Available at WWW.MATHEMATICSWEB.ORG POWERED BY SCIENCE @ DIRECT®

ELSEVIER

Applied Numerical Mathematics 44 (2003) 139–181

APPLIED NUMERICAL MATHEMATICS

www.elsevier.com/locate/apnum

Additive Runge–Kutta schemes for convection–diffusion–reaction equations

Christopher A. Kennedy [a,*], Mark H. Carpenter [b]

[a] *Combustion Research Facility, Sandia National Laboratories, Livermore, CA 94551-0969, USA*
[b] *Aeronautics and Aeroacoustic Methods Branch, NASA Langley Research Center, Hampton, VA 23681-0001, USA*

Abstract

Additive Runge–Kutta (ARK) methods are investigated for application to the spatially discretized one-dimensional convection–diffusion–reaction (CDR) equations. Accuracy, stability, conservation, and dense-output are first considered for the general case when N different Runge–Kutta methods are grouped into a single composite method. Then, implicit–explicit, $(N = 2)$, additive Runge–Kutta (ARK$_2$) methods from third- to fifth-order are presented that allow for integration of stiff terms by an L-stable, stiffly-accurate explicit, singly diagonally implicit Runge–Kutta (ESDIRK) method while the nonstiff terms are integrated with a traditional explicit Runge–Kutta method (ERK). Coupling error terms of the partitioned method are of equal order to those of the elemental methods. Derived ARK$_2$ methods have vanishing stability functions for very large values of the stiff scaled eigenvalue, $z^{[I]} \to -\infty$, and retain high stability efficiency in the absence of stiffness, $z^{[I]} \to 0$. Extrapolation-type stage-value predictors are provided based on dense-output formulae. Optimized methods minimize both leading order ARK$_2$ error terms and Butcher coefficient magnitudes as well as maximize conservation properties. Numerical tests of the new schemes on a CDR problem show negligible stiffness leakage and near classical order convergence rates. However, tests on three simple singular-perturbation problems reveal generally predictable order reduction. Error control is best managed with a PID-controller. While results for the fifth-order method are disappointing, both the new third- and fourth-order methods are at least as efficient as existing ARK$_2$ methods.

1. Introduction

It is oftentimes useful to consider the compressible Navier–Stokes equations (NSE) as evolution equations with several driving forces, each having somewhat different characteristics. Typically, one distinguishes among terms such as convection, diffusion, and reaction. As such, one often considers the

[*] Corresponding author.
E-mail address: cakenne@ca.sandia.gov (C.A. Kennedy).

0168-9274/02/$22.00 © 2002 IMACS. Published by Elsevier Science B.V. All rights reserved.
PII: S0168-9274(02)00138-1

more tractable convection–diffusion–reaction (CDR) equations as a prologue to the full compressible NSE [34]. In the search for ever more efficient integrators, it is intuitively appealing to seek individual integration methods that are ideally suited for specific parts of the governing equations. The individual methods are then rolled into a single composite method that, ideally, would be more efficient than any individual method applied to the full computation. To accomplish this, one may consider partitioned methods. Schemes constructed to take advantage of termwise partitioning of the CDR for integration purposes may be called additive methods [48]. Partitioning of the discretized form of the equations may also be performed on an equationwise or pointwise basis [24]. There are many different partitioning strategies [71]. Runge–Kutta methods, with their extensive theoretical foundation, allow for straightforward design and construction of stable, high-order, partitioned methods composed of arbitrary numbers of elemental Runge–Kutta schemes. In addition, they also allow for the direct control of partitioning (splitting or coupling) errors [47]. Direct numerical simulation (DNS) and large-eddy simulation (LES) of fluid phenomena, with their relatively strict error tolerances, are prime candidates for such methods. The need for these strategies is by no means limited to Navier–Stokes applications [7,19,54,68].

From a termwise point of view, a linearization of one-dimensional CDR equations can provide insight into the distinguishing characteristics of each term. Upon method-of-lines discretization using high-order, finite-difference techniques, the CDR equations may be written as a system of ordinary differential equations (ODEs) and analyzed with

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \lambda^C U + \lambda^D U + \lambda^R U, \tag{1}$$

where

$$z^C = \lambda^C(\Delta t), \qquad z^D = \lambda^D(\Delta t), \qquad z^R = \lambda^R(\Delta t),$$

and (Δt) is the time step. The discretized convection term contributes scaled eigenvalues, z^C, that are predominately imaginary while the diffusion terms have predominately real scaled eigenvalues, z^D. Reaction rate eigenvalues, λ^R, are mostly real and may give rise to relatively large scaled eigenvalues, z^R. Based on this knowledge, one might seek to construct a new method based on two separate methods: one optimized to smaller eigenvalues of the convection and diffusion terms and one that is capable of dealing with very large reaction-rate eigenvalues. It should be remarked that because of the high sound speeds introduced by the compressible equations, $|z^C|$ is generally larger than $|z^D|$, even in the DNS of a hypersonic boundary layer resolved down to $y^+ = 1$. Discretized incompressible flows, governed by index-2 differential algebraic equations, generally have $|z^D| > |z^C|$.

If the stability domain of the integrator contains all values of z^C, z^D, and z^R, then stable integration can be done. For accuracy purposes, the integration must proceed no faster than the fastest relevant physical processes contained within the governing equations. A situation may arise, however, where stable integration of the discretized governing equations can only proceed at a time scale substantially faster than any physically relevant time scale of the continuum-based compressible equations. This may render a numerical method unacceptably inefficient. It may occur in regions of intense grid clustering or while using stiff chemistry, but may be caused by other issues like interface boundary conditions within a multidomain formulation. There are two possible strategies to obtaining a solution at a reasonable cost when this occurs: change the governing equations and hence their characteristic time scales or change the numerical method. We choose the latter.

