIRK4(3)5L[2]SA. It is also strongly A-stable on stages 2, 3 and 4. Another option for an internally embedded method is to relax the order requirement to second-order and use the extra DOF to enforce L-stability of the embedded method, as shown in the previous section. Choosing \( c_4 = 1 \) and \( c_3 = \frac{(3+\sqrt{3})\gamma}{1+18\gamma-108\gamma^2+224\gamma^3-192\gamma^4+56\gamma^5} \) gives \( R_{\text{int}}^{(3)}(-\infty) = -0.3080, R_{\text{int}}^{(4)}(-\infty) = 0, \) and \( A^{(5)} = 0.04423 \). Increasing the SOV from \( \{2, 2, 2, 4\} \) to \( \{4, 2, 3, 4\} \) is accomplished by enforcing \( C(3, 3) \) and \( C(3, 4) \). Doing this gives \( c_3 = (3 \pm \sqrt{3})\gamma; \) however, if consideration is limited to \( c_3 = (3 - \sqrt{3})\gamma \), then \( c_4 = \frac{3(3+\sqrt{3})^2 - 4(5+\sqrt{3})\gamma + 24\gamma^2}{4(3+\sqrt{3} - 3(6+\sqrt{3})\gamma + 18\gamma^2)} \), \( R_{\text{int}}^{(3)}(-\infty) = -0.7321, R_{\text{int}}^{(4)}(-\infty) = 0.7189, \) and \( A^{(5)} = 0.04201 \). Cameron [89] derives a similar general solution without using \( c_3 \) as a parameter. Storage reduction [256] may be accomplished by setting \( c_4 = 5/6 \) and \( a_{41} = b_1 \) via \( c_3 = \frac{11287892608407}{18061209278125} \) where \( R_{\text{int}}^{(3)}(-\infty) = -0.5869, R_{\text{int}}^{(4)}(-\infty) = -0.5017 \) and \( A^{(5)} = 0.04043 \). Lastly, one might wish to maximize the contractivity radius. A computer search shows that it is likely impossible to simultaneously have all positive values of \( b_i \), therefore, the radius of conditional contractivity is likely zero.

Third-order, external error-control is accomplished by solving \( \hat{r}_1^{(1,2,3)} = \hat{p}_5 = 0 \), and \( \hat{R}(-\infty) = \hat{p}_4/\gamma^4 = -\gamma/3 \), yielding

\[
\begin{align*}
\hat{b}_5 &= \frac{4\gamma^2(-1 + 9\gamma - 18\gamma^2 + 6\gamma^3 - 2\gamma^4)}{-1 + 12\gamma - 36\gamma^2 + 24\gamma^3} \\
\hat{b}_4 &= \frac{-\gamma \phi_1(-3 + 4c_3 + 20\gamma - 24c_3\gamma - 24\gamma^2 + 24c_3\gamma^3)}{3c_4(c_4 - c_3)(c_4 - 2\gamma)(-1 + 12\gamma - 36\gamma^2 + 24\gamma^3)^2}
\end{align*}
\]
\[ \dot{b}_3 = \frac{1 - 3b_5 - 3b_4 c_2^2 - 3\gamma + 6b_5\gamma + 6b_4 c_4\gamma}{3c_3(c_3 - 2\gamma)} \] (249)

\[ \dot{b}_2 = \frac{2 - 6b_5 - 3c_3 + 6b_5c_3 + 6b_4 c_3 c_4 - 6b_4 c_4^2}{12\gamma(2\gamma - c_3)} \] (250)

with \( \phi_1 = (2 - 43\gamma + 336\gamma^2 - 1194\gamma^3 + 1966\gamma^4 - 1336\gamma^5 + 168\gamma^6 + 96\gamma^7) \) and yielding \( E_{10} = 0, \ E_8 = -\gamma^8(\gamma - 3)(3 + \gamma)/9 > 0, \ E_6 = (-1 + 24\gamma - 216\gamma^2 + 912\gamma^3 - 1872\gamma^4 + 1716\gamma^5 - 336\gamma^6 - 144\gamma^7)/36 > 0 \) and \( E_4 = (1 - 16\gamma + 72\gamma^2 - 96\gamma^3 + 24\gamma^4 + 8\gamma^5)/12 > 0 \). Therefore, the exterior embedded method is I-stable and hence A-stable but not L-stable. ESDIRK4(3)L[2]SA, with \( c_3 = 1/2 \) and \( c_4 = 3/4 \), has \( B^{(5)} = 1.767, \ C^{(5)} = 2.744 \) and \( E^{(5)} = 1.324 \). For the externally embedded method of ESDIRK4(3)L[2]SA, one has \( A^{(4)} = 0.03019, \ B^{(5)} = 1.766, \ C^{(5)} = 3.059 \) and \( E^{(5)} = 1.492 \). Skvortsov [396,398] also constructs a stiffly-accurate, stage-order two ESDIRK but chooses \( \gamma \approx 0.2204 \) so that the method is only \( L(\alpha) \)-stable.

### 6.2.2 QESDIRK

The primary shortcoming of the five-stage, fourth-order ESDIRK is the large value of \( \gamma \) and consequently \( c_2 \). By allowing \( a_{22} \neq a_{33} = a_{44} = a_{55} \) with a QESDIRK and retaining the same order conditions as the SOV = \{4, 2, 3, 4\} ESDIRK method described previously, one may also reduce the value of \( q_2^{(3)} \) to mimic stage-order three behavior at relatively lax tolerances. The Butcher array is that found in (236), and solutions use the smaller of the values for \( c_3 \) derived from a quadratic solve. If \( c_2 = 2\gamma \), then the ESDIRK is retrieved. Table 15 lists three possible QESDIRKs. Note that the values of \( \gamma \) come from the upper limits listed in Table 12. As with the four-stage QESDIRK, only the case where \( c_2 = \gamma \) seems worthy of any further consideration; hence, we give the three nontrivial abscissae: \( c_2 = 10393270312171/14930336774230, c_3 = 12329209447232/21762223217049 \), and \( c_4 = 2190473621641/2291448330983 \).

<table>
<thead>
<tr>
<th>( \gamma )</th>
<th>( c_2 )</th>
<th>( c_3 )</th>
<th>( c_4 )</th>
<th>( A^{(5)} )</th>
<th>( R_{\text{int}}^{(3)}(\infty) )</th>
<th>( R_{\text{int}}^{(4)}(\infty) )</th>
<th>( q_2^{(3)} )</th>
<th>( D )</th>
<th>( \lambda_{\text{min}}^{M} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5728</td>
<td>2( \gamma )</td>
<td>0.7263</td>
<td>0.7979</td>
<td>0.04201</td>
<td>-0.7321</td>
<td>-0.7189</td>
<td>0.2506</td>
<td>3.490</td>
<td>-22.78</td>
</tr>
<tr>
<td>0.6961</td>
<td>( \gamma )</td>
<td>0.5665</td>
<td>0.9559</td>
<td>0.05226</td>
<td>-0.7792</td>
<td>-0.6417</td>
<td>0.05662</td>
<td>2.721</td>
<td>-2.824</td>
</tr>
<tr>
<td>0.7787</td>
<td>2( \gamma/3 )</td>
<td>0.4562</td>
<td>1.023</td>
<td>0.08263</td>
<td>-0.8284</td>
<td>-0.6618</td>
<td>0.02332</td>
<td>11.00</td>
<td>-8.330</td>
</tr>
<tr>
<td>0.9000</td>
<td>( \gamma/3 )</td>
<td>0.2825</td>
<td>1.080</td>
<td>0.1677</td>
<td>-0.9015</td>
<td>-0.9685</td>
<td>0.004500</td>
<td>103.6</td>
<td>-75.50</td>
</tr>
</tbody>
</table>

### 6.2.3 SDIRK

Attempting a fourth-order, four-stage, stiffly-accurate, L-stable SDIRK, the exact solution for the third-order method is used. Upon solving \( \tau_{1,2}^{(4)} = 0 \), one may express both \( \tau_3^{(4)} \) and \( \tau_4^{(4)} \) as polynomial functions of \( \gamma \) alone. The first has five real roots
while the second has four. There are no common roots; hence, there are no fourth-order solutions to this scheme, as proven by Alexander [5].

Relenting on the desire for stiffly-accurate methods, one may derive a two-parameter family \( (c_3\) and \( c_4\) of L-stable, fourth-order methods with \( c_2 = (3\gamma(-1+2\gamma)^3)/(1-6\gamma+12\gamma^2) \) and a non-stiffly-accurate analog of (237), implying that \( \gamma = 0.5728160624821348554080014 \) and \( c_2 \approx 0.02429 \). This method is derived by solving \( \tau_1^{(1,2,3,4)} \) for \( b \) and \( \tau_2^{(3)} = \tau_2^{(4)} = 0 \) for \( a_{32}, a_{42}, a_{43} \) and \( c_2 \). The minimal error solution has \( A(5) = 0.03202 \) and \( D = 3.469 \), but also has a very large negative eigenvalue of the algebraic stability matrix: \( -101.3 \). This situation occurs in the immediate vicinity of \( c_3 = 11/20 \) and \( c_4 = 7/9 \). A more useful solution might be at \( c_3 = 5/7 \) and \( c_4 = 1/4 \) with \( A(5) = 0.05008 \) and \( D = 0.7143 \). Nørsett [308] has investigated this family of methods. In two papers, Jawais et al. [241, 242] study four-stage SDIRKs applied to linear ODEs. These methods are fourth-order on linear ODEs but only third-order on nonlinear ODEs. Coroian [119] derives an A- and an L-stable fourth-order SDIRK in four-stages.

### 6.3 Fifth-Order Methods

#### 6.3.1 ESDIRK

Before attempting to construct a fifth-order ESDIRK in five-stages, one may look at Table 4 to see that not even A-stable methods are possible. A-stability requires that \( p_5 = (1-20\gamma+120\gamma^2-240\gamma^3+120\gamma^4)/120 = 0 \). Solving \( p_5 = 0 \) produces four roots: \( \gamma = 0.09129173346525076086032944, \gamma = 0.1744841756515234920453601, \gamma = 0.3888576711028921132337183, \) and \( \gamma = 1.345366419780333633860592 \); however, none of these values give rise to a positive semidefinite E-polynomial. Hence, none of the methods can be recommended for general stiff equations. However, if one were to be tested, the \( \gamma \approx 0.3889 \) root might work best where \( R(-\infty) = 1.038 \). The form of the method is (233) without the stiffly-accurate assumption. One may construct this family of methods by solving \( \tau_1^{(1,2,3,4,5)} = q_3^{(2)} = \tau_3 = \tau_4 = 0 \). At \( \gamma \approx 0.3889, \ c_4 = 17/27 \) and \( c_5 = 13/16, A(6) = 0.002758 \) and the largest internal stability value in the far LHP is \(-1.755 \) on stage four. Minimum \( A(6) \) for each of these roots, in ascending magnitude of \( \gamma \), is \( A(6) = 0.0004670, 0.0008287, 0.002758, 0.9835 \). As is commonly observed, smaller values of \( \gamma \) generally correspond to smaller values of the leading order error. Kraaijevanger [266] constructs a fifth-order EDIRK with \( a_{55} = 0 \) and \( A(6) = 0.0002604, \) in the interest of studying nonlinear contractivity, but the method has \( R_{\text{int}}(-\infty) = \{1, -1, 9, -8, \infty\} \).

### 6.4 Sixth-Order Methods

#### 6.4.1 EDIRK

Al-Rabeh [18] has computed a sixth-order EDIRK in five-stages, four of which are implicit. It is impossible to make this family of methods A-stable since \( |R(z)| \to \infty \) as \( z \to -\infty \) because \( \deg Q(z) = \deg P(z) + 1 \). Al-Rabeh uses assumptions \( C(2) \) and \( D(2) \) to derive a method with \( a_{33} < 0 \); however, the method can also be
derived by using only $C(2)$ and $D(1)$ and solving $	au_1^{(1,2,3,4,5,6)} = q_2^{(2)} = r_1^{(1)} = \tau_4^{(5)} = \tau_6^{(6)} = 0$. From this two-parameter family of methods, one may set $a_{22} = a_{33} = a_{44} = \gamma$, where $\gamma > 0$ is one of the roots of $-1 + 18\gamma - 90\gamma^2 + 120\gamma^3 = 0$; $\gamma = 0.09484726491451296849036080$, $\gamma = 0.1881284503958047052442961$ and $\gamma = 0.4670242846896823262633431$

Note the symmetry of $b$ and $c$ (See §(2.8)). Other coefficients are given by $b_1 = (1 - 20\gamma + 40\gamma^2)/(120\gamma(-1 + 2\gamma))$, $b_2 = -1(120\gamma(1-2\gamma)(-1+4\gamma)^2)^{-1}$, $b_3 = (8(1-10\gamma + 20\gamma^2))/((15(-1+4\gamma)^2)\ a_{32} = (1-4\gamma)/(16\gamma)\ a_{42} = -((1+2\gamma)(-1+4\gamma)(3-40\gamma+80\gamma^2))/(4\gamma)\ a_{43} = 4(-1+2\gamma)(-1+4\gamma)(1-10\gamma+20\gamma^2), a_{52} = (1-10\gamma+20\gamma^2)(-3+40\gamma-128\gamma^2+128\gamma^3)/(4\gamma(-1+4\gamma)^2(1-20\gamma+40\gamma^2)), a_{53} = (-4(-1+2\gamma)(1-12\gamma+16\gamma^2)(1-10\gamma+20\gamma^2))/(((-1+4\gamma)^2(1-20\gamma+40\gamma^2))$ and $a_{54} = \gamma((-1+4\gamma)^2(1-20\gamma+40\gamma^2))^{-1}$. For $\gamma \approx 0.47$, $A^{(7)} = 0.003793$ and internal stability for far LHP eigenvalues is $R_{int}(-\infty) = \{1, -1, -0.5681, 0.5587, -\infty\}$, for $\gamma \approx 0.19; A^{(7)} = 0.0002288$ and $R_{int}(-\infty) = \{1, -1, -0.7837, 0.3629, -\infty\}$; and for $\gamma \approx 0.095, A^{(7)} = 0.00004368$ and $R_{int}(-\infty) = \{1, -1, 4.352, -21.92, -\infty\}$.

# 7 Five- and Six-stage Methods ($S_l = 5$)

## 7.1 Fourth-Order Methods

### 7.1.1 ESDIRK

Six-stage, stage-order two, stiffly-accurate ESDIRK methods

<table>
<thead>
<tr>
<th>$b_i$</th>
<th>$b_1$</th>
<th>$b_2$</th>
<th>$b_3$</th>
<th>$b_4$</th>
<th>$b_5$</th>
<th>$b_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2\gamma</td>
<td>0</td>
<td>\gamma</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$c_3$</td>
<td>$c_3 - a_{32} - \gamma$</td>
<td>$a_{32}$</td>
<td>\gamma</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$c_4$</td>
<td>$c_4 - a_{42} - a_{43} - \gamma$</td>
<td>$a_{42}$</td>
<td>$a_{43}$</td>
<td>\gamma</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$c_5$</td>
<td>$c_5 - a_{52} - a_{53} - a_{54} - \gamma$</td>
<td>$a_{52}$</td>
<td>$a_{53}$</td>
<td>$a_{54}$</td>
<td>\gamma</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$1 - b_2 - b_3 - b_4 - b_5 - \gamma$</td>
<td>$b_2$</td>
<td>$b_3$</td>
<td>$b_4$</td>
<td>$b_5$</td>
<td>\gamma</td>
</tr>
</tbody>
</table>

provide 16 DOF. Two of these are quickly dispensed with via $C(2,2)$ and $\tau_1^{(1)}$. L-stable methods may be found for a range $\gamma$ values determined from the zeros of the polynomials $E_2^2 - 4E_{10}E_6 = 0$, and $E_6 = 0$ where $E_{10} = \gamma^{10}$, $E_8 = (-1 + 40\gamma - 640\gamma^2 + 5280\gamma^3 - 24240\gamma^4 + 62400\gamma^5 - 86400\gamma^6 + 57600\gamma^7 - 11520\gamma^8)/576$, $E_6 = \ldots$
(1 – 30γ + 330γ^2 – 1680γ^3 + 3960γ^4 – 3600γ^5 + 720γ^6)/72. The first polynomial has 12 complex roots and four real roots. The smallest of the real roots constitutes the minimum value of γ for a six-stage, fourth-order, L-stable, stiffly-accurate method. The largest value that γ may take for these methods is the largest value of γ that keeps E_6 non-negative. It has two complex roots and four real roots. To 25 digits, one may determine, as have Hairer and Wanner [193], 0.2479946362127474551679910 ≤ γ ≤ 0.6760423932262813288723863. As smaller values of γ are generally more desirable, one can use the value γ = 1/4 for simplicity. Methods may be designed by using either C(2) or a truncated version of C(3). Using C(2), one solves at least τ_1(1,2,3,4) = p_5 = q_2(2) = τ_3(4) = 0, which leaves six DOF to optimize the method based on other considerations. Choosing a truncated C(3), one enforces at least τ_1(1,2,3,4) = p_5 = b_2 = q_2(2) = q_3(3) = τ_3(4) = 0, leaving three DOF for optimization. With such a large number of residual DOF, there are many optimization strategies. Internal stability and leading-order error are the primary concerns. Using C(2) and enforcing R^{(5)}_{int}(−∞) = 0 at stages i = 3, 4, 5, the system may be reduced to one depending on c_4, c_5 and γ = 1/4. Satisfying R^{(3)}_{int} = 0 requires c_3 = γ(2 ± √2). Solving ∂A^{(5)}/∂c_4 = 0 and ∂A^{(5)}/∂c_5 = 0 for each case, one finds c_4 ≈ 212/127, c_5 ≈ 357/121 and A^{(5)} = 0.001778 for c_3 = γ(2 – √2), and c_4 ≈ 71/686, c_5 ≈ 130/199 and A^{(5)} = 0.002331 for c_3 = γ(2 + √2). A third-order, L-stable, externally embedded scheme is added to each of these methods by solving τ_1(1,2,3,4) = p_6 = p_5 = 0, and τ_3(4) = ±1/1000, where the c_3 = γ(2 – √2) solution uses τ_1(4) = +1/1000, and the c_3 = γ(2 + √2) solution uses τ_1(4) = −1/1000. For c_3 = γ(2 – √2), one sets c_4 = 5/8, and c_5 = 26/25 to get A^{(5)} = 0.001830, D = 1.585, B^{(5)} = 1.297, C^{(5)} = 1.151, E^{(5)} = 0.5744, Eig(M) = {0.1978, −0.1971, 0.06250, −0.02687, −0.006706, 0.001393} and the smallest value of b_5 = −0.1083. The method is I-stable on all internal stages and therefore A-stable on stage 2 and L-stable on stages 3, 4, 5 and 6. This nonconfluent method, ESDIRK4(3)6L[2]SA, is listed in Table 16.

If internal error-control in a 4(3) pair is desired, then one may set c_3 = γ(2 – √2), c_5 = 1 and solve τ_1(3) = 0 to find c_4 = (8 – √2 + √186 + 64√2)/40 and A^{(5)} = 0.002254. L-stability of the five-stage, third-order, internal method requires, from before, that 0.2236478009341764510696898 ≤ γ ≤ 0.5728160624821348554080014. Hence, the lower bound of γ is set by the fourth-order main method, not the internally embedded third-order method. As R^{(5)}_{int}(−∞) = 0, the internally embedded method is L-stable. An externally embedded method may be added to this scheme by solving τ_1(1,2,3) = p_6 = p_5 = 0, τ_1(4) = +1/1000. The actual coefficients of ESDIRK4(3)6L[2]SA are given in Table 17. Other characterizations of this confluent method include D = 1.359, B^{(5)} = 1.161, C^{(5)} = 1.095 and E^{(5)} = 0.08703 for the internal method, B^{(5)} = 1.679, C^{(5)} = 1.095 and E^{(5)} = 0.6964 for the external method, Eig(M) = {0.0625, −0.2426, −0.05379, −0.01158, 0.01475, 0.3132}, and the smallest value of b_5 = 0.06251. Like ESDIRK4(3)6L[2]SA, the method is I-stable on all internal stages and, hence, A-stable on stage 2 and L-stable on stages 3, 4, 5 and 6. Again, an interesting feature of this class of schemes is that a_{i1} = a_{i2}. Another possibility within this family of methods is a reduced storage
Table 16. ESDIRK4(3)6L[2]SA with $a_{i1} = a_{i2}$.

<table>
<thead>
<tr>
<th></th>
<th>$0$</th>
<th>$0$</th>
<th>$0$</th>
<th>$0$</th>
<th>$0$</th>
<th>$0$</th>
<th>$0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0$</td>
<td>$a_{21}$</td>
<td>$\frac{1}{4}$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>$a_{31}$</td>
<td>$\frac{1 - \sqrt{2}}{8}$</td>
<td>$\frac{1}{4}$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
</tr>
<tr>
<td>$\frac{2 - \sqrt{2}}{4}$</td>
<td>$a_{41}$</td>
<td>$\frac{5 - 7\sqrt{2}}{64}$</td>
<td>$\frac{7(1 + \sqrt{2})}{32}$</td>
<td>$\frac{1}{4}$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
</tr>
<tr>
<td>$\frac{5}{8}$</td>
<td>$a_{51}$</td>
<td>$-\frac{13796 - 54539\sqrt{2}}{128000}$</td>
<td>$\frac{506605 + 132109\sqrt{2}}{43700}$</td>
<td>$\frac{166(-97 + 376\sqrt{2})}{109376}$</td>
<td>$\frac{1}{4}$</td>
<td>$0$</td>
<td>$0$</td>
</tr>
<tr>
<td>$\frac{26}{25}$</td>
<td>$b_1$</td>
<td>$\frac{1181 - 987\sqrt{2}}{13782}$</td>
<td>$\frac{47(-267 + 1783\sqrt{2})}{273343}$</td>
<td>$\frac{-16(-22922 + 3525\sqrt{2})}{571953}$</td>
<td>$\frac{-15625(97 + 376\sqrt{2})}{90749876}$</td>
<td>$\frac{1}{4}$</td>
<td>$0$</td>
</tr>
<tr>
<td>$1$</td>
<td>$b_1$</td>
<td>$\frac{1181 - 987\sqrt{2}}{13782}$</td>
<td>$\frac{47(-267 + 1783\sqrt{2})}{273343}$</td>
<td>$\frac{-16(-22922 + 3525\sqrt{2})}{571953}$</td>
<td>$\frac{-15625(97 + 376\sqrt{2})}{90749876}$</td>
<td>$\frac{1}{4}$</td>
<td>$0$</td>
</tr>
<tr>
<td>$\hat{b}_1$</td>
<td>$\hat{b}_1$</td>
<td>$-\frac{480923228411}{4982971448372}$</td>
<td>$\frac{6709447293961}{1283318909569}$</td>
<td>$\frac{3513175791894}{6748737351936}$</td>
<td>$\frac{-498863281070}{6042679550617}$</td>
<td>$\frac{2077005547802}{8945017450137}$</td>
<td>$0$</td>
</tr>
</tbody>
</table>

scheme. By setting $b_1 = a_{51}$ and minimizing $A^{(5)}$, one finds a method at $c_4 = 21/20$, $c_5 = \frac{1475994502638}{15257702071139}$ and $A^{(5)} = 0.01342$. It should be noted that this error is over six times larger than the methods discussed previously. While the value of $A^{(6)}$ may impact the accuracy of the method, such an increase in $A^{(5)}$ is an indication that the new method is not useful. Principal error decreases with increasing $c_4$ and decreasing $c_5$. At $c_4 = 11/10$, $A^{(5)} = 0.01255$, while at $c_4 = 5/4$, $A^{(5)} = 0.01180$. Order conditions for the externally embedded method are the same as discussed previously with the exception of $z^{(4)}_1 = -1/300$. Methods of this general class have also been given by Butcher and Chen [72], Skvortsov [396, 397], and Kennedy and Carpenter [255] as the implicit portion of ARK4(3)6L[2]SA, all with $\gamma = 1/4$ and $q^{(2)} = 0$. Butcher and Chen’s confluent method has $R_{\text{int}}(-\infty) = \{1, -1, -1/2, 7/9, 5/12, 0\}$ and $A^{(5)} = 0.006067$. Two confluent methods by Skvortsov [397] have $R_{\text{int}}(-\infty) = \{1, -1, -1/2, 1/2, -10/31, 0\}$ and $A^{(5)} = 0.003702$ (FDIRK4b) and $R_{\text{int}}(-\infty) = \{1, -1, -7/25, 0, 13/9, 0\}$ and $A^{(5)} = 0.001997$ (FDIRK4a). Method FDIRK4b [396] includes an embedded method. In a later paper, Skvortsov [398] designs an L($\alpha$)-stable method with $\alpha = 89.95$ degrees and $A^{(5)} = 0.002297$. The method is extremely internally unstable on the fourth and fifth stages. To investigate the effects of internal instability, five new methods have been constructed that are as identical to ESDIRK4(3)6L[2]SA as possible, except for their values of $|R_{\text{int}}^{(i)}(-\infty)| = n/2$, $i = 3, 4, 5$, $n = 0, 1, \ldots, 5$. These schemes are listed in Table 18.

Solutions to the six-stage, fourth-order ESDIRK using a truncated form of $C(3)$ may also be considered. Two of the three DOF of the system may be used to enforce $R_{\text{int}}^{(i)}(-\infty) = 0$ at stages $i = 4, 5$, while the third DOF is used to
Table 17. ESDIRK4(3I)6L[2]SA with $a_{i1} = a_{i2}$.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$a_{i2}$</th>
<th>$a_{i1}$</th>
<th>$\frac{\gamma}{3}$</th>
<th>$\frac{\gamma}{3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\frac{1}{7}$</td>
<td>$a_{21}$</td>
<td>$\frac{1}{3}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\frac{2-\sqrt{2}}{4}$</td>
<td>$a_{31}$</td>
<td>$-\frac{1356991263433}{2620883697614}$</td>
<td>$\frac{1}{4}$</td>
<td>0</td>
</tr>
<tr>
<td>201212234639789466</td>
<td>$a_{41}$</td>
<td>$-\frac{1778551891173}{14697912885533}$</td>
<td>$\frac{1}{4}$</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$a_{51}$</td>
<td>$-\frac{24076725932807}{39432440181437}$</td>
<td>$\frac{1}{4}$</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$b_1$</td>
<td>$\frac{657241292721}{9999460198745}$</td>
<td>$\frac{1}{4}$</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$b_i$</th>
<th>$b_i$</th>
<th>$\dot{b}_i$</th>
<th>$\dot{b}_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>$-\frac{7192245161075}{39999943759889}$</td>
<td>$-\frac{2633688882881}{5213126266014}$</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>$\frac{9733436383745}{81600254752689}$</td>
<td>$\frac{3295468053953}{1906441987965}$</td>
</tr>
</tbody>
</table>

select $\gamma$. With $\gamma = 1/4$, the best solution occurs at $c_3 = (3 - \sqrt{3})\gamma$, $c_4 = 1013918320559 / 1767732206903$ and $c_5 = 11104060336079 / 14410068582929$, where $A(5) = 0.002970$, $D = 1.192$, $R^{(3)}_{\text{int}}(-\infty) = -0.7321$, $\text{Eig}(M) = \{0.2869, -0.2659, 0.06250, -0.05517, -0.02035, 0.01571\}$ and $(b_i)_{\text{Min}} = -0.1125$. All internal stages are I-stable. An externally embedded method may be added to this method, ESDIRK4(3I)6L[2]SA.C(3), by solving $\tau^{(1,2,3)}_1 = \dot{\tau}^{(4)}_1 = +1/1000$. Rather than choosing $\gamma = 1/4$, one may select $\gamma = 0.38885767110289213237183$ to provide second-order damping to the L-stable method. With $c_3 = (3 - \sqrt{3})\gamma$, $c_4 = 20232276293216 / 41034790913227$ and $c_5 = 81138515489099 / 999741853997$, the method, ESDIRK4(3I)6L[2]SA_SOD, is characterized by $A(5) = 0.01670$, $D = 1.000$, $R^{(3)}_{\text{int}}(-\infty) = -0.7321$ and $\text{Eig}(M)_{\text{Min}} = -0.6705$. The price of second order damping is nearly a six-fold increase in leading-order error and a larger value of $\gamma$.

7.1.2 QESDIRK

If one is to pursue a fourth-order, six-stage QESDIRK, then one must first establish bounds on $\gamma$ for L-stability by computing the roots of $E_6$ and $E_6^2 - 4E_6 E_{10}$. These bounds are given in Table 19. One now enforces the following conditions: $\tau^{(1,2,3,4)}_1 = p_5 = b_2 = q^{(2)}_{2,3,4,5} = q^{(3)}_{3,4,5} = 0$ and $R^{(4,5)}_{\text{int}}(-\infty) = 0$ for an SOV = \{4, 2, 3, 3, 3, 4\}. Unlike previous attempts to make useful QESDIRKs, this particular class is more promising. Each of the two methods derived here have a value of $c_5$ slightly beyond the integration interval. For reference, the ESDIRK with $c_2 = 2\gamma = 250/1000$ has $q^{(3)}_2 = 0.02083$, $A(5) = 0.002970$ and $R^{(3)}_{\text{int}}(-\infty) = -0.7321$. Choosing $c_2 = \gamma = 295/1000$, one may derive a method with $q^{(3)}_2 = 0.004279$, $\text{Eig}(M) = \{2.138, 295/1000\}$.
2.064, 0.08703, -0.06570, -0.01498, 0.008700), \( A^{(5)} = 0.003189 \), \( D = 9.608 \),
c\( = (9 - \sqrt{33})\gamma/4 \), \( c_4 = \frac{28354460706980}{52673702457973} \), \( c_5 = \frac{14130336836023}{135806913221101} \) and \( R^{(3)}(-\infty) = -0.7792 \).

All internal stages are I-stable. To reduce \( q_2^{(3)} \) further, one may choose \( c_2 = 2\gamma/3 = 320/1000 \) and derive a method with \( q_2^{(3)} = 0.001618 \), \( \text{Eig}(\mathbf{M}) = \{ -8.115, 8.189, 0.1024, -0.06746, -0.01210, 0.005153 \} \), \( A^{(5)} = 0.004828 \), \( D = 35.10 \), \( c_3 = (2 - \sqrt{2})\gamma \), \( c_4 = \frac{129814247449}{241876472878387} \), \( c_5 = \frac{1156353931134}{11078645848867} \) and \( R^{(3)}(-\infty) = -0.8284 \). All internal stages are also I-stable. Note that moving from \( c_2 = 2\gamma \) to \( c_2 = 2\gamma/3 \) reduces \( q_2^{(3)} \) by a factor of 12.87. This method, QESDIRK4(3)6L[2]SA is presented in Table 20. The L-stable embedded method satisfies \( \hat{\tau}_1^{(1,2,3)} = \hat{p}_6 = \hat{p}_5 = 0, \hat{\tau}_1^{(4)} = 1/1000 \) and has \( B^{(5)} = 1.701 \) and \( E^{(5)} = 1.515 \). One may also consider second-order damping in QESDIRK methods. Values of \( \gamma \) to achieve this are provided in Table 21. Selecting \( c_2 = 2\gamma/3 = 2a_{22} \) and setting \( c_3 = (2 - \sqrt{2})\gamma \), \( c_4 = \frac{28292408695254}{15635736919643} \) and \( c_5 = \frac{85396608017}{51083462577462} \). The result is QESDIRK4(3)6L[2]SA_SOD with \( q_2^{(3)} = 0.005473 \), \( R^{(3)}(-\infty) = -0.8284 \), \( \text{Eig}(\mathbf{M})_{\text{Min}} = -56.61 \), \( A^{(5)} = 0.05458 \) and \( D = 16.54 \). The L-stable embedded method satisfies \( \hat{\tau}_1^{(4)} = +1/50 \) to give \( B^{(5)} = 1.080 \) and \( E^{(5)} = 1.515 \).

### Table 18. Scheme properties used to investigate the effects of internal stability.

| Scheme | \( R^{(3)}_\text{int}(-\infty) \) | \( R^{(4)}_\text{int}(-\infty) \) | \( R^{(5)}_\text{int}(-\infty) \) | \( A^{(5)} \) | \( B^{(5)} \) | \( D \) | \( ||\mathbf{M}|| \) |
|--------|--------------------------------|--------------------------------|--------------------------------|-------------|-------------|--------|---------------|
| ESDIRK4(3)6L[2]SA | 0 | 0 | 0 | 0.001830 | 1.279 | 1.585 | |
| ESDIRK4(3)6L[2]SA_A | 0.5 | -0.5 | -0.5 | 0.001457 | 1.396 | 1.000 | |
| ESDIRK4(3)6L[2]SA_B | 1.0 | -1.0 | -1.0 | 0.001962 | 1.411 | 1.000 | |
| ESDIRK4(3)6L[2]SA_C | 1.5 | -1.5 | -1.5 | 0.003490 | 1.340 | 1.059 | |
| ESDIRK4(3)6L[2]SA_D | 2.0 | -2.0 | -2.0 | 0.004193 | 1.462 | 1.112 | |
| ESDIRK4(3)6L[2]SA_E | 2.5 | -2.5 | -2.5 | 0.004610 | 1.571 | 1.161 | |

### Table 19. Bounds on \( \gamma \) for L-stability of six-stage, fourth-order QESDIRKs.

<table>
<thead>
<tr>
<th>( \gamma )</th>
<th>( c_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.24799463621274745516799 ( \leq \gamma \leq 0.67604239322628132887239 )</td>
<td>2( \gamma )</td>
</tr>
<tr>
<td>0.29098403905817045504814 ( \leq \gamma \leq 0.80176329817056659341130 )</td>
<td>( \gamma )</td>
</tr>
<tr>
<td>0.31645634286204433931952 ( \leq \gamma \leq 0.89587590082599420564813 )</td>
<td>2( \gamma/3 )</td>
</tr>
<tr>
<td>0.35017411257697641284118 ( \leq \gamma \leq 1.04766882255681262972286 )</td>
<td>( \gamma/3 )</td>
</tr>
</tbody>
</table>

Note that \( \gamma \) to achieve this are provided in Table 21. Selecting \( c_2 = 2\gamma/3 = 2a_{22} \) and setting \( c_3 = (2 - \sqrt{2})\gamma \), \( c_4 = \frac{28292408695254}{15635736919643} \) and \( c_5 = \frac{85396608017}{51083462577462} \). The result is QESDIRK4(3)6L[2]SA_SOD with \( q_2^{(3)} = 0.005473 \), \( R^{(3)}(-\infty) = -0.8284 \), \( \text{Eig}(\mathbf{M})_{\text{Min}} = -56.61 \), \( A^{(5)} = 0.05458 \) and \( D = 16.54 \). The L-stable embedded method satisfies \( \hat{\tau}_1^{(4)} = +1/50 \) to give \( B^{(5)} = 1.080 \) and \( E^{(5)} = 1.515 \).
Table 20. QESDIRK4(3)6L[2]SA.

<table>
<thead>
<tr>
<th>( \gamma )</th>
<th>( c_2 )</th>
<th>( c_3 )</th>
<th>( c_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \frac{16}{75} )</td>
<td>( \frac{8}{75} )</td>
<td>( \frac{8}{75} )</td>
<td>( \frac{8}{75} )</td>
</tr>
<tr>
<td>( \frac{8(2-\sqrt{2})}{25} )</td>
<td>( \frac{8}{25} )</td>
<td>( \frac{8}{25} )</td>
<td>( \frac{8}{25} )</td>
</tr>
<tr>
<td>( \frac{1}{(1-b_2-b_3-b_4-\gamma)} )</td>
<td>( b_2 )</td>
<td>( b_3 )</td>
<td>( b_4 )</td>
</tr>
<tr>
<td>( b_i )</td>
<td>( 1 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
</tr>
</tbody>
</table>

Table 21. Values for \( \gamma \) second-order damped, six-stage, fourth-order, L-stable QESDIRKs.

<table>
<thead>
<tr>
<th>( \gamma )</th>
<th>( c_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.38885767110289211323372</td>
<td>2( \gamma )</td>
</tr>
<tr>
<td>0.44773852423399424486223</td>
<td>( \gamma )</td>
</tr>
<tr>
<td>0.48034935707874730014389</td>
<td>2( \gamma/3 )</td>
</tr>
<tr>
<td>0.5220263373470430577704</td>
<td>( \gamma/3 )</td>
</tr>
</tbody>
</table>

7.1.3 SDIRK

Five-stage, stiffly-accurate, L-stable SDIRK methods permit \((s^2 - s + 2)/2 = 11\) DOF as seen below.

Solving all eight order conditions for fourth-order classical accuracy gives a three-parameter family of methods in \( \gamma, c_3 \) and \( c_4 \) (excluding special cases). The second abscissa, \( c_2 \), is found to be

\[
c_2 = \frac{3(\varphi_1 + c_4 \varphi_1)}{\varphi_3 + c_4 \varphi_4}
\]
with \( \varphi_1 = (-1 + 12\gamma - 50\gamma^2 + 76\gamma^3 - 40\gamma^4) \), \( \varphi_2 = (1 - 10\gamma + 36\gamma^2 - 48\gamma^3 + 24\gamma^4) \), \( \varphi_3 = (-1 + 14\gamma - 72\gamma^2 + 150\gamma^3 - 120\gamma^4) \) and \( \varphi_4 = (1 - 12\gamma + 54\gamma^2 - 96\gamma^3 + 72\gamma^4) \). The remaining coefficients are found to be

\[
b_2 = \frac{\varphi_7 - 2c_3\varphi_1 + 2c_3(-\varphi_1 + 3c_4\varphi_2)}{12(c_2-c_3)(c_2-c_4)(c_2-\gamma)} \quad (255)\\
b_3 = \frac{-\varphi_7 + 2c_4\varphi_1 - 2c_2(-\varphi_1 + 3c_4\varphi_2)}{12(c_2-c_3)(c_3-c_4)(c_3-\gamma)} \quad (256)\\
b_4 = \frac{-\varphi_7 + 2c_3\varphi_1 - 2c_2(-\varphi_1 + 3c_3\varphi_2)}{12(c_2-c_4)(c_3-c_4)(c_4-\gamma)} \quad (257)\\
a_{32} = \frac{-(c_2-c_3)(c_3-\gamma)(\varphi_8 + 4c_4\varphi_3)}{2(c_2-\gamma)(\varphi_7 - 2c_4\varphi_1 + 2c_2(-\varphi_1 + 3c_4\varphi_2)} \quad (258)\\
a_{42} = \frac{-(c_2-c_4)(c_4-\gamma)(2c_4\varphi_4 - 4c_2\varphi_3 + 3c_3\varphi_6 + c_3\varphi_5 + 4c_4\varphi_3)}{2(c_2-c_3)(c_2-\gamma)(\varphi_7 - 2c_3\varphi_1 + 2c_2(-\varphi_1 + 3c_3\varphi_2))} \quad (259)\\
a_{43} = \frac{(c_2-c_4)(c_4-\gamma)(c_4-\gamma)(\varphi_7 - 2c_3\varphi_1 + 2c_2(-\varphi_1 + 3c_3\varphi_2))}{(c_2-c_3)(c_3-\gamma)(\varphi_7 - 2c_3\varphi_1 + 2c_2(-\varphi_1 + 3c_3\varphi_2))} \quad (260)
\]