Partitioned Runge–Kutta methods may be designed to allow for the partitioning of equations by term, gridpoint, or equation. Implicit–explicit (IMEX) partitioned methods developed to date for first order

ODEs have usually considered a partitioning based on terms. They have combined explicit Runge–Kutta (ERK) schemes with variations on either the diagonally implicit Runge–Kutta (DIRK) [5,10,16,18,21, 31,51,61,75–78] or Rosenbrock family of methods [12–14,39,52,57,61,75–78]. In this paper, methods are derived using stiffly-accurate, explicit, singly diagonally implicit Runge–Kutta (ESDIRK) schemes for their stability properties and higher stage-order. Although the word explicit appears in the name of ESDIRKs, they are implicit methods. Of the partitioned IMEX Runge–Kutta methods currently available in the literature, some exhibit lower-order coupling errors, coupling stability problems, no error control, and poor ERK or DIRK/Rosenbrock stability properties. The new schemes endeavor to address all of these shortcomings without falling prey to new ones.

A multistep [4,17,35] approach to IMEX schemes is also possible. Higher-order SBDF methods given by Ascher et al. [4], based on BDF methods for the implicit portion, are of the form

$$U^{(n+1)} = -\sum_{j=0}^{k-1}\alpha_j U^{(n-j)} + (\Delta t)\beta_k \sum_{j=0}^{k-1}\left(\frac{(-1)^j k!}{(k-j-1)!(j+1)!}\right)F^{(n-j)}_{\text{explicit}} + (\Delta t)\beta_k F^{(n+1)}_{\text{implicit}},$$

where α_j and β_k are the values for the k-step, order-k, BDF methods [46]. These methods may be quite effective for equations whose stiff terms give rise to predominately real eigenvalues. At third-order and above the BDF methods are only $L(\alpha)$-stable; they lack A-stability but still have a vanishing stability function for extremely stiff eigenvalues. Strongly transient events like chemical ignition may be less appropriate for SBDF methods because they lack stepsize adjustment.

The goal of this paper is to provide complete methods for solving CDR equations using additive Runge–Kutta methods. By this, it is meant that beyond having good accuracy and stability properties, there are high quality embedded methods, error controllers, dense output, stage-value predictors, and implementation guidance. This is done in a two-fold manner. First, a general overview is given of the coupling of N different Runge–Kutta methods for first-order ODEs whose right-hand side is the summation of N terms: N-additive Runge–Kutta methods (ARK$_N$). Second, matters are then specialized to the case of $N = 2$: additive Runge–Kutta (ARK$_2$) methods using an implicit method that allows accurate and stable integration of the stiff terms while either integrating the nonstiff terms at the linear stability of the explicit method or integrating the entire method at some chosen error tolerance.

We do not consider the topics of storage reduction, contractivity, dispersion and dissipation, regularity, or boundary error. Sections 1 and 2 will provide an introduction to ARK$_N$ methods. Specialization of additive methods to ERK–ESDIRK combinations will be reviewed in Section 3. Third-, fourth-, and fifth-order schemes will be considered in Sections 4, 5, and 6. Testing of the ARK$_2$ schemes on a one-dimensional, convection–diffusion–reaction test problem and three, two-equation, singular-perturbation problems is discussed in Section 7. Merits of the additive schemes are discussed in Section 8 and comparisons are made with existing Runge–Kutta and multistep IMEX methods. In Section 9, conclusions are drawn as to utility of the various schemes. Appropriate appendices are also included. All new methods presented in this study were constructed completely using Mathematica [72,73]. Coefficients of the new schemes are provided to at least 25 digits of accuracy. A more detailed version of the current paper, referred to as the extended paper, is available [42].

2. N-additive Runge–Kutta methods

2.1. General

Following Araújo et al. [3], ARK$_N$ methods are used to solve equations of the form

$$\frac{\mathrm{d}U}{\mathrm{d}t} = F(U) = \sum_{\nu=1}^{N} F^{[\nu]}(U), \tag{2}$$

where $F(U)$ has been additively composed of N terms. They are applied as

$$U^{(i)} = U^{(n)} + (\Delta t)\sum_{\nu=1}^{N}\sum_{j=1}^{s} a_{ij}^{[\nu]} F^{[\nu]}\big(U^{(j)}\big), \tag{3}$$

$$\begin{aligned} U^{(n+1)} &= U^{(n)} + (\Delta t)\sum_{\nu=1}^{N}\sum_{i=1}^{s} b_{i}^{[\nu]} F^{[\nu]}\big(U^{(i)}\big), \\ \widehat{U}^{(n+1)} &= U^{(n)} + (\Delta t)\sum_{\nu=1}^{N}\sum_{i=1}^{s} \hat{b}_{i}^{[\nu]} F^{[\nu]}\big(U^{(i)}\big), \end{aligned} \tag{4}$$

where each of the N terms are integrated by its own s-stage Runge–Kutta method. Also, $U^{(n)} \simeq U(t^{(n)})$, $U^{(i)} \simeq U(t^{(n)} + c_i\Delta t)$ is the value of the U-vector on the ith-stage, and $U^{(n+1)} \simeq U(t^{(n)} + \Delta t)$. Both $U^{(n)}$ and $U^{(n+1)}$ are of classical order q. The U-vector associated with the embedded scheme, $\widehat{U}^{(n+1)}$, is of order p. Each of the respective Butcher coefficients $a_{ij}^{[\nu]}$, $b_i^{[\nu]}$, $\hat{b}_i^{[\nu]}$, and $c_i^{[\nu]}$, $\nu = 1, 2, \ldots, N$ are constrained, at a minimum, by certain order of accuracy and stability considerations.