where \( \varphi_1 = (2 - 9\gamma + 6\gamma^2) \), \( \varphi_2 = (1 - 4\gamma + 2\gamma^2) \), \( \varphi_3 = (-1 + 9\gamma - 18\gamma^2 + 6\gamma^3) \), \( \varphi_4 = (1-10\gamma+24\gamma^2-12\gamma^3) \), \( \varphi_5 = (3-28\gamma+60\gamma^2-24\gamma^3) \), \( \varphi_6 = (-5+48\gamma-108\gamma^2+48\gamma^3) \), \( \varphi_7 = (3-16\gamma+12\gamma^2) \), \( \varphi_8 = (3-28\gamma+60\gamma^2-24\gamma^3) \) and with L-stability being possible for 0.247994636212747551679910 \( \leq \gamma \leq 0.67604239322628.13288723863 \). Cash [97] appears to be the first to derive \( p(p-1) \) methods from this class; however, he set \( c_2 = -0.7 \). From this exact solution, two five-stage, stiffly-accurate SDIRK4 methods given by Hairer and Wanner [193] have \( \gamma = 1/4 \), \( c_2 = \frac{3}{4} \), \( c_3 = \frac{11}{20} \), \( c_4 = \frac{1}{2} \) and \( A^{(5)} = 0.002504 \) for SDIRK4(1); and \( \gamma = 4/15 \), \( c_2 = \frac{23}{30} \), \( c_3 = \frac{17}{30} \), \( c_4 = \frac{707}{1931} \), and \( A^{(5)} = 0.004229 \) for SDIRK4(2). The minimum value for the eigenvalues of the algebraic stability matrix and \( (b_i)_{\text{Min}} \) are \(-112.09\) and \(-7.083\) for SDIRK4(1) and \(-19.14\) and \(-2.999\) for SDIRK4(2). Both methods are provided with embedded methods that are not A-stable. For SDIRK(1), an A-stable embedded method with \( \hat{R}(\infty) = -0.5 \) is \( \hat{b} = \{973/960, -2203/120, 1015/128, -85/12, 23/80\} \). For SDIRK(2), one solves the four order conditions at third-order, along with \( \hat{b}_5 = 68097476/225705375 \). The minimum \( A^{(5)} \) for this family of methods, with \( \gamma = 1/4 \), occurs at approximately \( c_2 = \frac{27}{25} \), \( c_3 = \frac{72}{107} \), \( c_4 = \frac{19}{37} \) and \( A^{(5)} = 0.002006 \). Nearby, but having simpler coefficients, is SDIRK4M, with \( c_2 = \frac{9}{107} \), \( c_3 = \frac{19}{37} \), \( c_4 = \frac{2}{3} \) and \( A^{(5)} = 0.002013 \). The other six undetermined coefficients are \( a_{32} = -175/5148 \), \( a_{42} = +201/2990 \), \( a_{43} = +891/11500 \), \( b_2 = -400/819 \), \( b_3 = +99/35 \), \( b_4 = -575/252 \) and \( \lambda^M_{\text{Min}} = -13.03 \). One may append a third-order embedded method with \( \hat{R}(\infty) = -0.5 \) by setting \( \hat{b} = \{41911/60060, -83975/144144, 3393/1120, -27025/11088, 103/352\} \). The minimum \( A^{(5)} \) at any \( \gamma \) that ensures L-stability appears to occur at the minimum value of \( \gamma = 0.247994636212747551679910 \). At \( \gamma = 248/1000 \), \( c_2 = \frac{4238010953}{46647086000} \), \( c_3 = \frac{58}{85} \), \( c_4 = \frac{67}{109} \), \( A^{(5)} = 0.001923 \) and \( \lambda^M_{\text{Min}} = -10.78 \). A search for algebraically stable, stiffly-accurate, L-stable SDIRK methods is hampered by the ability to find values of \( c_3 \), \( c_4 \) and \( \gamma \), where all \( b_i > 0 \). A numerical search was only able to find one region, in the neighborhood of \( c_3 = 0.335 \), \( c_4 = 0.800 \), \( \gamma = 0.248 \), where \( D = 832.9 \) and \( A^{(5)} = 0.8787 \). The algebraic stability matrix is not positive semidefinite at
this point, and \( r_{F2} = 0.002081 \). It is likely that there is no algebraically stable solution to this method. Skvortsov [398] derives a confluent analog to SDIRK4(1) and SDIRK(2) called S54b with \( A^{(5)} = 0.003807 \) and particularly simple coefficients.

One may also solve for a stiffly-accurate, five-stage, fourth-order SDIRK using an incomplete specification of simplifying assumption \( C(2) \), \( q_{2,3,4}^{(2)} = 0 \), to generate methods having an SOV = \{1, 2, 2, 2, 4\}. This requires solving \( b_1 = \sum_{i=1}^{4} b_ia_i = \tau_1^{(1,2,3,4)} = \tau_3^{(4)} = q_{2,3,4}^{(2)} = 0 \). Two nonconfluent solutions exist to this method. Setting \( \gamma = 1/4 \), the two solutions are given in Table 22. The more promising nonconfluent solution for \( \gamma = 1/4 \), SDIRK4(3)5L[1]SA.C(2), appears to be at \( c_2 = (2 - \sqrt{2})/4 \) where \( A^{(5)} = 0.002454 \), \( \lambda_{\text{Min}}^M = -0.2045 \) and \( D = 1.844 \). This particular method satisfies \( \hat{\gamma}_{1,2,3}^{(1)} = \hat{\gamma}_{2}^{(3)} = 0 \), \( \hat{R}(\infty) = -0.5 \). The second solution, with \( c_2 = (2 + \sqrt{2})/4 \) has \( A^{(5)} = 0.004224 \), \( \lambda_{\text{Min}}^M = -0.4711 \) and \( D = 1.000 \).

\[
\begin{array}{cccccc}
\frac{1}{4} (2\sqrt{2}) & \frac{1}{4} (2\sqrt{2}) & 0 & 0 & 0 & 0 \\
\frac{1}{3} (\sqrt{2}) & \frac{1}{3} (\sqrt{2}) & \frac{2}{3} (\sqrt{2}) & 0 & 0 & 0 \\
\frac{1}{4} (\sqrt{2}) & \frac{1}{4} (\sqrt{2}) & \frac{1}{2} (\sqrt{2}) & \frac{1}{2} (\sqrt{2}) & 0 & 0 \\
0 & 0 & \frac{1}{4} (\sqrt{2}) & \frac{1}{4} (\sqrt{2}) & \frac{1}{4} (\sqrt{2}) & \frac{1}{4} (\sqrt{2}) \\
\end{array}
\]

Table 22. The two SDIRK4(4)5L[1]SA methods.

### 7.1.4 QSDIRK

A QSDIRK may be derived by solving the same equations as those solved for SDIRK4(3)5L[1]SA.C(2) along with the \( \gamma \) given by Table 19. In analogy with QES-DIRK4(3)6L[2]SA, one may set \( a_{11} = \gamma/3 \) and find a method, QSDIRK4(3)5L[1]SA, with \( \gamma = 8/25 \), \( c_1 = 8/75 \), \( c_2 = 529165024604/5921916928537 \), \( c_3 = 2779087652291/5736144612805 \) and \( c_4 = 9665836798102/8373691518525 \). The L-stable method has \( A^{(5)} = 0.006849 \), \( D = 19.55 \) and \( \lambda_{\text{Min}}^M = -3.126 \). Note that the second-order error on stage one is \( a_{11}^2/2 \). This means that the ratio of the second-order error on stage one for a QSDIRK \( (a_{11} = 8/75) \) relative to an SDIRK \( (a_{11} = 1/4) \) is 1024/5625 or about 18 percent. If \( a_{11} = 2/3 \) is chosen then \( c_1 = 59/400 \), \( c_2 = 2283312914982/20263697689405 \), \( c_3 = 2336504909662/4419237820643 \) and \( c_4 = 9335370403544/8318781608643 \). The L-stable method has \( A^{(5)} = 0.004856 \), \( D = 7.897 \) and \( \lambda_{\text{Min}}^M = -1.193 \). Both methods use embedded methods that satisfy \( \hat{\gamma}_{1,2,3}^{(1,2,3)} = \hat{\gamma}_{2}^{(3)} = 0 \), \( \hat{R}(\infty) = -0.5 \), but neither is A-stable.

### 7.2 Fifth-Order Methods

#### 7.2.1 ESDIRK

To derive six-stage, fifth-order, stiffly-accurate ESDIRK methods, simplifying assumption \( C(2) \) is used. The resulting 14 order conditions take the form \( \tau_{1,2,3,4,5}^{(1,2,3,4,5)} = \)
\[ p_5 = \tau_3^{(4)} = \tau_{4,5,8}^{(5)} = q_2^{(2)}_{3,4,5} = 0, \] leaving a two-parameter family of methods in terms of \( c_4 \) and \( c_5 \) with
\[
c_3 = \frac{4\gamma(-1 + 16\gamma - 92\gamma^2 + 212\gamma^3 - 180\gamma^4 + c_5\omega_1)}{-1 + 18\gamma - 120\gamma^2 + 356\gamma^3 - 360\gamma^4 + c_5\omega_2}
\]

(261)

and \( \omega_1 = 1 - 14\gamma + 72\gamma^2 - 150\gamma^3 + 120\gamma^4, \) \( \omega_2 = 1 - 16\gamma + 96\gamma^2 - 240\gamma^3 + 240\gamma^4, \) \( E_{10} = (1 - 10\gamma + 20\gamma^2)(1 - 15\gamma + 30\gamma^2)(-1 + 25\gamma - 200\gamma^2 + 600\gamma^3 - 600\gamma^4 + 240\gamma^5)/14400, \) \( E_8 = (3 - 120\gamma + 1880\gamma^2 - 14880\gamma^3 + 63600\gamma^4 - 144960\gamma^5 + 158400\gamma^6 - 57600\gamma^7)/2880, \) \( E_6 = (-1 + 30\gamma - 300\gamma^2 + 1200\gamma^3 - 1800\gamma^4 + 720\gamma^5)/360 \) and \( R_{\text{int}}^{(3)}(-\infty) = (c_3^2 - 4c_3\gamma + 2\gamma^2)/(2\gamma^2) \). Expressions for the Runge-Kutta coefficients, \( R_{\text{int}}^{(4)}(-\infty), R_{\text{int}}^{(5)}(-\infty) \) and \( A^{(6)} \), are too cumbersome to include. For \( L \)-stability of the overall scheme, \( \gamma \) must be a root of the polynomial \( 1 - 25\gamma + 200\gamma^2 - 600\gamma^3 + 600\gamma^4 - 120\gamma^5 = 0 \). Only one of the five roots that ensures \( R(-\infty) = 0 \) also provides I-stability: \( \gamma \approx 0.2780538411364523249315862 \). Upon inspection of \( |R_{\text{int}}^{(3)}(-\infty)|, |R_{\text{int}}^{(4)}(-\infty)|, \) and \( |R_{\text{int}}^{(5)}(-\infty)| \), it may be determined that if both \( c_4 \) and \( c_5 \) remain positive, that internal stability for very stiff scaled eigenvalues is only possible for specific regions within \( 0 \leq c_4 \leq 0.20782427518534172606939559 \) and \( 0 \leq c_5 \leq 0.23111961220951036305109788 \). Enforcing \( R_{\text{int}}^{(4)}(-\infty) = 0 \) and \( R_{\text{int}}^{(5)}(-\infty) = 0 \) forces \( c_4 = c_5 = 0.07838319999965920091092891 \) and \( R_{\text{int}}^{(3)}(-\infty) = 0.4876 \); however, because \( b_4 \) and \( b_5 \) are each proportional to \( (c_4 - c_5)^{-1} \), they become infinite. Separating the two abscissae a bit, at \( c_4 = 1/10 \) and \( c_5 = 3/50, R_{\text{int}}^{(3)}(-\infty) = 0.4438, R_{\text{int}}^{(4)}(-\infty) = -0.2198, R_{\text{int}}^{(5)}(-\infty) = 0.4549 \) and \( A^{(6)} = 0.004615 \). Stages 2, 3 and 6 are I-stable but stages 4 and 5 are not. It appears to be impossible to make all stages I-stable with positive abscissae. A third-order, \( L \)-stable, externally embedded method may be done as \( \tau_1^{(1,2,3)} = \hat{\gamma}_0 = \hat{p}_5 = 0, \) and \( \tau_1^{(4)} = -1/500. \) The resulting 5(3) method, ESDIRK5(3)6L2/SA, appears in Table 23. Other schemes are possible. One may attempt third-order internal error-control by setting \( c_5 = 1 \) and implying \( c_3 = \frac{4\gamma(-1+2\gamma)(-1+3\gamma)(-1+5\gamma)}{-1+12\gamma-48\gamma^2+60\gamma^3} \approx -0.3778. \) As this value is quite undesirable, internal error-control for this method cannot be recommended. Next, one can minimize sixth-order error by solving \( \frac{\partial A^{(6)}}{\partial c_4} = 2A^{(6)} = 0 \) to get \( c_4 \approx \frac{595380727}{839046035} \approx 0.7143, \) \( c_5 \approx \frac{1870580983}{1564388764} \approx 1.187, R_{\text{int}}^{(3)}(-\infty) = -0.08432, R_{\text{int}}^{(4)}(-\infty) = -2.930, R_{\text{int}}^{(5)}(-\infty) = -3.615 \) and \( A^{(6)} = 0.002005. \) Again, this scheme cannot be recommended because of internal instability. Including a fourth-order, \( A \)-stable externally embedded method to the internally stable main method given previously is not possible without using the main method to satisfy \( \tau_3^{(4)} = 0. \) Reduced storage and conditionally contractive methods are not pursued because of the small ranges of \( c_4 \) and \( c_5 \) that give rise to internally stable methods. Skvortsov [396, 398] construct fifth-order methods but selects \( \gamma \approx 0.1411. \) In an earlier paper, Skvortsov [396] designs a method that has \( A^{(6)} = 0.007660 \) and \( c_4 = 4/3. \) A later paper [398] constructs an \( L(\alpha) \)-stable method, ES65, with \( \alpha = 72.3 \) degrees. Shintani [384] achieves fifth-order but without using the stiffly-accurate assumption.

For completeness, one may also derive six-stage, fifth-order, stiffly-accurate ES-
DIRK methods by using simplifying assumption $C(3)$ in truncated form. The resulting 16 order conditions take the form \( r_1(1,2,3,4,5) = p_5 = b_2 = \sum_{i=1}^{s} b_i a_{i2} = r_5(5) = (23,4,5) = q_{3,4,5} = 0 \), leaving a zero-parameter family of methods with two solutions for which

\[
c_4 = \frac{-12 + 15c_4 + 15c_5 - 20c_3c_5 + 60\gamma - 60c_3\gamma - 60c_5\gamma + 60c_3c_5\gamma}{5(-3 + 4c_3 + 4c_5 - 6c_3c_5 + 12\gamma - 12c_3\gamma - 12c_5\gamma + 12c_3c_5\gamma)}.
\]

Choosing \( c_3 = (3 + \sqrt{3})\gamma, \) then

\[
c_5 = \frac{(-2 + \sqrt{3} + 32\gamma + 14\sqrt{3}\gamma \pm 184\gamma^2 - 64\sqrt{3}\gamma^2 \pm 424\gamma^3 + 88\sqrt{3}\gamma^3 \pm 360\gamma^4)}{(-2 - \sqrt{3} + 28\gamma + 12\sqrt{3}\gamma \pm 144\gamma^2 - 48\sqrt{3}\gamma^2 \pm 300\gamma^3 + 60\sqrt{3}\gamma^3 \pm 240\gamma^4)}.
\]

Using \( c_3 = (3 - \sqrt{3})\gamma, \) one has \( R^{(3)}_{\text{int}}(-\infty) = -0.7321, R^{(4)}_{\text{int}}(-\infty) = -0.5671, R^{(5)}_{\text{int}}(-\infty) = -10.98, D = 17.17, A^{(6)} = 0.00317, \text{Eig}(M) = \{ -1.526, 0.02888, 0.09559, 0.07731, -0.06133, -0.02711 \}, \) \( (b_i)_{\text{Min}} = -0.5367 \) and \( (c_i)_{\text{Max}} = 1.352. \) If instead, one uses \( c_3 = (3 + \sqrt{3})\gamma, \) then one has \( R^{(3)}_{\text{int}}(-\infty) = 2.732, R^{(4)}_{\text{int}}(-\infty) = 2.021, R^{(5)}_{\text{int}}(-\infty) = -0.04623, D = 1.316, A^{(6)} = 0.004115, \text{Eig}(M) = \{ -0.1167, 0.07731, 0.5699, -0.02888, -0.1894, 0.01759 \}, \) \( (b_i)_{\text{Min}} = -0.2736 \) and \( (c_i)_{\text{Max}} = 1.316. \)

### 7.2.2 SDIRK

Five-stage, fifth-order, stiffly-accurate, L-stable methods appear to be impossible based on the exact solution to the same fourth-order method. To satisfy L-stability and \( \tau_9^{(5)} = 0, \) one has \( \gamma \approx 0.2781 \) (See Table 5). This leaves both \( c_3 \) and \( c_4 \) to satisfy the other eight fifth-order conditions. As \( \tau_6^{(5)} \) and \( \tau_7^{(5)} \) are both linear in these variables, one may solve for \( c_3 \) and \( c_4 \) to verify that the resulting method is not fifth-order. In spite of this, Ababneh et al. [1] present a stiffly-accurate SDIRK in five implicit stages which is claimed to be L-stable.

If one is willing to forsake stiff accuracy but would like to retain L-stability, a solution can be obtained. Cooper and Sayfy [116] \( (\gamma = (6 - \sqrt{6})/10) \) and Shintani
and Yoshida [383, 385] (γ = 0.33) show that five-stage, fifth-order, A-stable methods exist. In fact, there is a pair of two-parameter solution families to the five-stage, fifth-order SDIRK method. The remaining DOF are γ and one abscissa, e.g., c4. The pair of families derives from the fact that one solution is quadratic in a Butcher coefficient; hence, there are two solutions. It appears that the general solution satisfies assumption D(1). Order conditions are then \( \tau_1^{(1,2,3,4,5)} = \tau_2^{(1)} = \tau_3^{(1,2,3,4,5)} = \tau_4^{(2)} = \tau_5^{(2,3,4,5,6)} \), and one finds \( c_2 = (\gamma(-3 + 49\gamma - 288\gamma^2 + 704\gamma^3 - 720\gamma^4 + 240\gamma^5 + c_4(3 - 40\gamma + 192\gamma^2 - 360\gamma^3 + 240\gamma^4)))/(1 + 19\gamma - 136\gamma^2 + 432\gamma^3 - 600\gamma^4 + 240\gamma^5 + c_4(1-16\gamma + 96\gamma^2 - 240\gamma^3 + 240\gamma^4)) \) and \( c_5 = 1 - \gamma \). One may select γ from Table 5 to determine the stability. For L-stability, \( \gamma = 0.278053811364523249315862 \), a nearly minimal \( A^{(6)} \) solution is found and given in Table 24 as SDIRK5(5)5L[1] at \( c_4 = 3/20 \) where \( A^{(6)} = 0.002549, D = 1.023 \) and the minimum value for the eigenvalues of the algebraic stability matrix is \(-0.3568\).

### 7.3 Sixth-Order Methods

#### 7.3.1 ESDIRK

Using simplifying assumptions C(2) and D(1) and dispensing with stiff accuracy, one may design a two-parameter family of A-stable, sixth-order methods. These two DOF are probably best used to rein in internal instability. The methods are obtained by solving \( \tau_1^{(1,2,3,4,5,6)} = \tau_2^{(2)} = \tau_3^{(1,2,3,4,5,6)} = \tau_4^{(5)} = \tau_5^{(6)} = 0 \). This leaves, for example, \( c_4, c_5 \), and \( \gamma \); however, for A-stability, \( p_6 \) must vanish. The vanishing of \( p_6 \) implies that \( \gamma \) must be a root of \( 1 - 30\gamma + 300\gamma^2 - 1200\gamma^3 + 1800\gamma^4 - 720\gamma^5 = 0 \), or from Table 4, \( \gamma = 0.4732638912582953244555885 \) with \( R(-\infty) = 0.8373 \). The final two DOF are used to control \( R_{\text{int}}^{(i)}(-\infty) \): \( c_4 = 2/5 \) and \( c_5 = 81/100 \). The resulting method has \( c_3 = 210618880064/9718174947659 \), \( c_6 = 1 - \gamma \), \( R_{\text{int}}(-\infty) = \{1, -1, 0.9095, -0.9710, 0.9873, 0.3560\} \), \( D = 22.83, A^{(7)} = 0.004899 \) and \( \text{Eig}(M) = \{5.480, -5.056, 0.2286, -0.06744, 0.0002664, 0\} \).

<table>
<thead>
<tr>
<th>( \bar{b}_1 )</th>
<th>0.2522762558582</th>
<th>0.182679036555</th>
<th>0.18292826351</th>
<th>0.501116467727</th>
<th>0.509636994297</th>
<th>0.3891594170934</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \bar{b}_2 )</td>
<td>1.2162329469185</td>
<td>1.2907214117901</td>
<td>1.3762606409857</td>
<td>1.2224457745173</td>
<td>1.3891594170934</td>
<td></td>
</tr>
</tbody>
</table>
7.4 Seventh-Order Methods

7.4.1 EDIRK

Al-Rabeh [18] proves that there does not exist an EDIRK method of order seven in six-stages that is both nonconfluent and has nonzero values of \( b_i \).

8 Six- and Seven-stage Methods (\( S_1 = 6 \))

8.1 Fifth-Order Methods

8.1.1 ESDIRK

Seven-stage, stiffly-accurate ESDIRK methods

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2\gamma & \gamma & \gamma & 0 & 0 & 0 & 0 & 0 \\
c_3 & (c_3 - a_{32} - \gamma) & a_{32} & \gamma & 0 & 0 & 0 & 0 \\
c_4 & (c_4 - a_{42} - a_{43} - \gamma) & a_{42} & a_{43} & \gamma & 0 & 0 & 0 \\
c_5 & (c_5 - a_{52} - a_{63} - a_{54} - \gamma) & a_{52} & a_{53} & a_{54} & \gamma & 0 & 0 \\
c_6 & (c_6 - a_{62} - a_{63} - a_{64} - a_{65} - \gamma) & a_{62} & a_{63} & a_{64} & a_{65} & \gamma & 0 \\
1 & (1 - b_2 - b_3 - b_4 - b_5 - b_6 - \gamma) & b_2 & b_3 & b_4 & b_5 & b_6 & \gamma \\
b_1 & (1 - b_2 - b_3 - b_4 - b_5 - b_6 - \gamma) & b_2 & b_3 & b_4 & b_5 & b_6 & \gamma \\
b_1 & (1 - b_2 - b_3 - b_4 - b_5 - b_6 - b_7) & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 \\
\end{array}
\]

provide 22 DOF, twenty after \( C(2, 2) \) and \( \tau_i^{(1)} \) are imposed. Row simplifying assumption \( C(2) \) ensures a stage-order two method. L-stable methods may be found for a range \( \gamma \) values determined from the polynomials \( 4E_{10}^3E_6 + 27E_{12}^2E_6^2 - 18E_{12}E_1E_8E_6 - E_{10}^2E_8^2 + 4E_{12}E_6^3 = 0 \) and \( E_6 = 0 \) where \( E_{12} = \gamma^{12} \), \( E_{10} = (-1 + 60\gamma - 1500\gamma^2 + 20400\gamma^3 - 165600\gamma^4 + 829440\gamma^5 - 2563200\gamma^6 + 4752000\gamma^7 - 4968000\gamma^8 + 2592000\gamma^9 - 432000\gamma^{10})/14400 \), \( E_8 = (1 - 48\gamma + 920\gamma^2 - 9120\gamma^3 + 50400\gamma^4 - 155520\gamma^5 + 250560\gamma^6 - 172800\gamma^7 + 28800\gamma^8)/960 \) and \( E_6 = (-1 + 36\gamma - 450\gamma^2 + 2400\gamma^3 - 5400\gamma^4 + 4320\gamma^5 - 720\gamma^6)/360 \). The first polynomial has 32 unique roots (26 complex and six real). The minimum value of \( \gamma \) for a seven-stage, fifth-order, L-stable, stiffly-accurate method is the fifth largest real root of the first polynomial. The largest value that \( \gamma \) may take for these methods is the fourth largest root \( E_6 \). To 25 digits, one may determine, as have Hairer and Wanner [193], \( 0.1839146536751751632321436 \leq \gamma \leq 0.3341423670680504359540301 \).

Stiffly accurate, seven-stage, fifth-order methods may be obtained by using assumptions \( C(2) \) or \( C(3) \) in truncated form. Considering the former, one must solve the 15 equations, \( \tau_1^{(1,2,3,4,5)} = q_{2,3,4,5,6} = p_6 = \tau_3^{(4)} = \tau_{4,5,8}^{(5)} = 0 \), with seven remaining DOF, e.g., \( c_3, c_4, c_5, c_6, \gamma \), and two others. Internal stability for stages three through six, \( R_{\text{int}}^{(3,4,5,6)}(-\infty) = 0 \), may be accomplished leaving \( c_4, c_6, \) and \( \gamma \). As \( \gamma \) is probably best chosen as small as possible, \( \gamma = 0.184 \), and \( c_3 = (2 \pm \sqrt{2})\gamma \) is the result of solving a quadratic equation. Hence, there appear to be two distinct
two-parameter families of solutions when $\gamma = 0.184$ is chosen. Stage four is internally I-stable between the two values of $c_4$; $c_4 = 3\gamma \pm \sqrt{3 + 2\sqrt{3}}$ for $\gamma \geq 0$ or $0.0841874 \leq c_4 \leq 1.01981$.

One solution to this set of equations is the basis for ESDIRK5(4)7L[2]SA where $c_4 = 13/25$ and $c_6 = 26/25$, which gives $A^{(6)} = 0.001846, D = 8.971$, $\text{Eig}(M) = \{0.0800, -0.0001, -0.08133, 0.04008, 0.03386, -0.01772, 0.063435\}$ and $(b_i)_{\text{Min}} = -0.07600$. Stages 3, 4, 5 and 7 are L-stable, stage 2 is strictly A-stable but stage 6 is not A-stable. Fourth-order external error-control may be made nearly L-stable by solving $\hat{z}^{(1,2,3,4)} = \hat{z}^{(4)} = \hat{p}_7 = 0$ with $\hat{p}_6 / \gamma^6 = \hat{R}(-\infty) = 7/20$ where $\hat{R}(-\infty)$ must not be zero to keep sufficient separation between the $b_i$ and $\hat{b}_i$ coefficients. The scheme is given in Table 25. The embedded method has $B^{(6)} = 0.6915, E^{(6)} = 0.8503$ and $\hat{A}^{(5)} = 0.002171$. Additional methods are also possible. Setting $\gamma = 0.184$ and $c_3 = (2 - \sqrt{2})\gamma$, $c_4 = 848/1000$ and $c_6 = 648/1000$ results in a high quality method. For $\gamma = 0.184$ and $c_3 = (2 + \sqrt{2})\gamma$, $c_4 = 13275/100000$ and $c_6 = 889/1000$ also results in a high quality method. Using a similar approach, Skvortsov [396, 398] generates a stiffly-accurate, L-stable 5(4) pair with $\gamma = 1/5, A^{(6)} = 0.003279$ and $R_{\text{int}}(-\infty) = \{1, -1, -1/2, -71/14, 31919/7875, -93187/15750, 0\}$.

An alternative approach to designing a 5(4) ESDIRK pair in seven stages is to use truncated assumption $C(3)$. Methods with externally embedded schemes only, designed by using this assumption, must satisfy the 18 equations for the main method: $\tau_1^{(1,2,3,4)} = q_2^{(2)} = q_3^{(3)} = b_2 = p_6 = \tau_5^{(5)} = \sum_{i=1}^{s} b_i a_{i2} = 0$. Four DOF, e.g., $c_4$, $c_5$, $c_6$, $\gamma$ to enforce internal stability. One has $R_{\text{int}}^{(i)}(-\infty) = 0, i = 4, 5, 6$ and by setting $\gamma = 0.184$, all four of the DOF are consumed. A fourth-order, strongly A-stable embedded method may be made by solving $\tau_1^{(1,2,3,4)} = b_2 = \hat{p}_7 = 0$ with $\hat{p}_6 / \gamma^6 = \hat{R}(-\infty)$. One may immediately solve $C(3,3)$ to find $c_3 = (3 \pm \sqrt{3})\gamma$ and solve the cubic equation $R_{\text{int}}^{(4)}(-\infty) = 0$ for $c_4$ to find $c_4 = \gamma[3 \pm 3\cos(\phi) + \sqrt{3}\sin(\phi)]$, or $c_4 = \gamma[3 - 2\sqrt{3}\sin(\phi)]$, with $\phi = (1/3)\cot^{-1}(\sqrt{2})$. Solving $R_{\text{int}}^{(5)}(-\infty) = 0$ involves a quartic equation in $c_5$; hence, 24 combinations of $c_3, c_4$ and

\begin{table}[h]
\centering
\caption{ESDIRK5(4)7L[2]SA.}
\begin{tabular}{c|cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
46 & 23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1518047705759 & 1185841472805 & 23 & 0 & 0 & 0 & 0 & 0 & 0 \\
25 & 216541625120 & 3350617878647 & 23 & 0 & 0 & 0 & 0 & 0 \\
5906118540659 & 11338990001125 & 23 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1319096626979 & 4356877330928 & 922991294344 & 4729382608034 & -30819909217 & 23 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\

\end{tabular}
\end{table}
\( c_5 \) satisfy \( C(3, 3) = R^{(4)}_{\text{int}}(\infty) = R^{(5)}_{\text{int}}(\infty) = 0 \). One may then adjust \( c_6 \) to obtain a desirable \( R^{(6)}_{\text{int}}(\infty) \). If one insists on \( R^{(5)}_{\text{int}}(\infty) = C(3, 3) = 0 \), this appears to imply a confluent method having infinite \( D \) and \( \text{Eig}(M) \). One of the better schemes from the broader class of methods is found at \( c_3 = 2325355958545/9967121867317 \), \( c_4 = 5768599414322/13664887829871 \), \( c_5 = 11/20 \) and \( c_6 = 17/25 \) where \( A^{(6)} = 0.001594 \), \( D = 4.158 \), \( R^{(4)}_{\text{int}}(\infty) = 0 \), \( R^{(5)}_{\text{int}}(\infty) = 0.007233 \), \( R^{(6)}_{\text{int}}(\infty) = -0.9875 \) and \( \text{Eig}(M)_{\text{Min}} = -33.34 \). If one relaxes the internal stability requirement, more balanced schemes may be possible, but it appears that these ESDIRK methods are best constructed by using only \( C(2) \).

A 5(4) ESDIRK pair using L-stable, internal error-control may also be approached by using assumption \( C(2) \) or a truncated assumption \( C(3) \). To keep both methods L-stable, \( \gamma \) is constrained by the seven-stage, fifth-order main method, \( 0.18391465367517516321436 \leq \gamma \leq 0.3341423670680504359540301 \), and the six-stage, fourth-order method, \( 0.2479946362127474551679910 \leq \gamma \leq 0.676042393226-2813288723863 \). Therefore, to satisfy each constraint simultaneously, \( 0.24799463621 \) \( -27474551679910 \) \( \leq \gamma \leq 0.3341423670680504359540301 \). Using \( C(2) \), one must satisfy the 20 conditions, \( \tau_1^{(1,2,3,4,5)} = \tau_1^{(3,4)} = q_2^{(2)} = q_3^{(4)} = \tau_3^{(5)} = \tau_4^{(8)} = 0 \), along with \( c_6 = 1 \) where \( \hat{b}_j = a_{6j} \). If \( \gamma \) is preselected to some convenient number between 1/4 and 1/3, then only one DOF remains to satisfy \( R^{(3,4,5)}_{\text{int}}(\infty) = 0 \). One may solve 18 of the 20 conditions by excluding \( p_6 = \hat{p}_5 = 0 \) and then adjusting \( c_3, c_4, c_6 \) and \( \gamma \) to find a good 5(4) pair that may not be L-stable in either of the methods. Following this procedure, no promising methods within this class have been found. If a truncated \( C(3) \) is selected, then the following 22 conditions must be satisfied: \( \tau_1^{(1,2,3,4,5)} = \hat{\tau}_1^{(4)} = q_2^{(2)} = q_3^{(3)} = b_2 = a_{62} = p_6 = \hat{p}_5 = \tau_5^{(5)} = \sum_{i=1}^{8} b_i a_{i2} = 0 \) and \( c_6 = 1 \). There are no residual DOF to set \( \gamma \) or to control \( R^{(i)}_{\text{int}}(\infty) \). The requirement \( \hat{p}_5 = 0 \) may be dropped if only A-stability in the embedded method is acceptable. Kværnø [281, 282] solves this system with \( \gamma = 0.26 \) and \( \hat{p}_5 \neq 0 \), deriving her DIRK5 having \( A^{(6)} = 0.003504 \), \( R^{(3)}_{\text{int}}(\infty) = 2.732 \), \( R^{(4)}_{\text{int}}(\infty) = 1.653 \) and \( c_3 = 1.230 \). As no promising methods have been found within the previous class, which has two DOF while this class has none, this method is not pursued further. Instead, a 5(4) pair using eight stages is considered in the next section. Ismail et al. [236] construct a 5(4) ESDIRK pair in six stages but it is not A-stable.

### 8.1.2 QESDIRK

L-stable, stiffly-accurate QESDIRKs using assumption \( C(3) \) in truncated form must satisfy 18 equations: \( \tau_1^{(1,2,3,4,5)} = q_2^{(2)} = q_3^{(3)} = b_2 = p_6 = \tau_5^{(5)} = \sum_{i=1}^{8} b_i a_{i2} = 0 \). Solving these and \( R^{(4,5)}_{\text{int}}(\infty) = 0 \) for \( c_4 \) and \( c_6 \) appears to imply \( c_3 = c_6 \). Table 26 lists the range of values for \( \gamma \) that enable L-stability. Table 19 provides the corresponding values for the embedded method. It is possible that useful methods exist within this class, but it does not appear likely.
Table 26. Bounds on $\gamma$ for L-stability of seven-stage, fifth-order QESDIRKs.

<table>
<thead>
<tr>
<th>$\gamma$</th>
<th>$c_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.183914653675175163231436</td>
<td>$2\gamma$</td>
</tr>
<tr>
<td>0.2069817179469426533972585</td>
<td>$\gamma$</td>
</tr>
<tr>
<td>0.2188084050720877043126620</td>
<td>$2\gamma/3$</td>
</tr>
<tr>
<td>0.2324427206385850277018155</td>
<td>$\gamma/3$</td>
</tr>
</tbody>
</table>

### 8.1.3 SDIRK

Sixth-stage, fifth-order, stiffly-accurate, L-stable methods offer 16 DOF to satisfy 17 order conditions. An alternative path to solving the unsimplified order condition is to invoke an incomplete specification of row simplifying assumption $C(2)$, where one is also required to satisfy 17 order conditions: $r_1^{(1,2,3,4,5)} = q_2^{(2)} = b_1 = r_4^{(5)} = \gamma$ and $\sum_{i=1}^{s} b_i a_{i1} = \sum_{i=1}^{s} b_i c_i a_{i1} = \sum_{i,j=1}^{s} b_i a_{ij} a_{j1} = 0$. Although there may exist solutions to this system, no effort will be expended in trying to find them. An alternative approach is to forsake stiff accuracy. As before, L-stable methods require $0.183914653675175163231436 \leq \gamma \leq 0.3341423670680504359540301$. Using $C(2)$ and $D(1)$, one solves $r_1^{(1,2,3,4,5)} = q_2^{(2)} = r_1^{(1)} = r_4^{(5)} = b_1 = \sum_{i=1}^{s} b_i C_i a_{i1} = p_6 = 0$ to find a pair of four-parameter family of methods in $c_3$, $c_4$, $c_5$ and $\gamma$ with $c_2 = (2 + \sqrt{2})\gamma$ and $c_6 = 1 - \gamma$. Setting $\gamma = 0.184$, it becomes a pair of three parameter family of solutions. A fourth-order embedded method may be generated by solving $\tilde{r}_1^{(1,2,3,4)} = \tilde{b}_1 = \sum_{i=1}^{s} \tilde{b}_i a_{i1} = \tilde{r}_3^{(4)} = 0$, but one must also ensure that it is at least A-stable and does not collide with the main method. The latter issues may be addressed by setting $\tilde{p}_6 = 0$ and $\tilde{r}_3^{(5)} \neq 0$. Although there are sufficient DOF to solve this problem, we will be content to inspect the analytic solution of the L-stable main method. At $\gamma = 0.184$, $c_2 = (2 + \sqrt{2})\gamma$, $c_3 = \gamma/10$, $c_4 = 1/12$ and $c_5 = 1/2$, one may find a solution having $A(6) = 0.0007703$, $\text{Eig}(M) = \{-0.9408, -0.2545, 0.2302, 0.1152, 0.001261, 0\}$ and $D = c_6 = 0.816$. Ismail and Suleiman [238] construct three non-stiffly-accurate 5(4) pairs in six stages by using assumptions $C(2)$ and $D(1)$. Each has a negative abscissa. Stability plots for RHP eigenvalues suggest their methods are likely A-stable, but nothing is stated in regard to the stability of the embedded method.

### 8.2 Sixth-Order Methods

#### 8.2.1 ESDIRK

Sixth-order, L-stable ESDIRKs in seven stages are not stiffly-accurate. They may be constructed by applying simplifying assumptions $C(2)$ and $D(1)$ or a truncated $C(3)$ and $D(1)$. More specifically, for $C(2)$ and $D(1)$, one must solve at least $r_1^{(1,2,3,4,5,6)} = q_2^{(2)} = r_1^{(1)} = 0$. One may immediately determine $c_2 = 2\gamma$ and $c_7 = 1 - \gamma$. This implies a general solution to the order and stability.
conditions which likely consists of a six parameter family in, e.g., $c_3, c_4, c_5, c_6, \gamma$ and one $a_{ij}$. For a truncated $C(3)$ and $D(1)$, one solves minimally $\tau_1^{(1,2,3,4,5,6)} = q_{3,3,4,5,6,7} = q_{3,3,4,5,6,7} = r_{1,2,3,4,5,6} = b_2 = p_6 = p_7 = r_6 = \sum_{i=1}^{6} b_i = a_{12} = 0$. It may be immediately determined that $c_2 = 2\gamma$, $c_3 = (3\pm\sqrt{3})\gamma$ and $c_7 = 1 - \gamma$. These results imply a general solution that likely consists of a pair of four-parameter families in, e.g., $c_4, c_5, c_6$ and $\gamma$. For stability, one controls $p_6$ with $\gamma$. $L$-stability is determined by the roots of $p_6 = (-1+36\gamma-450\gamma^2+2400\gamma^3-5400\gamma^4+4320\gamma^5-720\gamma^6)/360$. The only root of $p_6$ that provides $L$-stability is $\gamma = 0.3341423670680504359540301$. Notice that $c_2 = 2\gamma$ and $c_7 = 1 - \gamma$ are nearly identical (the method is very nearly confluent) if $L$-stability is chosen. This motivates the search for strongly A-stable methods and abandoning $p_6 = 0$. A-stability may be obtained for $0.284064638011792930387010 \geq \gamma \geq 0.5409068780733081049137798$. The first number corresponds to a root of the sextic portion of $E_{12} = (6\gamma - 1)(1-30\gamma+270\gamma^2-780\gamma^3+720\gamma^4)(1-36\gamma+450\gamma^2-2400\gamma^3+5400\gamma^4-4320\gamma^5+1440\gamma^6)/518400$, while the second corresponds to a root of $E_6 = (-1+48\gamma-888\gamma^2+8160\gamma^3-39600\gamma^4+97920\gamma^5-1065600\gamma^6+34560\gamma^7)/2880$.