2.1.1. Order conditions

Order-of-accuracy conditions for ARK$_N$ methods may be derived via N-trees [3]. These N-trees resemble the traditional Butcher 1-trees [8,26,27] but each node may be any one of N varieties or colors. Expressions for the equations of condition associated with the qth-order N-trees are of the form

$$\boldsymbol{\tau}_{k[n]}^{(q)} = \frac{1}{\sigma}\Phi_{k[n]}^{(q)} - \frac{\alpha}{q!} = \frac{1}{\sigma}\left(\Phi_{k[n]}^{(q)} - \frac{1}{\gamma}\right), \quad \text{where } \Phi_{k[n]}^{(q)} = \sum_{i}^{s} b_i \Phi_{i,k[n]}^{(q)}, \tag{5}$$

where $\Phi_{k[n]}^{(q)}$ and $\Phi_{i,k[n]}^{(q)}$ are scalar sums of Butcher coefficient products and $1 \leqslant n \leqslant N^q$ is used to distinguish between the many possible color variations of the kth 1-tree of order q. Both α and σ are color dependent; i.e., for any given 1-tree, the many corresponding N-trees may have different values of α and σ depending on the details of the node colorings. Tree density, γ, is color independent and consequently so is the product of α and σ, $\sigma\alpha = q!/\gamma$. Order conditions $\boldsymbol{\tau}_2^{(3)}$, $\boldsymbol{\tau}_{2,4}^{(4)}$, and $\boldsymbol{\tau}_{6,7,9}^{(5)}$ given in Appendix A, never exhibit color dependence. When an equation of condition, $\boldsymbol{\tau}_{k[n]}^{(q)}$, is made to vanish, color dependence is immaterial because $\Phi_{k[n]}^{(q)} = 1/\gamma$. For equations of condition that are not made to vanish, color dependence must be taken into consideration to accurately assess the leading order error terms. Order conditions for partitioned Runge–Kutta methods have also been derived by Jackiewicz and Vermiglio [37] following an approach of Albrecht.

2.1.2. Coupling conditions

Aside from satisfying the order conditions specific to each of the elemental methods of the ARK$_N$, one must also satisfy various coupling conditions. One may write the total number of order conditions for a general ARK$_N$ associated with each particular root node coloring, $\alpha_i^{[N]}$, using the expression [24,26,62]

$$\sum_{i=1}^{\infty} \alpha_i^{[N]} x^{(i-1)} = \prod_{i=1}^{\infty} \left(1 - x^i\right)^{-N\alpha_i^{[N]}}. \tag{6}$$

At order $i = \{1, 2, 3, 4, 5, 6, \ldots\}$, a general ARK$_N$ method has a total of $N\alpha_i^{[N]}$ order conditions, where $\alpha_i^{[1]} = \{1, 1, 2, 4, 9, 20, \ldots\}$, $\alpha_i^{[2]} = \{1, 2, 7, 26, 107, 458, \ldots\}$, $\alpha_i^{[3]} = \{1, 3, 15, 82, 495, 3144, \ldots\}$, $\alpha_i^{[4]} = \{1, 4, 26, 188, 1499, 12628, \ldots\}$, and $\alpha_i^{[5]} = \{1, 5, 40, 360, 3570, 37476, \ldots\}$. Some of these, $N\alpha_i^{[1]}$, are order conditions of the elemental methods which compose the ARK$_N$. This implies that $N(\alpha_i^{[N]} - \alpha_i^{[1]})$ of the order conditions are composed of portions of different elemental methods. These are coupling conditions. As both order of accuracy and N increase, their numbers grow explosively. Table 1 shows their numbers for orders up to five and for each 1-tree listed in Appendix A with which the coupling conditions are associated and also depending on whether the simplifying assumptions $b_i^{[\nu]} = b_i^{[\mu]}$ or $c_i^{[\nu]} = c_i^{[\mu]}$ are imposed. To satisfy all of these order conditions, there are no more than $N * s(s+1)$ independent Butcher coefficients.

Table 1
Number of ARK$_N$ coupling conditions

Eqn. of	$b_i^{[\nu]} \neq b_i^{[\mu]}$	$b_i^{[\nu]} = b_i^{[\mu]}$	$b_i^{[\nu]} \neq b_i^{[\mu]}$	$b_i^{[\nu]} = b_i^{[\mu]}$
cond.	$c_i^{[\nu]} \neq c_i^{[\mu]}$	$c_i^{[\nu]} \neq c_i^{[\mu]}$	$c_i^{[\nu]} = c_i^{[\mu]}$	$c_i^{[\nu]} = c_i^{[\mu]}$
$\tau_1^{(1)}$	0	0	0	0
$\tau_1^{(2)}$	$N(N-1)$	0	0	0
$\tau_1^{(3)}$	$N^2(N+1)/2!-N$	$N(N+1)/2!-N$	0	0
$\tau_2^{(3)}$	N^3-N	$N(N-1)$	$N(N-1)$	0
$q=3$	$N(N-1)(3N+4)/2!$	$3N(N-1)/2!$	$N(N-1)$	0
$q \leqslant 3$	$N(N-1)(3N+6)/2!$	3N(N-1)/2!	$N(N-1)$	0
$\tau_1^{(4)}$	$N^2(N+1)(N+2)/3!-N$	$N(N+1)(N+2)/3!-N$	0	0
$\tau_2^{(4)}$	N^4-N	N^3-N	$N(N-1)$	0
$\tau_3^{(4)}$	$N^3(N+1)/2!-N$	$N^2(N+1)/2!-N$	$N(N-1)$	0
$\tau_4^{(4)}$	N^4-N	N^3-N	N^3-N	$N(N-1)$
$q=4$	$N(N-1)(16N^2+22N+24)/3!$	$N(N-1)(16N+22)/3!$	$N(N-1)(N+3)$	$N(N-1)$
$q \leqslant 4$	$N(N-1)(16N^2+31N+42)/3!$	$N(N-1)(16N+31)/3!$	$N(N-1)(N+4)$	$N(N-1)$
$\tau_1^{(5)}$	$N^2(N+1)(N+2)(N+3)/4!-N$	$N(N+1)(N+2)(N+3)/4!-N$	0	0
$\tau_2^{(5)}$	$N^4(N+1)/2!-N$	$N^3(N+1)/2!-N$	$N(N-1)$	0
$\tau_3^{(5)}$	$N^3(N^2+1)/2!-N$	$N^2(N^2+1)/2!-N$	$N^2(N+1)/2!-N$	$N(N-1)/2!$
$\tau_4^{(5)}$	$N^4(N+1)/2!-N$	$N^3(N+1)/2!-N$	$N(N-1)$	0
$\tau_5^{(5)}$	$N^3(N+1)(N+2)/3!-N$	$N^2(N+1)(N+2)/3!-N$	$N(N-1)$	0
$\tau_6^{(5)}$	N^5-N	N^4-N	N^3-N	$N(N-1)$
$\tau_7^{(5)}$	N^5-N	N^4-N	N^3-N	$N(N-1)$
$\tau_8^{(5)}$	$N^4(N+1)/2!-N$	$N^3(N+1)/2!-N$	N^3-N	$N(N-1)$
$\tau_9^{(5)}$	N^5-N	N^4-N	N^4-N	N^3-N
$q=5$	$N(N-1)(125N^3+179N^2+210N+216)/4!$	$N(N-1)(125N^2+179N+210)/4!$	$N(N-1)(2N^2+9N+16)/2!$	$N(N-1)(2N+9)/2!$
$q \leqslant 5$	$N(N-1)(125N^3+243N^2+334N+384)/4!$	$N(N-1)(125N^2+243N+334)/4!$	$N(N-1)(2N^2+11N+24)/2!$	$N(N-1)(2N+11)/2!$