Using $C(2)$ and $D(1)$ and $\gamma = 5/16$, $c_3 = (2 - \sqrt{2})\gamma$, $c_4 = 81/100$, $c_5 = 89/100$, $c_6 = 3/20$, all order conditions may be solved including $p_7 = R_{int}^{(3,4)}(-\infty) = 0$. To add a fourth-order, L-stable, embedded method, one must also solve $\tau_1^{(1,2,3,4,5,6)} = \tau_3^{(4)} = \hat{p}_6 = \hat{p}_7 = 0$. The resulting method, ESDIRK6(4)7A[2], is given in Table 27. It is characterized by $R_{int}^{(5)}(-\infty) = \{1, -1, 0, 0, 0.03558, 0.1676, 0.07056\}$, $R(-\infty) = -0.3766, D = 1.308, A^{(1)} = 0.002379, \text{Eig}(M) = \{-1.885, 1.036, -0.5801, -0.3379, 0.02402, 0.0003842, 0\}$, and $A^{(5)} = 0.002177$. All internal stages are I-stable. If instead, one solves a truncated $C(3)$ and $D(1)$ and sets $\gamma = 29/100$, $c_3 = (3 - \sqrt{3})\gamma$, $c_4 = 13/20, c_5 = 12/25, c_6 = 2/25$, a method may be obtained for which $R_{int}^{(5)}(-\infty) = \{1, -1, -0.7321, -0.06517, -0.3498, -0.1073, -0.04261\}$, $R(-\infty) = -0.8592, D = 6.896, A^{(7)} = 0.001982$ and $\text{Eig}(M) = \{-111.5, 2.914, -1.165, 1.062, -0.08010, 0.01706, 0\}$.

Table 27. ESDIRK6(4)7A[2].

<table>
<thead>
<tr>
<th>$b_i$</th>
<th>$0.00000000000$</th>
<th>$0.00000000000$</th>
<th>$0.00000000000$</th>
<th>$0.00000000000$</th>
<th>$0.00000000000$</th>
<th>$0.00000000000$</th>
<th>$0.00000000000$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_i$</td>
<td>$0.00000000000$</td>
<td>$0.00000000000$</td>
<td>$0.00000000000$</td>
<td>$0.00000000000$</td>
<td>$0.00000000000$</td>
<td>$0.00000000000$</td>
<td>$0.00000000000$</td>
</tr>
</tbody>
</table>
8.2.2 DIRK

Using simplifying assumptions $C(2)$ and $D(2)$, Cooper and Sayfy [116] derive a sixth-order, A-stable DIRK in six stages with $a_{11} = a_{22} = a_{33} = a_{44} = a_{55} = \gamma$; however, the sixth stage is explicit ($a_{66} = 0$). Because of the assumptions used, $b_1 = b_2 = a_{66} = 0$ and $c_6 = 1$. Additionally, one enforces $r_1^{(1)} = q_3^{(2)} = r_1^{(2)} = r_4^{(6)} = E_6 = 0$. The system of equations has two solutions, one of which is given by Cooper and Sayfy [116]. As part of this general solution, one may derive $c_3 = (2 - 5c_4 - 10\gamma + 20c_4\gamma)/(5(1 - 4c_4 - 4\gamma + 12c_4\gamma)), c_4 = (5 - 45\gamma + 80\gamma^2 \pm \sqrt{5(2 - 36\gamma + 225\gamma^2 - 560\gamma^3 + 480\gamma^4)})/(5(3 - 24\gamma + 40\gamma^2)), c_5 = (1 - 2\gamma)$ and $c_6 = 1$. A-stability is ensured by selecting the root of $E_6 = (-40 + 1200\gamma - 12000\gamma^2 + 48000\gamma^3 - 72000\gamma^4 + 28800\gamma^5)/14400$ that provides I-stability; $\gamma = 0.4732683912582953244555885$. Cooper and Sayfy provide the minus solution with $c_2 = -6547598275475/14857672547324$, $D = 1.489$, $A^{(7)} = 0.01238$ and Eig$(\mathbf{M}) = \{0.656534, -0.282101, 0.238065, -0.0652908, 0, 0\}$. With positive abcissae and one-third the error, the plus solution has $c_2 = 3496971800427/10494969937463$, $D = 17.82$, $A^{(7)} = 0.004480$ and Eig$(\mathbf{M}) = \{3.0919, -2.89351, 0.406624, -0.0578076, 0, 0\}$.

9 Seven- and Eight-stage Methods ($S_1 = 7$)

9.1 Fifth-Order Methods

9.1.1 ESDIRK

Eight-stage, stiffly-accurate, stage-order two ESDIRK methods with internal error-control

\[
\begin{array}{c|cccccccc}
\tau & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2\gamma & c_3 & (c_3 - a_{32} - \gamma) & a_{32} & 0 & 0 & 0 & 0 & 0 \\
c_4 & (c_4 - a_{42} - a_{43} - \gamma) & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 \\
c_5 & (c_5 - a_{52} - a_{53} - a_{54} - \gamma) & a_{52} & a_{53} & a_{54} & \gamma & 0 & 0 & 0 \\
c_6 & (c_6 - a_{62} - a_{63} - a_{64} - a_{65} - \gamma) & a_{62} & a_{63} & a_{64} & a_{65} & \gamma & 0 & 0 \\
1 & (1 - a_{72} - a_{73} - a_{74} - a_{75} - a_{76} - \gamma) & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & \gamma & 0 \\
1 & (1 - b_2 - b_3 - b_4 - b_5 - b_6 - b_7 - \gamma) & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & \gamma \\
b_1 & (1 - b_2 - b_3 - b_4 - b_5 - b_6 - b_7 - b_8) & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & \gamma \\
b_1 & (1 - b_2 - b_3 - b_4 - b_5 - b_6 - b_7 - b_8) & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & \gamma \\
\end{array}
\]

provide 29 DOF. Three of these are quickly dispensed via $C(2,2)$ ($a_{21} = \gamma$), $r_1^{(1)}$ ($c_7 = 1$) and $r_1^{(1)}$ ($c_8 = 1$). Methods having both L-stable main and embedded methods may be constructed by using $C(2)$ or $C(3)$ in truncated form. In the former case, one must solve the 25 conditions $r_1^{(1,2,3,4,5)} = q_3^{(2)} = r_1^{(2)} = r_4^{(6)} = E_6 = 0$. The resulting solution leaves, for example, $c_4$, $c_5$, $c_6$ and $\gamma$, but with no guarantee of I-stability for either the main or
embedded methods because \( p < s_t - 1 \) and \( \hat{p} < s_t - 2 \). If truncated assumption \( C(3) \) is chosen, then one must solve 27 conditions \( \tau_1(1,2,3,4,5) = 0 \), \( \hat{c}_2(4,5,6,7) = \tau_5(1,4) = \hat{p}_6 = R_{\text{int}}(4,5,6)(-\infty) = 0 \), leaving, e.g., \( c_6 \) and \( \gamma \). Using assumption \( C(2) \) seems like the better choice, particularly because the \( E \)-polynomial will be a function of the residual abscissae. This analysis also suggests that a QESDIRK, which uses this form of \( C(3) \), is not worth pursuing.

In principle, one can nearly solve the 25 conditions analytically. Five values of \( b_i \) may be used to satisfy \( \tau_1(1,2,3,4,5) = 0 \), while \( \hat{c}_2(4,5,6,7) = \tau_5(1,4) = \hat{p}_6 = R_{\text{int}}(4,5,6)(-\infty) = 0 \) may be used to solve for \( c_2 = 2\gamma, \ c_3 = (2 \pm \sqrt{2})\gamma, \ \gamma = 1 \) along with all \( a_{ij} \) except, e.g., \( a_{76} \). What remains is \( a_{76} \) and two \( b_i \) to satisfy \( \hat{p}_7 = \hat{p}_6 = R_{\text{int}}(6)(-\infty) = 0 \). An L-stable, externally embedded method may also be included by solving \( \hat{\tau}^0_1(1,2,3,4,5) = \hat{p}_7 = \hat{p}_7 = 0 \) along with \( \hat{\tau}^5_1 = 1/5000 \). Selecting \( \gamma = 1/4 \) to keep both \( E_6 \) and \( \hat{E}_6 \) positive, we have \( A(6) = 0.002690, \ D = 9.743, \ R_{\text{int}}(-\infty) = \{1,-1,0,0,0,0,0.06771,0,0\} \) and \( E_{\hat{E}}(M) = \{ -105.0, -3.300, 1.573, 0.06294, 0.06250, -0.05885, -0.01757, -0.01039 \} \). Each internal stage is I-stable and, hence, stages 3, 4, 5 and 7 are L-stable while stage 6 is nearly L-stable. The internal embedded method has \( \hat{A}(6) = 0.001385, \hat{B}(6) = 3.832 \) and \( \hat{E}(6) = 1.943 \), while the externally embedded method has \( \hat{A}(5) = 0.002739, \hat{B}(6) = 1.084 \) and \( \hat{E}(6) = 0.9824 \). ESDIRK5(4I)8L[2]SA is given in Table 28. Skvortsov [398] generates a fifth-order ESDIRK which is L(\( \alpha \)) and \( \alpha = 72.3 \) degrees. He also derives a sixth-order ESDIRK in eight implicit stages which is also L(\( \alpha \)) and \( \alpha = 88.7 \) degrees with \( \gamma = 1/6 \).

Table 28. ESDIRK5(4I)8L[2]SA.

<table>
<thead>
<tr>
<th>( b_i )</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{b}_i )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In Table 28, Skvortsov [398] generates a fifth-order ESDIRK which is \( L(\alpha) \) and \( \alpha = 72.3 \) degrees. He also derives a sixth-order ESDIRK in eight implicit stages which is also \( L(\alpha) \) and \( \alpha = 88.7 \) degrees with \( \gamma = 1/6 \).


10 Test Problems

Testing of schemes is conducted on three singular perturbation problems: Kaps’ problem, van der Pol’s equation and the Pareschi and Russo problem. The equations exhibit increasing stiffness as \( \varepsilon \rightarrow 0 \) and, in the limit of \( \varepsilon = 0 \), the systems become differential algebraic equation systems. Outside of singular perturbation problems, other simple test equations for stiff problems may be found in the literature [143, 193, 288, 336, 378]. Anyone testing using the DETEST problem set should consult comments made by Shampine [374].

The first singular perturbation problem considered is Kaps’ problem [129]

\[
\dot{y}_1(t) = -(\varepsilon^{-1} + 2) y_1(t) + \varepsilon^{-1} y_2^2(t), \quad \dot{y}_2(t) = y_1(t) - y_2(t) - y_2^2(t), \quad (266)
\]

where \( 0 \leq t \leq 1 \) and whose exact solution is \( y_1(t) = y_2^2(t) \), \( y_2(t) = \exp(-t) \). Equilibrium (unperturbed) initial conditions (ICs) are given by \( y_1(0) = y_2(0) = 1 \). In the limit of \( \varepsilon = 0 \), the system becomes an index-1 differential algebraic equation system.

Van der Pol’s (vdP) equation [190, 193], the second problem, is given by,

\[
\dot{y}_1(t) = y_2(t), \quad \dot{y}_2(t) = \varepsilon^{-1} (1 - y_1(t)^2) y_2(t) - y_1(t). \quad (267)
\]

Unperturbed ICs are given by [172, 193] \( y_1(0) = 2 \), \( y_2(0) = -2/3 + 10\varepsilon/81 - 292\varepsilon^2/2187 - 1814\varepsilon^3/19683 + \cdots \). In the limit of \( \varepsilon = 0 \), the system becomes an index-1 differential algebraic equation system. At large \( \varepsilon \), this series expansion fails. However, nonequilibrium ICs will be specified for this equation by using this expansion and large values of \( \varepsilon \).

Pareschi and Russo [336] have constructed a simple singular perturbation equation

\[
\dot{y}_1(t) = -y_2(t), \quad \dot{y}_2(t) = y_1(t) + \varepsilon^{-1} (\sin(y_1(t)) - y_2(t)). \quad (268)
\]

Equilibrium ICs are \( y_1(0) = \pi/2 \), \( y_2(0) = 1 \). Nonequilibrium, or perturbed, data may be specified by replacing the condition on \( y_2 \) with \( y_2(0) = 1/2 \). Further comparisons between integration methods including DIRK-type schemes are available in the literature. In comparisons with other popular schemes, Hairer and Wanner [193] find the performance of the stage-order one SDIRK4 disappointing. This prompted Cash [100] not even to include SDIRK methods in his stiff solver comparison. However, comparisons conducted by Gaffney [161] and Sandu et al. [353] cast SDIRK methods in a more favorable light. With Navier-Stokes applications in mind, Bijl et al. [38, 39] show that ESDIRK schemes outperform SDIRK and BDF-methods in simulating laminar flows over airfoils and in complex multi-disciplinary flows involving fluid-structure interactions. Jothiprasad et al. [248] conduct a similar study by focusing particular attention on solution procedures for the coupled nonlinear stage and step equations. Carpenter et al. [93, 94] compare ESDIRK, BDF and MEBDF methods on flows over airfoils by using moderately stiff turbulence models. They find that ESDIRK methods are accurate even with very large timesteps, but often become decreasingly efficient in that limit. This inefficiency can result primarily
from dramatic slowdown of the algebraic equation solver. A partial remedy would be the development of effective stage-value-predictors, so that ESDIRKs do not spend inordinate amounts of time converging stage and step iterations. Clemens et al. [108] compare a 3(2) SDIRK and Rosenbrock pair while solving nonlinear equations associated with magnetic fields. Conclusions from all of these tests need to be made while considering leading-order error of the methods, stage-order, local integration error tolerance, equation stiffness and stage-value prediction techniques.

11 Discussion


It should be kept in mind that one very important aspect of these methods has
been largely forsaken: stage-value predictors. A thorough investigation of construction choices and of testing the best approaches to stage-value predictor design is a very large undertaking beyond the scope of this paper. We shall be content with the analysis provided in §2.18. However, one should expect that higher stage-order, as well as good internal stability, will facilitate quality predictors. No testing is conducted on discontinuities. Due to the very large number of graphs that would be required to support a thorough examination of each scheme’s behavior on all performance attributes considered, a strictly verbal description follows.

11.1 Convergence

All convergence tests are done using the three singular perturbation test problems, with the same ICs for each test problem, and integrating them over the same time interval. Step-sizes are held constant. Singular perturbation problems are very useful for detecting shortcomings of methods but may not be particularly illuminating at distinguishing which well designed method is more accurate in practical settings. They also fail to exercise many important implementation matters. For these reasons, we will be content to offer qualitative assessments of scheme convergence performance rather than quantifying performance in such simple settings.

11.1.1 Second-Order Methods

The two second-order methods tested in this paper were ESDIRK2(1)3L[2]SA and SDIRK2(1)2A[1]Alg. In terms of leading order accuracy, SDIRK2(1)2A[1]Alg is more accurate with $A^{(3)} = 0.02329$ versus $A^{(3)} = 0.05719$. Method ESDIRK2(1)3L-[2]SA is stiffly-accurate, L-stable and has a stage-order of 2. In contrast, SDIRK2(1)2A[1]Alg is algebraically stable but not L-stable or stiffly-accurate, has a stage-order of 1 and is strongly regular. Each has a similar value of $\gamma$ (0.2500 versus $\approx 0.2929$). Since the second stage of ESDIRK2(1)3L[2]SA is essentially a shortened Trapezoidal rule, the stage is neutrally stable to stiff eigenvalues.

At $\epsilon = 0$, the SDIRK is predictably more accurate than the ESDIRK on the vdP equation. As stiffness increases, each method behaves differently. For the SDIRK, the differential variable becomes more accurate while retaining second order accuracy. However, the algebraic variable becomes substantially less accurate, its order reduces slightly and it has a rather erratic convergence behavior at $\epsilon = 10^{-3}$. Convergence behavior of the ESDIRK is smoother, suffers from no order reduction and does not experience as large an increase in algebraic variable error as stiffness is applied. For very large step-sizes, the convergence behavior of the ESDIRK is erratic. On Kaps’ and the Pareschi and Russo problem, stiffness increases have less of an influence on the methods than when solving the vdP equation. SDIRK2(1)2A[1]Alg still has more difficulty with the algebraic variable. This is made dramatically apparent using nonequilibrium ICs on the Pareschi and Russo problem. Stage-order 2 plus L-stability seems to be more useful than stage-order 1 plus algebraic stability and strong regularity when solving stiff problems, particularly with nonequilibrium ICs.
11.1.2 Third-Order Methods

Five third-order methods have been tested: ESDIRK3(2)4L[2]SA, ESDIRK3(2I)4L- [2]SA, ESDIRK3(2)5L[2]SA, ESDIRK3(2I)5L[2]SA and ESDIRK3(2)5L[2]SA_SOD. Each method is L-stable, stiffly-accurate and has a stage order of 2. Method ESDIRK3(2)5L[2]SA_SOD is different from the others in that it invokes $q_{3,4}^{(3)} = 0$ and has a second-order damped stability function. The four-stage methods are both characterized by $\gamma \approx 0.4359$ and have $R^{(3)}_{\text{int}} = -0.8057, (-0.9567)$ and $A^{(4)} = 0.03663, (0.4907)$ for the method without (with) internal error-control. Similarly, for the five stage methods without SOD, $\gamma = 0.225$ and have $R^{(3,4)}_{\text{int}} = 0, 0, (17/81, 0)$ and $A^{(4)} = 0.0007769, (0.01149)$ for the method without (with) internal error-control. ESDIRK3(2)5L[2]SA_SOD has $\gamma \approx 0.3025$, $R^{(3,4)}_{\text{int}} = \{1 - \sqrt{3}, 0\}$ and $A^{(4)} = 0.01806$.

The behavior of third-order methods strongly depends on whether one is solving the difficult vdP equation, the easier Kaps’ problem, or the even milder Pareschi and Russo problem. In the most severe circumstances, solving the vdP equation, ESDIRK3(2)5L[2]SA and ESDIRK3(2I)5L[2]SA are decidedly better performing methods than the others. ESDIRK3(2)5L[2]SA_SOD has severe solvability problems that render it rather useless at stiffness levels of $\varepsilon \leq 10^{-3}$. Both ESDIRK3(2)4L- [2]SA and ESDIRK3(2I)4L[2]SA methods suffer from solvability problems at larger step-sizes and $\varepsilon \leq 10^{-3}$. Between ESDIRK3(2)5L[2]SA and ESDIRK3(2I)5L[2]SA, internal error control is obtained at a substantial accuracy penalty as ESDIRK- 3(2)5L[2]SA is decidedly more accurate. Also, ESDIRK3(2)5L[2]SA appears to superconverge, to order $p + 1$, at large time steps for the differential variable. None of the other 3(2) pairs exhibits superconvergence. Order reduction of each of the five 3(2) pairs is modest and is generally confined to the algebraic variable. On the easier problems, order reduction was nearly nonexistent except for ESDIRK3(2I)5L[2]SA on Kaps’s problem at $\varepsilon = 10^{-3}$. All methods converge at essentially third-order, independent of stiffness. ESDIRK3(2)5L[2]SA exhibits superconvergence on the Pareschi and Russo problem at large step sizes. The most obvious difference between methods that do or do not have internal error-control is the performance of the algebraic variable, but this difference is not manifested consistently. Methods ESDIRK3(2I)5L[2]SA and ESDIRK3(2)4L[2]SA have more erratic convergence of their algebraic variables. This may suggest that something other than the internal error-control, and the attendant confluent abscissae, is influencing matters. Of all five 3(2) pairs, only ESDIRK3(2)5L[2]SA_SOD cannot be recommended. Redesigning it without invoking $q_{3,4}^{(3)} = 0$ might make for better SOD methods.

11.1.3 Fourth-Order Methods

Fourteen are stage-order two (Q)ESDIRKs and four are stage-order one (Q)SDIRKs. Twelve of the 18 methods have $\gamma = 1/4$.

To test stage-order one methods, the two SDIRK4 methods of Hairer and Wanner [193] are used as well as two methods employing $q_{2,3,4} = 0$: SDIRK4(3)5L[1]SA.C(2) and QSDIRK4(3)5L[1]SA. These four methods have $\gamma = 1/4$, except QSDIRK4(3)5L[1]SA where $\gamma = 8/25$. Leading order errors for the methods SDIRK4(1), SDIRK4(2), SDIRK4(3)5L[1]SA.C(2) and QSDIRK4(3)5L[1]SA are $A^{(5)} = 0.002504, 0.004229, 0.002454, 0.006849$, respectively. Amongst the 14 stage-order two methods with $a_{11} = 0$, two are five-stage methods: ESDIRK4(3)5L[2]SA and ESDIRK4(3I)5L[2]SA. Both have $\gamma \approx 0.5728$ and nonzero values of $R_{\text{int}}^{(3,4)}(\infty)$. Leading order errors for these two methods, $A_{\text{int}}^{(5)} = 0.03857(0.04506)$ for the method without (with) interior error control, with four implicit stages, are often substantially higher than the remaining 12 stage-order two methods with five implicit stages. Two of the methods are QESDIRKs with $a_{22} = \gamma/3$, $\gamma = 0.3200$ (QESDIRK4(3)6L[2]SA), and $\gamma = 0.4803$ (QESDIRK4(3)6L[2]SA.SOD), $q_{3,4,5} = 0$, and each with the relatively large values of $A^{(5)} = 0.004828$ and $A^{(5)} = 0.05848$, respectively. Two of the methods have second order damped stability functions with $\gamma \approx 0.3889$ (ESDIRK4(3)6L[2]SA.SOD with $A^{(5)} = 0.01670$) and $\gamma \approx 0.4803$ (QESDIRK4(3)6L[2]SA.SOD), along with $q_{3,4,5} = 0$. Five methods modeled after ESDIRK4(3)6L[2]SA, ESDIRK4(3)6L[2]SA.A,B,C,D,E, are used to test the consequences of setting $\left| R_{\text{int}}^{(3,4,5)}(\infty) \right| = n/2$ where $n = 1, 2, \cdots, 5$. All have similar leading order errors (See Table 18). Two methods are constructed with both internal and external error-control, ESDIRK4(3I)5L[2]SA and ESDIRK4(3I)6L[2]SA, and may be compared against their similar and more traditional analogs, ESDIRK4(3)5L[2]SA and ESDIRK4(3)6L[2]SA. Leading order error for the four methods is $A^{(5)} = 0.04506, 0.002254, 0.03857, 0.001830$, respectively. Methods ESDIRK4(3)6L[2]SA ($A^{(5)} = 0.001830$), ESDIRK4(3)6L[2]SA.Ark ($A^{(5)} = 0.003401$) and ESDIRK4(3)6L[2]SA.C(3) ($A^{(5)} = 0.002970$) are each straightforward methods that could potentially be applied widely. While each is stage-order two, ESDIRK4(3)6L[2]SA.C(3) also satisfies $q_{3,4,5} = 0$. Both ESDIRK4(3)6L[2]SA and ESDIRK4(3)6L[2]SA.C(3) have fully damped internal stability functions at stages where this is possible.

Beginning with the easiest problem, no discernable order reduction is seen for any of the stage-order two methods on the Pareschi and Russo problem with equilibrium ICs. Order reduction in stage-order one methods was mild and occurred at very tight tolerances with $\varepsilon = 10^{-3}$. Use of $q_{2,3,4} = 0$ appeared to mildly mitigate order reduction on the stage-order one methods. Methods ESDIRK4(3)6L[2]SA.A,B,C,D,E applied using nonequilibrium ICs showed worse behavior of the algebraic variable relative to ESDIRK4(3)6L[2]SA and other stage-order two methods. Nonequilibrium ICs induce order reduction of the algebraic variable at small step-sizes on all methods. QESDIRK4(3)6L[2]SA had the most accurate algebraic variable at moderate to large step sizes. (Q)SDIRK methods using $q_{2,3,4} = 0$ had less erratic convergence behavior of the algebraic variable at $\varepsilon = 10^{-3}$ and small
step-sizes than the other stage-order one methods. Order reduction for all methods was mild, and little difference existed between stage-order two and stage-order one methods. All in all, the Pareschi and Russo problem is not terribly demanding of the integration method. For Kaps’ problem, order reduction is more pronounced at $\varepsilon = 10^{-3}$ than it is on the Pareschi and Russo problem. Schemes used to test the effects of internal instability, ESDIRK4(3)6L[2]SA_A, B, C, D, E, have erratic convergence rates of the algebraic variable that grow increasingly erratic at small step-sizes and as $n$ increases with $|R_{\text{int}}^{(3,4,5)}(-\infty)| = n/2$, $n = 1, 2, \cdots, 5$. QESDIRK4(3)6L[2]SA appears to offer no advantage over ESDIRK4(3)6L[2]SA on Kaps’ problem. Adding a truncated version of assumption $C(3)$ in the form $q_{3,4,5}^{(3)} = 0$ with ESDIRK4(3)6L[2]SA_C(3) also offers no benefit. QSDIRK4(3)5L[1]SA clearly improves algebraic variable error at large step sizes over SDIRK(1), SDIRK(2) and SDIRK4(3)5L[1]SA_C(2). Stage-order two methods have dramatically better algebraic variable error at $\varepsilon = 10^{-3}$ than stage-order one methods. This also applies to $\varepsilon = 10^{-6}$ at small step sizes. On the most difficult problem, the vdP equation, QESDIRK4(3)6L[2]SA has slightly better algebraic variable error than ESDIRK4(3)6L[2]SA at lax error tolerances. Otherwise, methods ESDIRK4(3)5L[2]SA, ESDIRK4(3I)5L[2]SA, ESDIRK4(3)6L[2]SA, ESDIRK4(5)6L[2]SA, SDIRK(1) and SDIRK(2) and SDIRK4(3)5L[1]SA_C(2). Stage-order two methods have dramatically better algebraic variable error at $\varepsilon = 10^{-3}$ than stage-order one methods. This also applies to $\varepsilon = 10^{-6}$ at small step sizes.

11.1.4 Fifth- and Sixth-Order Methods

Convergence rate testing of higher-order ESDIRKs involves three fifth-order methods: ESDIRK5(3)6L[2]SA, ESDIRK5(4)7L[2]SA, ESDIRK5(4I)8L[2]SA and one sixth-order method: ESDIRK6(4)7A[2]. All methods are stage-order two, but only the fifth-order methods are stiffly-accurate and L-stable. The sixth-order method is strongly A-stable. Values of $\gamma$ and $A^{(6)}$ for the fifth-order methods are $\gamma \approx 0.2781$, 0.1840, 0.2500, and $A^{(6)} = 0.004615$, 0.001846, 0.002690, respectively. For ESDIRK6(4)7A[2], $\gamma = 0.3125$, $R(-\infty) = -0.3766$, and $A^{(7)} = 0.002379$. For stages 3, 4, 5 and 6, only ESDIRK5(3)6L[2]SA is not fully damped in the far LHP.

is still quite accurate. Methods ESDIRK5(4)7L[2]SA and ESDIRK5(3)6L[2]SA behave similarly, but ESDIRK5(4I)8L[2]SA is less accurate than either. Kaps’ problem is severe enough to cause the algebraic variable error in method ESDIRK6(4)7A[2] to seriously degrade with increasing stiffness rather than simply near \( \varepsilon \approx 10^{-3} \). However, the differential variable is still resolved better with this method than the other three methods. Methods ESDIRK5(4)7L[2]SA and ESDIRK5(3)6L[2]SA are of similar accuracy, but ESDIRK5(4)7L[2]SA is more accurate for very stiff modes while each is more accurate than ESDIRK5(4I)8L[2]SA. Solvability problems associated with ESDIRK5(3)6L[2]SA on the vdP equation at stiffnesses \( \varepsilon \leq 10^{-3} \) make its use inadvisable on such nasty problems. Both ESDIRK5(4)7L[2]SA and ESDIRK5(4I)8L[2]SA work relatively well, with the former being more accurate. Without L-stability and stiff accuracy, ESDIRK6(4)7A[2] not only has a very inaccurate algebraic variable for nontrivial stiffnesses, it also loses accuracy in the differential variable at \( \varepsilon \approx 10^{-3} \). It is an interesting question to ask how much the lack of internal I-stability on stage 6 influenced the performance of ESDIRK5(4)7L[2]SA on these tests.

11.2 Solvability

Generally, the maximum step-size possible with a DIRK-type method is given by \( \nu(\Delta t) < \min (a_{ii})^{-1} \). Beyond this value of \( (\Delta t) \), one is no longer assured a unique solution to the coupled nonlinear stage and step equations. This relation implies that reduced values of \( a_{ii} \) enable larger maximum step-sizes. However, testing of the many methods shows that this maximum step-size depends on more than \( \nu \) and \( a_{ii} \). While no attempt was made to compute \( \nu \) and compare results to the expression \( \nu(\Delta t) < \min (a_{ii})^{-1} \), one may look into whether stiffness has any affect on the maximum step-size. This stiffness dependence only becomes clearly apparent when methods are tested on the more severe vdP equation but not on Kaps’ problem or Pareschi and Russo’s problem. However, no investigation is made about solvability behavior on large systems of equations. Small anomalies occur on Kaps’ problem for \( 10^{-1} \leq \varepsilon \leq 10^{0} \) with several methods. Switching from equilibrium to nonequilibrium ICs on Pareschi and Russo’s problem has no noticeable affect. By solving the vdP equation, one may place the 29 methods into one of three categories: essentially unaffected, moderately affected and severely affected. Thirteen of the 29 were essentially or totally unaffected: SDIRK2(1)2A[1]Alg, ESDIRK3(2)5L[2]SA, ESDIRK3(2I)5L-\[2\]SA, ESDIRK4(3I)5L[2]SA, ESDIRK4(3)6L[2]SA, ESDIRK4(3)6L[2]SA\_C(3), ESDIRK4(3I)6L[2]SA, QESDIRK4(3)6L[2]SA, ESDIRK4(3)6L[2]SA\_ARK, SDIRK4(3)5L[1]SA\_C(2), QSDIRK4(3)5L[1]SA, ESDIRK5(4I)8L[2]SA and ESDIRK-6(4)7A[2]. Those methods that were moderately affected are: ESDIRK2(1)3L[2]SA, ESDIRK3(2)4L[2]SA, ESDIRK3(2I)4L[2]SA, ESDIRK4(3)5L[2]SA, ESDIRK5(4)7L-\[2\]SA, ESDIRK4(3)6L[2]SA\_SOD and ESDIRK4(3)6L[2]SA\_B. Methods whose maximum step-size was strongly affected by stiffness are: ESDIRK4(3)6L[2]SA\_A,C,D,E, ESDIRK3(2)5L[2]SA\_SOD, QESDIRK4(3)6L[2]SA\_SOD, ESDIRK5(3)6L[2]SA, SDIRK4(1) and SDIRK4(2). It is not obvious why certain methods behaved differently from others in this test. Of the five methods designed specifically to investigate
the affects of internal instability, four of them were severely affected by stiffness while the fifth was only moderately affected. We tentatively conclude that internal stability is an important attribute to possess for solvability. Note that each of these five methods controls $R^{(i)}_{\text{int}}(\infty)$, $i = 3, 4, 5$ but not $R^{(i)}_{\text{int}}(z)$, $i = 3, 4, 5$. Methods having second-order damped, L-stability fared poorly on this test. One may speculate that the highly constrained second-order damped methods with $q_{3,4,\ldots,(s-1)}^{(3)} = 0$ possessed poor internal stability away from $z = -\infty$. This hypothesis is not actually investigated. As method ESDIRK5(3)6L[2]SA has nonzero values of $R^{(i)}_{\text{int}}(\infty)$, $i = 3, 4, 5$, it is also suspected that poor performance of the method is caused by insufficient damping of stiff modes in the internal stages. Lastly, the behavior of the four fourth-order (Q)SDIRK methods seemed to depend on whether they were designed using simplifying assumption $C(2)$ in the truncated form $q^{(2)}_{2,3,4} = 0$. Methods SDIRK-4(3)5L[1]SA_C(2) and QESDIRK4(3)5L[1]SA behaved much better than SDIRK4(1) and SDIRK4(2) on this solvability test. While internal stability is not ruled out, it appears that an incomplete specification of assumption $C(2)$ benefits stage-order one methods.

11.3 Error Estimation and Step-Size Control

Step-size selection based on an error estimate from an embedded method and an error controller, but not in conjunction with iteration control, is tested on the three test problems with both equilibrium and nonequilibrium initial conditions. Ideally, the error controller delivers a local error nearly equal to the error tolerance along with a smooth step-size sequence. As there are 27 $p(p-1)$ pairs being tested and numerous error-controllers available, the first test is to see how each method behaves with the established controllers: I, H211, PC, PID, H312, PPID, H321, H0321 and H0330. Each method with each controller is used to solve the vdP equation on the same time interval using equilibrium ICs and $\varepsilon = 10^{-5}$. Methods having stage-order one and $a_1 \neq 0$ were all rather insensitive to the controller choice and delivered an error slightly larger than the requested error. Among these methods, imposing $q_{i}^{(2)} = 0$, $i = 2, 3, 4$ modestly improved the ability of the controllers to deliver the requested error tolerance. It should be noted that the embedded methods used with SDIRK4(1) and SDIRK4(2) were the revised A-stable methods and not the original methods. Stage-order two ESDIRKs and QESDIRKs behaved quite differently. Their delivered local error, relative to the requested error, was strongly dependent on the controller choice. From the published controllers, ESDIRKs and QESDIRKs worked well with only the PPID and H321 controllers. Each is from the class of controllers having third-order dynamics, second-order adaptivity and first-order filtering but having different roots to the characteristic polynomial. Expanding to $H_{0PDAPF}$ controllers having $PD = PA + PF$ and $PD = 4, 5$ produced no useful controllers for these methods on the vdP equation. It would seem that nonzero roots to the controller are essential for good behavior in this context. This may explain why H312, which has the roots $(0, 0, 1/2)$, did not perform well. The next test also uses the vdP equation with equilibrium ICs but uses only method ESDIRK4(3)6L[2]SA. Both H321_{general} and H312_{general} controllers are investigated.
to find the affect of the characteristic root placement. The behavior of the con-
trollers is strongly influenced by root placement. Several roots were found during a
course search, which provide a high-quality error controller for both H321\textsubscript{general} and
H312\textsubscript{general} with ESDIRK4(3)6L[2]SA: $(q_1, q_2, q_3) = (-0.5, 0.7, 0.8), (0.3, 0.3, 0.8),
(0.4, 0.5, 0.6), (0.4, 0.5, 0.7)$ and $(0.6, 0.6, 0.6)$. Again, H321 and PPID have roots
given by $\{1/3, 1/2, 2/3\}$ and $\{0.4838 \pm 0.3275I, 0.7325\}$, respectively. Roots having
complex conjugate pairs were not investigated. It appears that good controllers
having only real roots have roots $-0.5 \leq q_i \leq 0.8$. Further tests showed that an
H321\textsubscript{general} or H312\textsubscript{general} controller with ESDIRK4(3)6L[2]SA that worked well on
the vdP equation having equilibrium ICs also worked well on the other test prob-
lems, including those with nonequilibrium ICs. However, controllers that worked
well with one ESDIRK could behave poorly with a different ESDIRK. The ability
of any H321\textsubscript{general} or H312\textsubscript{general} controller to deliver the requested error tolerance
is often similar, given the same roots to the characteristic polynomial of the con-
trollers. For the ESDIRKs and QESDIRKs, the controllers seem to be sensitive
to the method but not the problem. This is not due to matters such as internal
stability. Embedded method construction is guided by parameters defining a good
embedded method for an ERK. It is possible that the parameters, $B(\hat{p}+2)$,
$C(\hat{p}+2)$ and $E(\hat{p}+2)$, (29), should be carefully chosen using different criteria or that another
parameter is needed for the design of ESDIRK and QESDIRK embedded methods.
The uniform lack of sensitivity of SDIRK and QSDIRK methods to controller choice
is puzzling in view of the consistent sensitivity of the ESDIRK and QESDIRK to
controller choice (using externally or internally embedded methods).

Another attribute of a good controller is that it minimizes the work required
to complete an integration by not prompting rejected steps. Controllers that con-
sistently delivered an error that is significantly larger than the requested error are
prone to have large numbers of rejected steps and are very inefficient at delivering
accurate solutions. All of the good controllers yield similar efficiency on the vdP
equation over the entire spectrum of target errors. A common technique in dealing
with a rejected step is to simply halve (or some other factor) the step-size and be-
gin the step again. Doing this rather than computing a new step-size that would
deliver the requested error based on the order of the method could interfere with
the operation of the error controller and might best be avoided. The delivered error
on the vdP equation at large step-sizes was very erratic for the ESDIRK schemes.
This was partially attributable to rounding error.

Tests using ESDIRK methods that possess both internally and externally em-
bedded methods, show similar results in all situations. For the methods presented
in this paper applied to stiff ODEs, it would seem unlikely that the methods having
internally embedded methods offer any tangible benefit over traditional methods.
However, no attempt was made to utilize both simultaneously for something like
stiffness estimation or order reduction detection.

It is sometimes remarked that an SDIRK has an advantage over an ESDIRK
when the ICs to the step are effectively nonequilibrium because the ESDIRK in-
corporates the function evaluation of this IC into future values of the integration
variable [246]. ESDIRK and QESDIRK methods designed in this paper did not
seem to have any more difficulty with nonequilibrium ICs than SDIRKs. One could always integrate the very first step using an SDIRK if this should ever be a problem.

No attempt has been made to investigate error-control strategies for \( p(p – 2) \) pairs.

### 11.4 Dense Output

The dense output for all 15 new schemes is tested using the vdP equation with nonequilibrium ICs and and \( \varepsilon = 10^{-3} \). The initial condition is advanced one time step with interpolation done at the points around \( t_{ref} + (\Delta t) \). The dense output is then compared to a numerically “exact” solution obtained using \( (\Delta t)/3 \) and run to the dense output time. A refinement study performed using only one time-step in the variable \( \Delta t \) determines the local order of accuracy of the dense output. Note that the nature of the refinement study in the variable \( \Delta t \) determines the local error of the dense output and, hence, the derived order is the local order or the global order plus one. Table 29 summarizes a study using the fourth-order formula associated with ESDIRK4(3)6L[2]SA scheme. The interpolated values are predicted at \( \frac{2}{3} \Delta t \). Design order \( (p^* \text{ in appendix B}) \) is asymptotically achieved. Similar results showing design order dense output were obtain for all 14 other new schemes.

Table 29. Convergence rate of dense output in interpolation mode as calculated with the ESDIRK4(3)6L[2]SA scheme.

<table>
<thead>
<tr>
<th>( \Delta t )</th>
<th>error</th>
<th>local order</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25056E-02</td>
<td>0.75866E-04</td>
<td>2.11640</td>
</tr>
<tr>
<td>0.14094E-02</td>
<td>0.16841E-04</td>
<td>2.79227</td>
</tr>
<tr>
<td>0.79280E-03</td>
<td>0.24970E-05</td>
<td>3.48633</td>
</tr>
<tr>
<td>0.44595E-03</td>
<td>0.26161E-06</td>
<td>4.05169</td>
</tr>
<tr>
<td>0.25085E-03</td>
<td>0.21345E-07</td>
<td>4.44270</td>
</tr>
<tr>
<td>0.14110E-03</td>
<td>0.14857E-08</td>
<td>4.68427</td>
</tr>
<tr>
<td>0.79370E-04</td>
<td>0.94198E-10</td>
<td>4.82379</td>
</tr>
<tr>
<td>0.44646E-04</td>
<td>0.56668E-11</td>
<td>4.90191</td>
</tr>
<tr>
<td>0.25113E-04</td>
<td>0.33113E-12</td>
<td>4.94449</td>
</tr>
<tr>
<td>0.14126E-04</td>
<td>0.18920E-13</td>
<td>4.98420</td>
</tr>
</tbody>
</table>

### 12 Conclusions

This paper presents a comprehensive review of DIRK-type methods applied to first-order ODEs. The goal of this applied review is to summarize the characteristics, assess the potential, and then design several nearly optimal, general purpose, DIRK-type methods. Over 20 important aspects of DIRK-type methods are reviewed: general structure, order conditions, simplifying assumptions, error, linear stability, nonlinear stability, internal stability, dense output, conservation, symplecticity,
symmetry, dissipation and dispersion accuracy, memory economization, regularity,
boundary and smoothness order reduction, efficiency, solvability, implementation,
step-size control, iteration control, stage-value predictors, discontinuities and existing
software. Following this, an historical look is made into the earliest examples of
DIRK-type methods. An exhaustive design study is then conducted on DIRK-type
methods having from two to seven implicit stages. Extending previous results and
deriving many new methods, one is able to clearly see what is possible with DIRK-
type methods and at what cost. From this detailed investigation of methods, 15
schemes are selected as being promising for general purpose application. Based on
the review of method characteristics, these methods focus on having a stage-order
of two, stiff accuracy, L-stability (except ESDIRK6(4)7A[2]), high quality embed-
ded and dense-output methods, small magnitudes of the algebraic stability matrix
eigenvalues, small values of $a_{ii}$, and small or vanishing values of the internal stability
functions for large eigenvalues. These choices are also consistent with maximizing
scheme efficiency. As stage-order governs the severity of order reduction, focusing
on stage-order two methods facilitates accuracy. Little effort is expended pursuing
algebraic stability, symplecticity, symmetry, memory economization, minimal phase
error or regularity. Due to the scope of the stage-value predictor topic, this import-
tant subject is comprehensively reviewed, but no attempt is made to create optimal
predictors. Similarly, precise event detection for unforeseen discontinuities in not
pursued.