It may also be seen from this that as long as $b_i^{[\nu]} = b_i^{[\mu]}$ (identical root nodes) and $c_i^{[\nu]} = c_i^{[\mu]}$ (identical canopy nodes), an arbitrary number of independent third-order methods having the same number of stages may be coupled together with no associated coupling error. Selecting $b_i^{[\nu]} = b_i^{[\mu]}$ or $c_i^{[\nu]} = c_i^{[\mu]}$ allows second-order error-free coupling of an arbitrary number of schemes. Choosing neither of these assumptions reduces error-free coupling, as seen from Table 1, to first-order. Further, if $c_i^{[\nu]} = c_i^{[\mu]}$ is invoked, only autonomous-problem order conditions need be considered.

2.1.3. Error

Error in an elemental qth-order Runge–Kutta scheme contained within an ARK$_N$ method may be quantified in a general way by taking the L_2 principal error norm [41] $A^{(q+1)} = \|\boldsymbol{\tau}^{(q+1)}\|_2$, where $\boldsymbol{\tau}_j^{(q+1)}$ are the $\alpha_{q+1}^{[1]}$ error coefficients associated with order of accuracy $q+1$. For embedded schemes where $p = q-1$, additional definitions are useful such as

$$\hat{\boldsymbol{\tau}}_k^{(p)} = \frac{1}{\sigma}\sum_i^s \hat{b}_i \Phi_{i,k}^{(p)} - \frac{\alpha}{p!}, \qquad \widehat{A}^{(p+1)} = \left\|\hat{\boldsymbol{\tau}}^{(p+1)}\right\|_2, \qquad D = \mathrm{Max}\left\{\left|a_{ij}^{[\nu]}\right|, \left|b_i^{[\nu]}\right|, \left|\hat{b}_i^{[\nu]}\right|, \left|c_i^{[\nu]}\right|\right\}, \tag{7}$$

$$B^{(p+2)} = \frac{\widehat{A}^{(p+2)}}{\widehat{A}^{(p+1)}}, \qquad C^{(p+2)} = \frac{\|\hat{\boldsymbol{\tau}}^{(p+2)} - \boldsymbol{\tau}^{(p+2)}\|_2}{\widehat{A}^{(p+1)}}, \qquad E^{(p+2)} = \frac{A^{(p+2)}}{\widehat{A}^{(p+1)}}, \tag{8}$$

where the superscript circumflex denotes the values with respect to the embedded method. In the case of ARK$_N$ methods, there is generally no reason to assume any particular order condition is more important than another, therefore we generalize the traditional expression for $A^{(q+1)}$ to

$$A^{(q+1)} = \left\|\boldsymbol{\tau}^{(q+1)}\right\|_2 = \sqrt{\sum_{k=1}^{\alpha_{q+1}^{[1]}} \sum_{n=1}^{n_{N,k,(q+1)}} \left(\boldsymbol{\tau}_{k[n]}^{(q+1)}\right)^2}, \tag{9}$$

where we have used $n_{N,k,(q+1)}$ to denote the total number of ARK$_N$ order conditions arising from all variations of the kth 1-tree at order $(q+1)$. In certain special cases one may be able to rule out extended subquadrature, or nonlinear order conditions based on the structure of the ODE at hand [69]. To evaluate the value of the error estimate, one may evaluate the quality parameters $B^{(p+2)}$, $C^{(p+2)}$, and $E^{(p+2)}$ using each $\tau_k^{(q+1)}$ from each elemental method or using every $\tau_{k[n]}^{(q+1)}$. All embedded schemes considered here are applied in local extrapolation mode. For a given order of accuracy, one strives to minimize $A^{(q+1)}$. Based on experience with $q = p+1$ ERK pairs, $B^{(p+2)}$, $C^{(p+2)}$, and $E^{(p+2)}$ are ideally kept of order unity while D is usually kept less than 20.

2.1.4. Simplifying assumptions

Butcher [8,27] row and column simplifying assumptions will be helpful in designing methods because they can reduce and simplify the order conditions. By surveying Table 1, one quickly surmises that higher-order methods effectively require the use of the assumptions $b_i^{[\nu]} = b_i^{[\mu]} = b_i$ and $c_i^{[\nu]} = c_i^{[\mu]} = c_i$. Also, without identical root or canopy nodes, application of Butcher simplifying assumptions would become very awkward; therefore, simplifying assumptions are considered in the form

$$C^{[\nu]}(\eta, i) \ : \quad \sum_{j=1}^{s} a_{ij}^{[\nu]} c_j^{q-1} = \frac{c_i^q}{q}, \quad i = 1, \ldots, s, \ q = 1, \ldots, \eta, \tag{10}$$

$$D^{[\nu]}(\zeta, j)\ :\quad \sum_{i=1}^{s} b_i c_i^{q-1} a_{ij}^{[\nu]} = \frac{b_j}{q}\bigl(1 - c_j^q\bigr), \quad j = 1, \ldots, s,\ q = 1, \ldots, \zeta. \tag{11}$$

The notation $C^{[\nu]}(\eta, i \neq 2)$ should be interpreted as $C^{[\nu]}(\eta, i)$ for all i except 2.