Testing of the 15 chosen methods is done on three singular perturbation prob-
lems to ensure that the methods behave well and have no obvious shortcomings. To
supplement the 15 methods, 14 additional methods are created or taken from the
literature, in part, to shed light on design choices. Several tentative conclusions may
be made from these tests on singular perturbation problems. Stage-order two meth-
ods are better for stiff problems than stage-order one methods in spite of not being
able to possess algebraic stability. Stiffly-accurate and L-stable methods work better
than algebraically- and A-stable methods, particularly when using nonequilibrium
ICs. Observed order reduction is very problem dependent. Methods exhibited little
order reduction on the Pareschi and Russo problem, moderate order reduction on
Kaps’ problem, and severe order reduction on the vdP problem. Therefore, the
choice of the optimal method is to some extent a function of the problem severity.
Solvability behavior of a scheme appears to be a function of not only the values
of $a_{ii}$ and $\nu$ but also is a function of the internal stability characteristics of the
method. Incomplete application of simplifying assumption $C(3)$ to methods do not
appear to fortify the methods and, in preliminary tests, seemed to undermine the
stage-value predictor. This may be attributable to the influence of $C(3)$ on $R_{int}^{(i)}(z)$. However, incomplete application of simplifying assumption $C(2)$ to SDIRKs and
QSDIRKs does appear to provide some fortifying effect on the methods relative to
SDIRK(1) and SDIRK(2). Attempting to achieve quasi stage-order three behavior
by satisfying all of the the design conditions for QESDIRKs results in strained meth-
ods having large Runge-Kutta coefficients, large leading-order errors as well as very
large negative eigenvalues of the algebraic stability matrix. They exhibit slightly
better accuracy for the algebraic variable at large step-sizes. The performance of
methods having second-order damped stability functions was disappointing but may be partly due to the incorporation of $q_{3,4,...,(s-1)}^{(3)} = 0$. Even without this, the values of $\gamma$ are unpleasantly large.

Error controller choice for the (Q)SDIRKs tested is largely insensitive to both the method and the problem. However, controllers are far more sensitive to which (Q)ESDIRK is being used but they are still rather insensitive to the problem. Controllers for 27 of the 29 DIRK-type methods tested are probably best chosen from the $H_{312_{\text{general}}}$ and $H_{321_{\text{general}}}$ classes when solving stiff problems. For the stage-order two (Q)ESDIRKs, the controller performance is very strongly dependent on the location of the characteristic polynomial roots. The root placement must be tuned for each (Q)ESDIRK, but probably not for each problem. There was no situation encountered during testing where the use of an internally embedded method offered any discernable benefit over an externally embedded method with the five methods which possessed both. As the incorporation of an internally embedded method diminishes the main method, the embedded method, the dense output method and the stage-value predictors by virtue of the method being confluent, it appears to be an attribute best not included. In a DAE context, a different conclusion might be reached. No effort has been made to further the topic of error-control strategies for $p(p-2)$ pairs.

Users wishing to have a high quality, general purpose DIRK-type method might consider trying ESDIRK4(3)6L\[2\]SA, Table 16, in conjunction with the $H_{321}$ or PPID error-controller, Table 8, and the dense output offered in appendix B as a default method. Some experimentation with controller roots might prove useful. Tentative stage-value predictors could use the dense-output method for stage two and analogs of (202), (203) and (204) for later stages. Other ESDIRK methods may be investigated as time and effort permits. If the problem at hand is not particularly demanding then the higher-order methods could also be considered if error tolerances are sufficiently demanding.

While it appears that many matters relevant to DIRK-type methods are quite mature, several topics seem unfinished. Two that readily come to mind are improved stage value prediction techniques and improved iteration control for both direct and indirect solvers. Each of these has the potential to substantially improve the efficiency of integrators. Lastly, we pose several questions that were not satisfactorily addressed in this paper. Under what conditions does the lack of regularity adversely affect the performance of these methods, and do B- and BR-regular methods provide a meaningful improvement? Can integration techniques using $p(p-2)$ pairs be made as robust as those based on $p(p-1)$ pairs? Can order conditions be derived for DIRK-type methods that mitigate boundary order reduction in the nonlinear case? What stability, contractivity and monotonicity preserving characteristics or attributes really matter on practical problems? Does the number of equations being integrated influence the solvability behavior on stiff problems? Can schemes with both internally and externally embedded methods offer any useful information?
References


60. J.C. Butcher, Implicit Runge-Kutta processes, Math. Comp. 18(85) (1964) 50-64.


109. M.D. Compere, SDIRK4: Octave function to solve stiff system of first order ODEs, [http://econpapers.repec.org/software/codoctave/c031003.htm](http://econpapers.repec.org/software/codoctave/c031003.htm)


120. M. Crouzeix, Sur l’Approximation Équations Différentielles Opérationnelles Linéaires par des Méthodes de Runge-Kutta, Thesis (3rd cycle), Université de Paris 6 (1975). [In French]


219. W.H. Hundsdorfer and J. Schneid, An algebraic characterization of B-


221. W.H. Hundsdorfer and M.N. Spijker, Boundedness and strong stability of

222. W. Hundsdorfer and J. Verwer, Numerical Solution of Time-Dependent

223. S. Ilie, G. Söderlind and R.M. Corless, Adaptivity and computational com-

224. S.O. Imoni, F.O. Otunta and T.R. Ramamohan, Embedded implicit Runge-
Kutta Nyström method for solving second-order differential equations, Int. J.

225. A. Iserles, Nonlinear stability and asymptotics of O.D.E. solvers, In: Numer-

226. A. Iserles and S.P. Nørsett, On the theory of parallel Runge-Kutta methods,


228. A. Iserles, A.T. Peplow and A.M. Stuart, A unified approach to spurious solu-

229. A. Iserles, G. Ramaswami and M. Sofroniou, Runge-Kutta methods for

230. A. Iserles and A. Zanna, Preserving algebraic invariants with Runge-Kutta

231. F. Ismail, Embedded singly diagonally implicit Runge-Kutta-Nyström method
order 5(4) for the integration of special second order ODEs, Int. J. Comp. and

232. F. Ismail, Sixth order singly diagonally implicit Runge-Kutta Nyström method
with explicit first stage for solving second order ordinary differential equations,

233. F. Ismail, Diagonally implicit Runge-Kutta Nyström general method order five
for solving second order IVPs, WSEAS Trans. Math., 7(9) (2010) 550-560. [ISSN
1109-2769]


318. S.P. Nørsett and A. Wolfbrandt, Attainable order of rational approximations to the exponential function with only real poles, BIT, 17(2) (1977) 200-208.


326. B. Orel, Real pole approximations to the exponential function, BIT, 31(1) (1991) 144-159.


452. A. Wolfbrandt, A note on a recent result of rational approximations to the exponential function, BIT, 17(3) (1977) 367-368.


Appendix A

Runge-Kutta Order Conditions

Up to and including sixth-order, all relevant order conditions for Runge-Kutta methods applied to ODEs are:

\[
\begin{align*}
\tau_1^{(1)} &= \sum_{i=1}^{s} b_i - \frac{1}{\Pi} \\
\tau_1^{(3)} &= \frac{1}{3} \sum_{i=1}^{s} b_i c_i^2 - \frac{1}{3!} \\
\tau_1^{(4)} &= \frac{1}{6} \sum_{i=1}^{s} b_i c_i^3 - \frac{1}{4!} \\
\tau_3^{(4)} &= \frac{1}{3} \sum_{i,j=1}^{s} b_i a_{ij} c_j^2 - \frac{1}{4!} \\
\tau_1^{(5)} &= \frac{1}{3} \sum_{i=1}^{s} b_i c_i^4 - \frac{1}{5!} \\
\tau_3^{(5)} &= \frac{1}{3} \sum_{i,j=1}^{s} b_i a_{ij} c_j a_{jk} c_k - \frac{3}{5!} \\
\tau_5^{(5)} &= \frac{1}{3} \sum_{i,j=1}^{s} b_i a_{ij} c_j^3 - \frac{3}{5!} \\
\tau_7^{(5)} &= \sum_{i,j,k=1}^{s} b_i a_{ij} c_j a_{jk} c_k - \frac{3}{5!} \\
\tau_2^{(6)} &= \frac{1}{6} \sum_{i,j=1}^{s} b_i c_i^3 a_{ij} c_j - \frac{10}{6!} \\
\tau_4^{(6)} &= \frac{1}{6} \sum_{i,j=1}^{s} b_i c_i^2 a_{ij} c_j^2 - \frac{10}{6!} \\
\tau_6^{(6)} &= \frac{1}{6} \sum_{i,j=1}^{s} b_i c_i c_j a_{ij} c_j^3 - \frac{5}{6!} \\
\tau_8^{(6)} &= \frac{1}{6} \sum_{i,j=1}^{s} b_i c_i^2 a_{ij} a_{jk} c_k - \frac{10}{6!} \\
\tau_10^{(6)} &= \frac{1}{6} \sum_{i,j=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{3}{6!} \\
\tau_12^{(6)} &= \frac{1}{6} \sum_{i,j=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{3}{6!} \\
\tau_14^{(6)} &= \frac{1}{6} \sum_{i,j=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{3}{6!} \\
\tau_16^{(6)} &= \frac{1}{6} \sum_{i,j=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{1}{6!} \\
\tau_18^{(6)} &= \frac{1}{6} \sum_{i,j=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{1}{6!} \\
\tau_20^{(6)} &= \frac{1}{6} \sum_{i,j=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{1}{6!} \\
\tau_2^{(2)} &= \sum_{i=1}^{s} b_i c_i - \frac{1}{2!} \\
\tau_4^{(2)} &= \frac{1}{2} \sum_{i,j=1}^{s} b_i a_{ij} c_j - \frac{1}{3!} \\
\tau_2^{(3)} &= \frac{1}{2} \sum_{i,j=1}^{s} b_i c_i a_{ij} c_j - \frac{1}{4!} \\
\tau_4^{(3)} &= \frac{1}{2} \sum_{i,j=1}^{s} b_i a_{ij} a_{jk} c_k - \frac{1}{4!} \\
\tau_4^{(4)} &= \frac{1}{2} \sum_{i,j=1}^{s} b_i c_i^2 a_{ij} c_j - \frac{1}{4!} \\
\tau_4^{(5)} &= \frac{1}{2} \sum_{i,j=1}^{s} b_i a_{ij} c_j^2 - \frac{1}{4!} \\
\tau_6^{(5)} &= \sum_{i,j,k=1}^{s} b_i c_i a_{ij} a_{jk} c_k - \frac{4}{5!} \\
\tau_8^{(5)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k^2 - \frac{1}{5!} \\
\tau_10^{(5)} &= \frac{1}{120} \sum_{i=1}^{s} b_i c_i^5 - \frac{1}{6!} \\
\tau_12^{(5)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i c_i a_{ij} a_{jk} c_k - \frac{15}{6!} \\
\tau_14^{(5)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i a_{ij} c_j a_{jk} c_k - \frac{10}{6!} \\
\tau_16^{(5)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k - \frac{10}{6!} \\
\tau_18^{(5)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{10}{6!} \\
\tau_20^{(5)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{10}{6!} \\
\tau_2^{(6)} &= \frac{1}{120} \sum_{i=1}^{s} b_i c_i^5 - \frac{1}{6!} \\
\tau_4^{(6)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i c_i a_{ij} a_{jk} c_k - \frac{15}{6!} \\
\tau_6^{(6)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i a_{ij} c_j a_{jk} c_k - \frac{10}{6!} \\
\tau_8^{(6)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k - \frac{10}{6!} \\
\tau_{10}^{(6)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{10}{6!} \\
\tau_{12}^{(6)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{10}{6!} \\
\tau_{14}^{(6)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{10}{6!} \\
\tau_{16}^{(6)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{10}{6!} \\
\tau_{18}^{(6)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{10}{6!} \\
\tau_{20}^{(6)} &= \frac{1}{2} \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{10}{6!} \\
\end{align*}
\]
Appendix B

Dense Output Coefficients

ESDIRK2(1)3L[2]SA - \((p^* = 2), \tau_1^{*(1,2)} = p_3^* = 0\).

<table>
<thead>
<tr>
<th>(b_{ij}^{*})</th>
<th>(i = 1)</th>
<th>(i = 2)</th>
<th>(i = 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(j = 1)</td>
<td>8822750406821</td>
<td>8822750406821</td>
<td>-5168247530883</td>
</tr>
<tr>
<td></td>
<td>12477253282759</td>
<td>12477253282759</td>
<td>12477253282759</td>
</tr>
<tr>
<td>(j = 2)</td>
<td>-1827251437969</td>
<td>-1827251437969</td>
<td>8822750406821</td>
</tr>
<tr>
<td></td>
<td>116824+530888</td>
<td>116824+530888</td>
<td>12477253282759</td>
</tr>
</tbody>
</table>

ESDIRK3(2)4L[2]SA - \((p^* = 3), \tau_1^{*(1,2,3)} = p_4^* = 0\).

<table>
<thead>
<tr>
<th>(b_{ij}^{*})</th>
<th>(i = 1)</th>
<th>(i = 2)</th>
<th>(i = 3)</th>
<th>(i = 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(j = 1)</td>
<td>6071615849858</td>
<td>24823866123060</td>
<td>-4639021340861</td>
<td>-4782987747279</td>
</tr>
<tr>
<td></td>
<td>14664667831369</td>
<td>5641321411296</td>
<td>4798821326060</td>
<td></td>
</tr>
<tr>
<td>(j = 2)</td>
<td>-9135504192562</td>
<td>-18458657789355</td>
<td>36951656213070</td>
<td>22547150295437</td>
</tr>
<tr>
<td></td>
<td>34769930461409</td>
<td>810384546449</td>
<td>942010570113</td>
<td></td>
</tr>
<tr>
<td>(j = 3)</td>
<td>8091909798020</td>
<td>13565043189019</td>
<td>-9445293799577</td>
<td>-8621837051676</td>
</tr>
<tr>
<td></td>
<td>3414897167914</td>
<td>9402290144509</td>
<td>9402290144509</td>
<td>9402290144509</td>
</tr>
</tbody>
</table>

ESDIRK3(2I)4L[2]SA - \((p^* = 3), \tau_1^{*(1,2,3)} = p_4^* = 0\).

<table>
<thead>
<tr>
<th>(b_{ij}^{*})</th>
<th>(i = 1)</th>
<th>(i = 2)</th>
<th>(i = 3)</th>
<th>(i = 4)</th>
<th>(i = 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(j = 1)</td>
<td>18390937872020</td>
<td>18390937872020</td>
<td>4782987747279</td>
<td>4782987747279</td>
<td>8822750406821</td>
</tr>
<tr>
<td></td>
<td>16541361141131</td>
<td>16541361141131</td>
<td>5641321411296</td>
<td>5641321411296</td>
<td>12477253282759</td>
</tr>
<tr>
<td>(j = 2)</td>
<td>-3795568847221</td>
<td>48583111130060</td>
<td>-11027633676463</td>
<td>-11027633676463</td>
<td>-4639021340861</td>
</tr>
<tr>
<td></td>
<td>12094846356927</td>
<td>19629874870884</td>
<td>4798821326060</td>
<td>4798821326060</td>
<td>12477253282759</td>
</tr>
<tr>
<td>(j = 3)</td>
<td>2027866641118</td>
<td>-18802157847199</td>
<td>181992010637990</td>
<td>181992010637990</td>
<td>-9445293799577</td>
</tr>
<tr>
<td></td>
<td>13565043189019</td>
<td>3414897167914</td>
<td>9402290144509</td>
<td>9402290144509</td>
<td>9402290144509</td>
</tr>
</tbody>
</table>

ESDIRK3(2)5L[2]SA - \((p^* = 3), \tau_1^{*(1,2,3)} = \tau_3^{*4} = p_{4,5}^* = 0\).

<table>
<thead>
<tr>
<th>(b_{ij}^{*})</th>
<th>(i = 1)</th>
<th>(i = 2)</th>
<th>(i = 3)</th>
<th>(i = 4)</th>
<th>(i = 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(j = 1)</td>
<td>18390937872020</td>
<td>18390937872020</td>
<td>4587276387100</td>
<td>-169812</td>
<td>-6561</td>
</tr>
<tr>
<td></td>
<td>16541361141131</td>
<td>16541361141131</td>
<td>11281280648079</td>
<td>40789</td>
<td>5824</td>
</tr>
<tr>
<td>(j = 2)</td>
<td>-25205650154962</td>
<td>-25205650154962</td>
<td>-83784512863764</td>
<td>493900</td>
<td>14580</td>
</tr>
<tr>
<td></td>
<td>1994999269151</td>
<td>1994999269151</td>
<td>-7610486703441</td>
<td>40789</td>
<td>5824</td>
</tr>
<tr>
<td>(j = 3)</td>
<td>4643928352124</td>
<td>4643928352124</td>
<td>8758481818205</td>
<td>-8139200</td>
<td>-7200</td>
</tr>
<tr>
<td></td>
<td>5273764340929</td>
<td>5273764340929</td>
<td>1279801860862</td>
<td>1101303</td>
<td>5824</td>
</tr>
</tbody>
</table>

152
ESDIRK3(2I)5L[2]SA - \( (p^* = 3), \tau_1^{(1,2,3)} = \tau_3^{(4)} = p_{4,5}^* = 0. \)

<table>
<thead>
<tr>
<th>( b_{ij}^* )</th>
<th>( i = 1 )</th>
<th>( i = 2 )</th>
<th>( i = 3 )</th>
<th>( i = 4 )</th>
<th>( i = 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( j = 1 )</td>
<td>545</td>
<td>729</td>
<td>-455</td>
<td>1528308</td>
<td>-6561</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>938147</td>
<td>5827</td>
</tr>
<tr>
<td>( j = 2 )</td>
<td>-12205</td>
<td>-3040</td>
<td>2353</td>
<td>-4445100</td>
<td>14580</td>
</tr>
<tr>
<td></td>
<td>11152</td>
<td>10143</td>
<td>9016</td>
<td>938147</td>
<td>5827</td>
</tr>
<tr>
<td>( j = 3 )</td>
<td>1823</td>
<td>4000</td>
<td>-2325</td>
<td>8139200</td>
<td>-7280</td>
</tr>
<tr>
<td></td>
<td>2347</td>
<td>30129</td>
<td>-1127</td>
<td>2981444</td>
<td>5827</td>
</tr>
</tbody>
</table>

ESDIRK4(3)5L[2]SA - \( (p^* = 3), \tau_1^{(1,2,3)} = p_{4,5}^* = 0. \)

<table>
<thead>
<tr>
<th>( b_{ij}^* )</th>
<th>( i = 1 )</th>
<th>( i = 2 )</th>
<th>( i = 3 )</th>
<th>( i = 4 )</th>
<th>( i = 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( j = 1 )</td>
<td>2448612054971</td>
<td>-1306452724769</td>
<td>47239002818071</td>
<td>-60230645086332</td>
<td>2454515595281</td>
</tr>
<tr>
<td></td>
<td>7442989176026</td>
<td>-7607699585395</td>
<td>9565057903650</td>
<td>-9640595487472</td>
<td>9520609176942</td>
</tr>
<tr>
<td>( j = 2 )</td>
<td>-5143529007961</td>
<td>6553122538256</td>
<td>-863580884391876</td>
<td>13418521236339</td>
<td>-52206835153223</td>
</tr>
<tr>
<td></td>
<td>7772935284608</td>
<td>77060575415020</td>
<td>23121359989535</td>
<td>41896654175741</td>
<td>-11015268282971</td>
</tr>
<tr>
<td>( j = 3 )</td>
<td>3995567809954</td>
<td>1415023963286</td>
<td>429256212956</td>
<td>-6804693451504</td>
<td>5428318762715</td>
</tr>
<tr>
<td></td>
<td>9592491338095</td>
<td>1156647375329</td>
<td>2996140495957</td>
<td>8676318289197</td>
<td>5589821095638</td>
</tr>
</tbody>
</table>