2.1.5. *Stability*

The linear stability function for N-additive methods is considered using the equation

$$F(U) = \sum_{\nu=1}^{N} \lambda^{[\nu]} U, \tag{12}$$

from which it is determined that the stability function is [10]

$$R\bigl(z^{[1]}, z^{[2]}, \ldots, z^{[N]}\bigr) = \frac{P(z^{[1]}, z^{[2]}, \ldots, z^{[N]})}{Q(z^{[1]}, z^{[2]}, \ldots, z^{[N]})} \tag{13}$$

$$= \frac{\mathrm{Det}[\boldsymbol{I} - \sum_{\nu=1}^{N}(z^{[\nu]}\boldsymbol{A}^{[\nu]}) + \sum_{\nu=1}^{N}(z^{[\nu]}\boldsymbol{e} \otimes \boldsymbol{b}^{[\nu]^{\mathrm{T}}})]}{\mathrm{Det}[\boldsymbol{I} - \sum_{\nu=1}^{N}(z^{[\nu]}\boldsymbol{A}^{[\nu]})]}, \tag{14}$$

where $\boldsymbol{A}^{[\nu]} = a_{ij}^{[\nu]}$, $\boldsymbol{b}^{[\nu]} = b_i^{[\nu]}$, $\boldsymbol{I} = \delta_{ij}$, $z^{[\nu]} = \lambda^{[\nu]}\Delta t$, and $\boldsymbol{e} = \{1, 1, \ldots, 1\}$. Stability for the embedded method is evaluated using the expression above with $\hat{\boldsymbol{b}}^{[\nu]}$ replacing $\boldsymbol{b}^{[\nu]}$. We remark that this form of $F(U)$ is a more severe simplification than the traditional one ($N = 1$) because one cannot, in general, simultaneously diagonalize all N jacobians.

2.1.6. *Conservation*

Conservation of certain integrals or invariants may also be of interest in additive Runge–Kutta methods [3,28]. Similar to the algebraic stability matrix, one may define

$$M_{ij}^{[\nu,\mu]} = b_i^{[\nu]} a_{ij}^{[\mu]} + b_j^{[\mu]} a_{ji}^{[\nu]} - b_i^{[\nu]} b_j^{[\mu]}. \tag{15}$$

ARK$_N$ methods conserve linear first integrals, in general, only if $b_i^{[\nu]} - b_i^{[\mu]} = 0$, and conserve certain quadratic first integrals, in general, only if $b_i^{[\nu]} - b_i^{[\mu]} = 0$ and $M_{ij}^{[\nu,\mu]} = O$, where $i, j = 1, 2, \ldots, s,\ \nu, \mu = 1, 2, \ldots, N,\ \nu \neq \mu$. Conservation of cubic invariants with Runge–Kutta methods is not possible [9,28,36].

2.1.7. *Dense output*

The purpose of dense output [26,50] has traditionally been to allow high-order interpolation of the integration variables at any point, $t^{(n)} + \theta\Delta t$, inside the current integration step where $0 \leqslant \theta \leqslant 1$. It may also be used, albeit more cautiously, for extrapolating integration variable values to enable better stage value guesses when one or more of the elemental methods is implicit [49,55]. For an ARK$_N$ method, it is accomplished as

$$\begin{aligned} U\bigl(t^{(n)} + \theta\Delta t\bigr) &= U^{(n)} + (\Delta t)\sum_{\nu=1}^{N}\sum_{i=1}^{s} b_i^{*[\nu]}(\theta) F^{[\nu]}\bigl(U^{(i)}\bigr), \\ b_i^{*[\nu]}(\theta) &= \sum_{j=1}^{p^*} b_{ij}^{*[\nu]}\theta^j, \qquad b_i^{*[\nu]}(\theta = 1) = b_i^{[\nu]}, \end{aligned} \tag{16}$$

where p^* is the lowest order of the interpolant on the interval $0 \leqslant \theta \leqslant 1$. By construction, $b_i^{*[\nu]}(\theta = 0) = 0$. Order conditions, at order m, for the dense output method are given by

$$\boldsymbol{\tau}_{k[n]}^{*(m)} = \frac{1}{\sigma} \sum_{i}^{s} b_i^* \Phi_{i,k[n]}^{(m)} - \frac{\alpha \theta^m}{m!}. \tag{17}$$

Setting $m = q$ and $\theta = 1$, we retrieve (5). As with the main and embedded formulae, one may write terms like $A^{*(p^*+1)} = A^{*(p^*+1)}(\theta)$ to access the accuracy of the dense output method. When used as an extrapolation device ($\theta > 1$), the stability function $R^*(z^{[1]}, z^{[2]}, \ldots, z^{[N]}, \theta)$ must be considered [6,20]. It is derived by replacing $\boldsymbol{b}^{[\nu]}$ in the traditional stability function with $\boldsymbol{b}^{*[\nu]}(\theta)$. Throughout this paper, we will assume $b_{ij}^{*[\nu]} = b_{ij}^{*[\mu]} = b_{ij}^*$.

3. Implicit–explicit ARK$_2$ methods

Given that the CDR equations have three terms, a natural question is whether an IMEX ARK$_2$ or an ARK$_3$ (one implicit and two explicit) method is more appropriate. Stability of the method should include L-stability of the overall method as well as good stability from explicit portions for vanishing values of the stiff scaled eigenvalue. L-stability imposes three conditions for $N = 2$ and four for $N = 3$. The total number of free tall trees associated with the explicit methods is $(s - q)$ for $N = 2$ and $(s - q)^2$ for $N = 3$. In terms of accuracy, the difference in accumulated number of coupling conditions at orders $\{1, 2, 3, 4, 5, 6\}$ is $\{0, 0, 0, 4, 36, 231\}$. Also, as N increases, the total number of principal error terms increases yielding relatively higher values of $A^{(q+1)}$. As we ultimately seek methods of orders four or five, it would seem prudent to limit further discussion to ARK$_2$ schemes. The equations are therefore cast in the form

$$\frac{\mathrm{d}U}{\mathrm{d}t} = F_{\mathrm{ns}} + F_{\mathrm{s}}, \tag{18}$$

where F_{ns} represents the nonstiff terms and F_{s} represents the stiff terms, and we then consider implicit–explicit ARK$_2$ methods. ERK methods are used to integrate the nonstiff terms. Stiff terms are treated with ESDIRK methods [2,32,43,44]. Coefficients for the ERK and ESDIRK methods will be distinguished by $a_{ij}^{[E]}$ and $a_{ij}^{[I]}$, respectively. ESDIRKs offer the advantages of allowing L-stability, stiff accuracy, and a stage-order of two. They differ from the more traditional SDIRK [1,27] methods by having an explicit first stage. A consequence of allowing a stage-order of two is that algebraic stability becomes impossible [65]. As we will always invoke $b_i^{[E]} = b_i^{[I]} = b_i$, $\hat{b}_i^{[E]} = \hat{b}_i^{[I]} = b_i$ and $c_i^{[E]} = c_i^{[I]} = c_i$ in this paper, their superscripts are henceforth superfluous. In general, an ERK method has $s(s+1)/2$ degrees of freedom (DOF) available to satisfy all order and any other conditions, where s is the number of stages. An ESDIRK has $(s^2 + s + 2)/2$ available DOF. Combining the two into an ARK$_2$ scheme, if each b_i and each c_i are made equal, $(2s - 1)$ DOF are lost, leaving $(s^2 - s + 2)$ DOF. A further assumption of $a_{sj}^{[I]} = b_j$, the stiffly-accurate assumption, reduces this to $(s^2 - 2s + 2)$ and facilitates L-stability as well as forces the stiff part of $U^{(n+1)}$ to be computed implicitly. It is particularly useful in cases of singular-perturbation type problems and, when combined with I-stability, generally tolerates stiffness better than nonstiffly accurate L-stable methods. Incorporating stiff accuracy and a stage-order of 2 into all of the ESDIRKs, the IMEX ARK$_2$ methods then take the form

$$\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
2\gamma & 2\gamma & 0 & 0 & 0 & \cdots & 0 \\
c_3 & a_{31}^{[E]} & a_{32}^{[E]} & 0 & 0 & \ddots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\
c_{s-1} & a_{s-1,1}^{[E]} & a_{s-1,2}^{[E]} & a_{s-1,3}^{[E]} & \ddots & 0 & 0 \\
1 & a_{s,1}^{[E]} & a_{s,2}^{[E]} & a_{s,3}^{[E]} & \cdots & a_{s,s-1}^{[E]} & 0 \\
\hline
 & b_1 & b_2 & b_3 & \cdots & b_{s-1} & \gamma \\
\hline
 & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \cdots & \hat{b}_{s-1} & \hat{b}_s
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
2\gamma & \gamma & \gamma & 0 & 0 & \cdots & 0 \\
c_3 & a_{31}^{[I]} & a_{32}^{[I]} & \gamma & 0 & \ddots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\
c_{s-1} & a_{s-1,1}^{[I]} & a_{s-1,2}^{[I]} & a_{s-1,3}^{[I]} & \cdots & \gamma & 0 \\
1 & b_1 & b_2 & b_3 & \cdots & b_{s-1} & \gamma \\
\hline
 & b_1 & b_2 & b_3 & \cdots & b_{s-1} & \gamma \\
\hline
 & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \cdots & \hat{b}_{s-1} & \hat{b}_s,
\end{array}$$

where $\gamma = a_{ii}^{[I]}$, $i = 2, 3, \ldots, s$ and should not be confused with the density of an N-tree, γ, discussed in Section 2.1.1.

To identify the schemes derived in this paper, a nomenclature similar to that originally devised by Dormand and Prince is followed [41]. Schemes will be named ARK$q(p)sS[q_{\text{so}}]X$, where q is the order of the main method, p is the order of the embedded method, s is the number of stages, S is some stability characterization of the method, q_{so} is the stage-order of the implicit method, and X is used for any other important characteristic of the method. For S, we use L to denote an L-stable ARK$_2$. Distinguishing between L-stable methods that are or are not stiffly accurate is important; hence, we use X as *SA* to denote stiffly accurate.

3.1. *Design*

3.1.1. *Accuracy*

Both ERK and ESDIRK methods are subject to the $\{1, 1, 2, 4, 9, 20\}$ order conditions for orders $\{1, 2, 3, 4, 5, 6\}$. These order conditions are listed up to fifth-order for 1-trees in Appendix A and to fifth-order for general 2-trees in the extended paper [42]. With the assumptions $b_i^{[I]} = b_i^{[E]} = b_i$ and $c_i^{[I]} = c_i^{[E]} = c_i$, there are $\{0, 0, 0, 2, 13, 63\}$ coupling conditions at the same orders (see [26, Section II.15]). At fourth- and fifth-order, these coupling order conditions are shown as bicolored trees in Fig. 1.

Further reduction of the number and complexity of order conditions is possible by using Butcher simplifying assumptions. Unfortunately, they may conflict with one another. For instance, applying assumptions $D^{[E]}(1, j)$ and $D^{[I]}(1, j)$ gives rise to two inconsistent equations at $j = s$,

$$b_s\gamma = b_s(1 - c_s), \qquad 0 = b_s(1 - c_s). \tag{19}$$

The first implies $c_s = 1 - \gamma$ while the second implies either $b_s = 0$ or $c_s = 1$. Hence, no value of c_s can satisfy both equations and consequently the column simplifying assumption can only be applied to one of the methods. In conjunction with the stiffly accurate assumption which forces $c_s = 1$, it may only be used on the ERK. We avoid the option of setting both $a_{11}^{[I]}$ and $a_{ss}^{[I]}$ to zero because of the complication that will arise in enforcing L-stability and the possibility of explicitly computed stage values. Additional interscheme conflict occurs upon imposition of $C^{[E]}(3, i \neq 2)$ and $C^{[I]}(3, i)$. A stage-order of two on the stiffly accurate ESDIRK is imposed by enforcing $C^{[I]}(2, i) = 0$ for $i = 2, \ldots, (s - 1)$. A stage-order of three is impossible because of the second stage where $\sum_{j=1}^{s} a_{2j}^{[I]} c_j^2 - c_2^3/3 = 4\gamma^3/3 \neq 0$. Reducing the truncation error of the second stage is clearly facilitated by smaller values of γ.