ESDIRK4(3I)5L[2]SA - \( (p^* = 3), \tau_1^{(1,2,3)} = p_{4,5}^* = 0. \)

<table>
<thead>
<tr>
<th>( b_{ij}^* )</th>
<th>( i = 1 )</th>
<th>( i = 2 )</th>
<th>( i = 3 )</th>
<th>( i = 4 )</th>
<th>( i = 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( j = 1 )</td>
<td>1094126747374</td>
<td>12387998990496</td>
<td>50966886923506</td>
<td>-137558177997005</td>
<td>2454515595281</td>
</tr>
<tr>
<td></td>
<td>5032806710219</td>
<td>19890238794661</td>
<td>11897153581169</td>
<td>9441337973237</td>
<td>4920609176942</td>
</tr>
<tr>
<td>( j = 2 )</td>
<td>-3113653767609</td>
<td>-29688901167957</td>
<td>-3161430744844</td>
<td>145504386170428</td>
<td>-52206835153223</td>
</tr>
<tr>
<td></td>
<td>83670653342</td>
<td>4114912884820</td>
<td>10804162464371</td>
<td>9440597972891</td>
<td>-11015268282971</td>
</tr>
<tr>
<td>( j = 3 )</td>
<td>3145703962191</td>
<td>40752634139353</td>
<td>1286958559169</td>
<td>-24862816003436</td>
<td>5428318762715</td>
</tr>
<tr>
<td></td>
<td>11339919304645</td>
<td>19417583977863</td>
<td>1830973958079</td>
<td>7858184989993</td>
<td>5589821095633</td>
</tr>
</tbody>
</table>

ESDIRK4(3)6L[2]SA - \( (p^* = 4), \tau_1^{(1,2,3,4)} = \tau_3^{(4)} = p_{6}^* = 0. \)

<table>
<thead>
<tr>
<th>( b_{ij}^* )</th>
<th>( i = 1 )</th>
<th>( i = 2 )</th>
<th>( i = 3 )</th>
<th>( i = 4 )</th>
<th>( i = 5 )</th>
<th>( i = 6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( j = 1 )</td>
<td>11963910384665</td>
<td>1196391038465</td>
<td>-28603264624</td>
<td>-3524425447183</td>
<td>-17173522440186</td>
<td>-10195024317061</td>
</tr>
<tr>
<td></td>
<td>12483345400363</td>
<td>12483345400363</td>
<td>1970169029871</td>
<td>268317707070205</td>
<td>-10195024317061</td>
<td>1300506014233</td>
</tr>
<tr>
<td>( j = 2 )</td>
<td>-69996760340788</td>
<td>-69996760340788</td>
<td>102610171901103</td>
<td>749576296707620</td>
<td>113851399235623</td>
<td>-8422932543950</td>
</tr>
<tr>
<td></td>
<td>185265959514355</td>
<td>185265959514355</td>
<td>26266595787953</td>
<td>12279805097313</td>
<td>9882566320290</td>
<td>60777405593999</td>
</tr>
<tr>
<td>( j = 3 )</td>
<td>324736354929419</td>
<td>324736354929419</td>
<td>-3886637253484</td>
<td>-2670571722886</td>
<td>-121105832411520</td>
<td>112002854703824</td>
</tr>
<tr>
<td></td>
<td>7031701515065</td>
<td>7031701515065</td>
<td>624935826105</td>
<td>426567113133</td>
<td>-6658412667527</td>
<td>51424476870755</td>
</tr>
<tr>
<td>( j = 4 )</td>
<td>-146682528638283</td>
<td>-146682528638283</td>
<td>21103558580591</td>
<td>30155591475533</td>
<td>1199583757102688</td>
<td>-63862213973224</td>
</tr>
<tr>
<td></td>
<td>5053461017165</td>
<td>5053461017165</td>
<td>57177948659092</td>
<td>9296909409044</td>
<td>14392846999991</td>
<td>-6939839420717</td>
</tr>
</tbody>
</table>

153
ESDIRK4(3)6L[2]SA_C3 - $(p^* = 4), \tau_1^{*(1,2,3,4)} = \tau_3^{*(4)} = p_6^* = 0$.

<table>
<thead>
<tr>
<th>$b_{1,i}$</th>
<th>$i = 1$</th>
<th>$i = 2$</th>
<th>$i = 3$</th>
<th>$i = 4$</th>
<th>$i = 5$</th>
<th>$i = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>j = 1</td>
<td>8569188290151</td>
<td>10994399572142</td>
<td>0</td>
<td>10318667322223</td>
<td>-11939140141175</td>
<td>4891659247449</td>
</tr>
<tr>
<td>j = 2</td>
<td>-2579945991926</td>
<td>-1438397299599</td>
<td>0</td>
<td>-16227090938990</td>
<td>97074040415774</td>
<td>-22781540631859</td>
</tr>
<tr>
<td>j = 3</td>
<td>7530213627617</td>
<td>15959880102338</td>
<td>0</td>
<td>29191903871692</td>
<td>-56594149574717</td>
<td>65637117360727</td>
</tr>
<tr>
<td>j = 4</td>
<td>-2524520504241</td>
<td>7216797214699</td>
<td>0</td>
<td>-4093470269096</td>
<td>12693912957420</td>
<td>-22692698394349</td>
</tr>
</tbody>
</table>

QESDIRK4(3)6L[2]SA - $(p^* = 4), \tau_1^{*(1,2,3,4)} = \tau_3^{*(4)} = p_6^* = 0$.

<table>
<thead>
<tr>
<th>$b_{1,i}$</th>
<th>$i = 1$</th>
<th>$i = 2$</th>
<th>$i = 3$</th>
<th>$i = 4$</th>
<th>$i = 5$</th>
<th>$i = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>j = 1</td>
<td>30919413421679</td>
<td>0</td>
<td>1395097886185</td>
<td>1753631998906</td>
<td>1210068607139</td>
<td>1990115420181</td>
</tr>
<tr>
<td>j = 2</td>
<td>-10083671929737</td>
<td>0</td>
<td>-1707368234419</td>
<td>5694984320416</td>
<td>11639130678959</td>
<td>811609611777</td>
</tr>
<tr>
<td>j = 3</td>
<td>1958791018141</td>
<td>0</td>
<td>2660148292392</td>
<td>2134307940221</td>
<td>-16262995508332</td>
<td>4226360292941</td>
</tr>
<tr>
<td>j = 4</td>
<td>-1073516272895</td>
<td>0</td>
<td>-380444841507</td>
<td>1580764210107</td>
<td>15454682310995</td>
<td>-36615454820305</td>
</tr>
</tbody>
</table>

ESDIRK4(3I)6L[2]SA - $(p^* = 3), \tau_1^{*(1,2,3)} = \tau_3^{*(4)} = p_{5,6}^* = 0$.

<table>
<thead>
<tr>
<th>$b_{1,i}$</th>
<th>$i = 1$</th>
<th>$i = 2$</th>
<th>$i = 3$</th>
<th>$i = 4$</th>
<th>$i = 5$</th>
<th>$i = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>j = 1</td>
<td>-15963545394223</td>
<td>0</td>
<td>-15563545394223</td>
<td>18260690586330</td>
<td>-5353066601167</td>
<td>317</td>
</tr>
<tr>
<td>j = 2</td>
<td>88669663770139</td>
<td>0</td>
<td>-14849207136675</td>
<td>2197675841675</td>
<td>-102</td>
<td>12</td>
</tr>
<tr>
<td>j = 3</td>
<td>-12758519994857</td>
<td>0</td>
<td>77012120412031</td>
<td>8828178732128</td>
<td>296</td>
<td>-32</td>
</tr>
<tr>
<td>j = 4</td>
<td>51444619706198</td>
<td>0</td>
<td>7110803304477</td>
<td>9934911810022</td>
<td>21</td>
<td>3</td>
</tr>
</tbody>
</table>

ESDIRK5(3)6L[2]SA - $(p^* = 4), \tau_1^{*(1,2,3,4)} = \tau_3^{*(4)} = p_6^* = 0$.

<table>
<thead>
<tr>
<th>$b_{1,i}$</th>
<th>$i = 1$</th>
<th>$i = 2$</th>
<th>$i = 3$</th>
<th>$i = 4$</th>
<th>$i = 5$</th>
<th>$i = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>j = 1</td>
<td>2323665026058</td>
<td>0</td>
<td>445253991667</td>
<td>736506820574</td>
<td>18177212952131</td>
<td>-33541179251535</td>
</tr>
<tr>
<td>j = 2</td>
<td>-24290617855491</td>
<td>0</td>
<td>2410462087771</td>
<td>539085104555</td>
<td>-110401299028930</td>
<td>215126192174603</td>
</tr>
<tr>
<td>j = 3</td>
<td>24919615044541</td>
<td>0</td>
<td>24748696994846</td>
<td>15551018103727</td>
<td>126745652682189</td>
<td>-13577131446793</td>
</tr>
<tr>
<td>j = 4</td>
<td>-314084723539471</td>
<td>0</td>
<td>32967269088779</td>
<td>3617504046863</td>
<td>-27496122456930</td>
<td>32925565290571</td>
</tr>
<tr>
<td>j = 5</td>
<td>38813534033374</td>
<td>0</td>
<td>7127362680564</td>
<td>6873295955553</td>
<td>8436293721225</td>
<td>7468291342706</td>
</tr>
</tbody>
</table>

154
\[
\text{ESDIRK5(4)7L[2]SA - } (p^* = 4), \tau_1^{(1,2,3,4)} = \tau_3^{(4)} = p_6^{*} = 0.
\]

<table>
<thead>
<tr>
<th>(b_{ij}^*)</th>
<th>(i = 1)</th>
<th>(i = 2)</th>
<th>(i = 3)</th>
<th>(i = 4)</th>
<th>(i = 5)</th>
<th>(i = 6)</th>
<th>(i = 7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(j = 1)</td>
<td>0.37276707487715</td>
<td>0.37276707487715</td>
<td>0.9961707013941</td>
<td>0.9961707013941</td>
<td>2.6548122345106</td>
<td>1.251403284075</td>
<td>0.9459581935351</td>
</tr>
<tr>
<td>(j = 2)</td>
<td>0.6559662850736</td>
<td>0.6559662850736</td>
<td>0.9243059376682</td>
<td>2.31063818727</td>
<td>-5.2742949967696</td>
<td>0.7169971713011</td>
<td>0.1359580725169</td>
</tr>
<tr>
<td>(j = 3)</td>
<td>-4.9960714819066</td>
<td>-4.9960714819066</td>
<td>0.241311615437</td>
<td>0.3540629594230</td>
<td>0.2921414557025</td>
<td>-0.7791626760391</td>
<td></td>
</tr>
<tr>
<td>(j = 4)</td>
<td>0.294487841061</td>
<td>0.294487841061</td>
<td>-3.350221127663</td>
<td>1.5182083246731</td>
<td>0.9110172521395</td>
<td>0.8458206974485</td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{ESDIRK5(4I)8L[2]SA - } (p^* = 4), \tau_1^{(1,2,3,4)} = \tau_3^{(4)} = \tau_9^{(5)} = p_8^{*} = 0.
\]

<table>
<thead>
<tr>
<th>(b_{ij}^*)</th>
<th>(i = 1)</th>
<th>(i = 2)</th>
<th>(i = 3)</th>
<th>(i = 4)</th>
<th>(i = 5)</th>
<th>(i = 6)</th>
<th>(i = 7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(j = 1)</td>
<td>1.1765908361881</td>
<td>1.24852821439</td>
<td>-8.4842141872035</td>
<td>1.6069095514627</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(j = 2)</td>
<td>0.1048110876174</td>
<td>0.7857941723213</td>
<td>0.843747849189451</td>
<td>1.8664745120152</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(j = 3)</td>
<td>-1.09190633585</td>
<td>-0.9087567427743</td>
<td>0.0551218618292</td>
<td>0.6607739408042</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(j = 4)</td>
<td>8.7923027904111</td>
<td>1.5020606926489</td>
<td>1.719925309182460</td>
<td>-0.672113599330516</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(j = 5)</td>
<td>6.3516441177474</td>
<td>-3.0233403050287</td>
<td>0.5074818172957</td>
<td>0.9985071187814</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{ESDIRK6(4)7L[2]SA - } (p^* = 4), \tau_1^{(1,2,3,4)} = \tau_3^{(4)} = p_6^{*} = 0.
\]

<table>
<thead>
<tr>
<th>(b_{ij}^*)</th>
<th>(i = 1)</th>
<th>(i = 2)</th>
<th>(i = 3)</th>
<th>(i = 4)</th>
<th>(i = 5)</th>
<th>(i = 6)</th>
<th>(i = 7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(j = 1)</td>
<td>0.9604644101905</td>
<td>-1.2477204502231</td>
<td>0.2078090578231</td>
<td>-1.4991040467097</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(j = 2)</td>
<td>-0.3152857710737</td>
<td>0.3617084135296</td>
<td>-0.5247224054092</td>
<td>0.9293118148770</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(j = 3)</td>
<td>-0.9099671799825</td>
<td>-0.7999658479874</td>
<td>0.31917462687507</td>
<td>-0.7663737359291</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(j = 4)</td>
<td>-0.1247846817527</td>
<td>0.14630060929664</td>
<td>-0.1170448301144</td>
<td>0.45218709836216</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(j = 5)</td>
<td>0.5081265544977</td>
<td>-0.0650736226368</td>
<td>0.3725041773767</td>
<td>-0.619341826691</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{ESDIRK6(4I)8L[2]SA - } (p^* = 4), \tau_1^{(1,2,3,4)} = \tau_3^{(4)} = \tau_9^{(5)} = p_8^{*} = 0.
\]
### Appendix C

**Method Properties**

Various method properties are presented below for the 15 methods presented in this document as well as those for both SDIRK4 methods. Several letters are used in the table are shorthand for explicit, E, implicit, I, revised, R, yes, Y and no, N. Other shorthand includes S.A. for stiffly-accurate and A-St. for A-stable. Other terms are given in the body of the document.

| Scheme                  | (s, r) | A(p+1) | B(p+1) | C(p+1) | D | E | A-St. | R(∞) | S.A. | λ_{min} | | |  | η_{max} |
|-------------------------|--------|--------|--------|--------|---|---|-------|------|-----|---------|---|---|--------|
| ESDIRK2(1)IL(2)SA      | (3, 2) | 0.0519 | 0.0251 | 3.105  | 1 | 2.276 | Y     | 0.2929 | 0.2929 | 0.1279 | 0.1751 | 0.2929 |
| ESDIRK3(2)IL(2)SA      | (3, 2) | 0.03653 | 0.02552 | 2.967  | 1.271 | 1.435 | Y     | 0.2179 | -0.5953 | 0.1909 | 1.538 | 0.4339 |
| ESDIRK3(2)IL(2)SA      | (3, 2) | 0.0907 | 0.01223 | 1.192 | 1.923 | E | Y     | 0.2179 | -0.9671 | 1.395 | 1.87 | 0.3459 |
| ESDIRK3(2)IL(2)SA      | (3, 2) | 0.003199 | 0.002437 | 1.208 | 0.3296 | Y     | 0 | -0.3409 | 0.05529 | 1.148 | 1.00064 |
| ESDIRK3(2)IL(2)SA      | (3, 2) | 0.01496 | 0.002436 | 1.184 | 1.163 | E | Y     | 0 | -0.2859 | 0.2970 | 0.2250 |
| ESDIRK3(2)IL(2)SA      | (3, 2) | 0.0857 | 0.01214 | 1.576 | 1.210 | Y | Y | 0.1909 | 0.3987 | 0.3281 | 1.209 | 1.146 |
| ESDIRK4(3)IL(2)SA      | (3, 2) | 0.04966 | 0.001603 | 1.160 | 1.146 | Y | Y | 0 | -1.14 | 1.180 | 0.5728 |
| ESDIRK4(3)IL(2)SA      | (3, 2) | 0.002549 | 0.004511 | 2.147 | 7.930 | Y | Y | 0 | -0.7623 | 0.121 | 0.4935 | 2.64 | 1.146 |
| SDIRK4(1)              | (3, 1) | 0.002564 | 0.001659 | 2.555 | 9.705 | 1.192 | Y | 0 | -0.500 | -0.360 | 8.783 | 11.2 | 1.0000 |
| (Original and Revised b) | (3, 1) | 0.007642 | 0.006417 | 0.926 | 0.926 | Y | Y | 0 | -0.1083 | 0.1978 | 0.2847 | 1.04 | 0.9667 |
| SDIRK4(2)              | (3, 1) | 0.002479 | 0.005382 | 1.150 | 1.237 | Y | Y | 0 | -0.2069 | 0.4008 | 0.2000 |
| (Original and Revised b) | (3, 1) | 0.003136 | 0.001647 | 0.926 | 0.926 | Y | Y | 0 | -0.1125 | 0.2869 | 0.3115 | 1 | 0.9999 |
| ESDIRK3(3)IL(3)SA      | (3, 2) | 0.008183 | 0.001079 | 1.270 | 2.17 | Y | Y | 0 | -0.1971 | 0.2875 | 0.2500 |
| ESDIRK3(3)IL(3)SA      | (3, 2) | 0.008286 | 0.005420 | 1.998 | 1.515 | Y | Y | 0 | -0.1330 | 8.189 | 9.109 | 1.044 |
| ESDIRK3(3)IL(3)SA      | (3, 2) | 0.002254 | 0.003364 | 1.159 | 0.823 | Y | Y | 0 | -0.2426 | 0.4051 | 0.2500 |
| ESDIRK4(3)IL(3)SA      | (3, 2) | 0.008364 | 0.001079 | 1.270 | 2.17 | Y | Y | 0 | -0.1071 | 0.3132 | 0.2864 | 1 |
| ESDIRK4(3)IL(3)SA      | (3, 2) | 0.004974 | 0.001367 | 1.150 | 1.237 | Y | Y | 0 | -0.1971 | 0.2875 | 0.2500 |
| ESDIRK4(3)IL(3)SA      | (3, 2) | 0.008183 | 0.001079 | 1.270 | 2.17 | Y | Y | 0 | -0.1971 | 0.2875 | 0.2500 |
| ESDIRK4(3)IL(3)SA      | (3, 2) | 0.008286 | 0.005420 | 1.998 | 1.515 | Y | Y | 0 | -0.1330 | 8.189 | 9.109 | 1.044 |
| ESDIRK4(3)IL(3)SA      | (3, 2) | 0.002254 | 0.003364 | 1.159 | 0.823 | Y | Y | 0 | -0.2426 | 0.4051 | 0.2500 |

156
A review of diagonally implicit Runge-Kutta (DIRK) methods applied to first-order ordinary differential equations (ODEs) is undertaken. The goal of this review is to summarize the characteristics, assess the potential, and then design several nearly optimal, general purpose, DIRK-type methods. Over 20 important aspects of DIRK-type methods are reviewed. A design study is then conducted on DIRK-type methods having from two to seven implicit stages. From this, 15 schemes are selected for general purpose application. Testing of the 15 chosen methods is done on three singular perturbation problems. Based on the review of method characteristics, these methods focus on having a stage order of two, stiff accuracy, L-stability, high quality embedded and dense-output methods, small magnitudes of the algebraic stability matrix eigenvalues, small values of $a_{ii}$, and small or vanishing values of the internal stability function for large eigenvalues of the Jacobian. Among the 15 new methods, ESDIRK4(3)6L[2]SA is recommended as a good default method for solving stiff problems at moderate error tolerances.