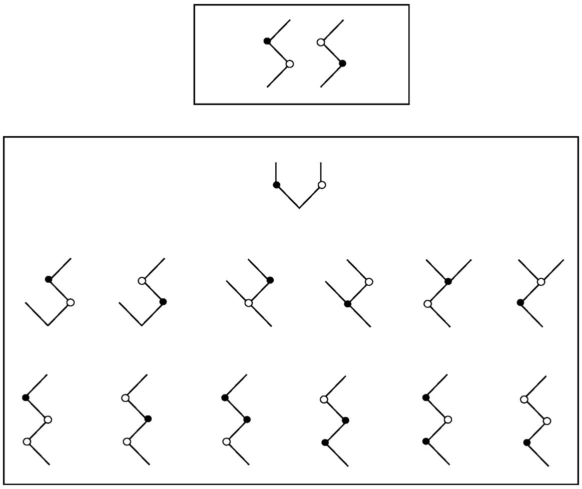

Fig. 1. Fourth- and fifth-order bicolored coupling conditions for $b_i^{[I]} = b_i^{[E]}$ and $c_i^{[I]} = c_i^{[E]}$.

3.1.2. Stability

Linear stability for an ERK–ESDIRK ARK$_2$ method with $b_i^{[I]} = b_i^{[E]} = \boldsymbol{b}$ is analyzed using the stability function [10]

$$R\left(z^{[E]}, z^{[I]}\right) = \frac{\mathrm{Det}[\boldsymbol{I} - z^{[E]}\boldsymbol{A}^{[E]} - z^{[I]}\boldsymbol{A}^{[I]} + (z^{[E]} + z^{[I]})\boldsymbol{e} \otimes \boldsymbol{b}^{\mathrm{T}}]}{\mathrm{Det}[\boldsymbol{I} - z^{[I]}\boldsymbol{A}^{[I]}]} = \frac{P(z^{[E]}, z^{[I]})}{Q(z^{[I]})}, \tag{20}$$

where

$$P\left(z^{[E]}, z^{[I]}\right) = \sum_{i=0}^{s} \left\{ \sum_{j=0}^{s-i} p_{ij} \left(z^{[E]}\right)^j \right\} \left(z^{[I]}\right)^i, \qquad Q\left(z^{[I]}\right) = 1 + \sum_{i=1}^{s-1} q_i \left(z^{[I]}\right)^i = \left(1 - \gamma z^{[I]}\right)^{s-1}. \tag{21}$$

To recover the ERK stability function only, $p_{ij} = p_{0j}$ or $z^{[I]} = 0$ and $q_i = 0$. The ESDIRK is retrieved with $p_{ij} = p_{i0}$ or $z^{[E]} = 0$. Both the ARK$_2$ and ESDIRK methods share the same q_i coefficients. In all cases $p_{00} = 1$. A total of $(s-1)(s-2)/2$ of the p_{ij} coefficients are coupling stability terms where $i, j \neq 0$. For an ESDIRK to be L-stable, it is required that $\gamma > 0$ so that the stability function remains analytic in the complex left-half-plane, the method must be I-stable, and $p_{s,0} = p_{s-1,0} = 0$ so that the stability function vanishes as $z^{[I]}$ tends toward infinity. I-stability of the ESDIRK method is determined using the E-polynomial [27] given by

$$E(y) = Q(+iy)Q(-iy) - P(0, +iy)P(0, -iy) = \sum_{j=0}^{s} E_{2j} y^{2j}, \tag{22}$$

where $i = \sqrt{-1}$. Imaginary axis (I-)stability requires that $E(y) \geqslant 0$ for all real values of y. It is sufficient but not necessary to have all $E_{2j} \geqslant 0$. An L-stable ESDIRK will have $E_{2s} = 0$. An order q ESDIRK will have $E_{2j} = 0$ for $2j \leqslant q$.

Above and beyond L-stability of the ESDIRK method, it may be useful to control the damping of the large scaled eigenvalues, $z^{[I]}$, at intermediate stages [70]. The internal stability function at the

nth-stage may be constructed for DIRK-type methods by using portions of the $a_{ij}^{[I]}$ matrix. Denoting $a_{ij}^{[I]},\ i,j=1,2,\ldots,n$ as $\mathcal{A}$ and $a_{nj}^{[I]},\ j=1,2,\ldots,n$ as $\mathcal{B}_j$, the internal stability function is given by

$$R_{\text{int}}^{(n)}\bigl(z^{[I]}\bigr)=\frac{\operatorname{Det}[\mathcal{I}-z^{[I]}\mathcal{A}^{[I]}+z^{[I]}\mathcal{E}\otimes\mathcal{B}^{\mathrm{T}}]}{\operatorname{Det}[\mathcal{I}-z^{[I]}\mathcal{A}^{[I]}]}=\frac{P_{\text{int}}^{(n)}(z^{[I]})}{Q_{\text{int}}^{(n)}(z^{[I]})},\tag{23}$$

where $\mathcal{I}$ is the $(n\times n)$ identity matrix, and $\mathcal{E}$ is the one-vector of length n. Our concern will be the value of $R_{\text{int}}^{(n)}(-\infty)$. $P_{\text{int}}^{(n)}$ is, in general, a polynomial of degree $n-1$ in $z^{[I]}$ because $\mathcal{A}_{nj}=\mathcal{B}_j$ while $Q_{\text{int}}^{(n)}$ is, in general, of degree n. Consequently, SDIRKs have $R_{\text{int}}^{(n)}(-\infty)=0$. ESDIRKs, with $Q_{\text{int}}^{(n)}$ reduced to degree $n-1$ because $a_{11}^{[I]}=0$, do not generally satisfy $R_{\text{int}}^{(n)}(-\infty)=0$.

In terms of step-wise stability, choosing the stiffly accurate assumption forces $p_{s,0}=0$. Placing $a_{i1}^{[I]}=0$ forces $p_{s-1,0}=0$ but sacrifices $(s-1)$ DOF and the possibility of higher stage-order. A consequence of setting $a_{11}^{[I]}=0$, what effectively distinguishes the ESDIRK from the SDIRK, is that it forces $q_s=0$. Achieving an L-acceptable stability function for the ARK$_2$,

$$R\bigl(z^{[E]},z^{[I]}\bigr)=\frac{1+\cdots+\{p_{s-1,1}z^{[E]}+p_{s-1,0}\}(z^{[I]})^{s-1}+p_{s,0}(z^{[I]})^{s}}{1+\cdots+(-\gamma)^{s-1}(z^{[I]})^{s-1}},\tag{24}$$

and not just the ESDIRK, is now more complicated because of $p_{s-1,1}$. In both the ESDIRK and IMEX ARK$_2$ cases, $p_{s,0}$ and $p_{s-1,0}$ must vanish for L-stability, but the IMEX scheme must also satisfy $p_{s-1,1}=0$. Several of the methods given by Ascher et al. [5] and Griepentrog [21] do not account for this and consequently have $R(z^{[E]},-\infty)$ depending linearly on $z^{[E]}$. Similar comments apply to $\hat{p}_{ij}$ and $p_{ij}^{*}=\sum_{k=1}^{p^{*}}p_{ijk}^{*}\theta^{k}$, the coefficients of the P-polynomials for the embedded and dense-output formulae. A benefit of a zero-column in $a_{ij}^{[I]},\ j=1$, for an additive method is that on problems such as Kreiss's problem [15,27], which may act like an index-2 differential algebraic equation, the initial value of the algebraic variable is not propagated along with the solution [38]. Ascher et al. [5] and Calvo et al. [10] both choose $a_{sj}^{[I]}=b_j$ as well as zero-padded SDIRKs ($a_{i1}^{[I]}=0$) and consequently their methods perform relatively well on Kreiss's problem.

3.1.3. Conservation

As the scheme weights, b_i, for the current ERK and ESDIRK methods are the same, linear first integrals will be conserved. Certain quadratic first integrals will, however, only be conserved if linear first integrals are conserved and $M_{ij}^{[E,I]}=0$ vanishes. We will adopt the point of view that although none of the methods will make this vanish, minimizing the magnitude of this matrix,

$$\bigl\|M_{ij}^{[E,I]}\bigr\|=\sqrt{\sum_{i,j=1}^{s}M_{ij}^{[E,I]}M_{ji}^{[E,I]}},\tag{25}$$

should enhance conservation characteristics.

3.2. Implementation

3.2.1. Stage values

Using the definitions $F_{\mathrm{ns}}^{(j)} \simeq F_{\mathrm{ns}}(U^{(j)}, t^{(n)} + c_j \Delta t)$ and $F_{\mathrm{s}}^{(j)} \simeq F_{\mathrm{s}}(U^{(j)}, t^{(n)} + c_j \Delta t)$, one must solve

$$U^{(i)} = U^{(n)} + X^{(i)} + (\Delta t)\gamma F_{\mathrm{s}}^{(i)}, \quad i \geqslant 2, \qquad X^{(i)} = (\Delta t)\sum_{j=1}^{i-1}\left(a_{ij}^{[E]} F_{\mathrm{ns}}^{(j)} + a_{ij}^{[I]} F_{\mathrm{s}}^{(j)}\right), \tag{26}$$

where $X^{(i)}$ is explicitly computed from existing data. Combining this with an appropriate starting guess, a modified [23,33,58,60] Newton iteration provides $U^{(i)}$ and $F_{\mathrm{s}}^{(i)}$. In cases where direct methods are appropriate, this is accomplished by solving

$$\left(I - (\Delta t)\gamma \left.\frac{\partial F_s}{\partial U}\right|_k\right)\left(U^{(i)} - U_k^{(i)}\right) = -\left(U_k^{(i)} - U^{(n)}\right) + X^{(i)} + (\Delta t)\gamma F_s\left(U_k^{(i)}\right), \tag{27}$$

where the subscript k denotes the value on the kth iteration, $M = [I - \gamma(\Delta t)(\partial F_s/\partial U)|_k]$ is the iteration matrix, and $d_k^{(i)} = (U^{(i)} - U_k^{(i)})$ is the displacement. On the kth iteration one has

$$M d_k^{(i)} = r_k^{(i)}, \qquad U_{k+1}^{(i)} = U_k^{(i)} + M^{-1} r_k^{(i)}, \tag{28}$$

where $r_k^{(i)}$ is the residual. The iteration is terminated when either $d_k^{(i)}$ (displacement test) or $r_k^{(i)}$ (residual test) are sufficiently small [33,66],

$$\tau_{\text{residual}} = c\epsilon \geqslant r_k^{(i)}, \quad \text{or} \quad \tau_{\text{displacement}} = c\epsilon \geqslant d_k^{(i)}, \quad c \approx 0.005,$$

where ϵ is the integration error tolerance and c is the tolerance ratio. $F_{\mathrm{ns}}^{(i)}$ may then be computed using $U^{(i)}$. Inexact [30,40,53] Newton methods may be more appropriate for larger systems of coupled equations.

3.2.2. Stage-value predictors

Stage-value iteration convergence rates may be substantially improved and convergence failures may be minimized by choosing a good starting guess. The most primitive approach to obtaining a guess for the integration variables at the next stage is to use the most recent stage values; the trivial guess. An oftentimes better way to obtain stage-value starting guesses is by using a dense output formula [27,32,49,55]. Second and later steps may use the function evaluations from the previous step to extrapolate into the current step. Stage-value guesses for the ith stage of the step $n+1$ are obtained using function evaluations from step n as

$$U^{(i)}\left(t^n + \theta_i \Delta t\right) = U^{(n)} + (\Delta t)\sum_{j=1}^{s} b_j^*(\theta_i)\left(F_{\mathrm{ns}}^{(j)} + F_{\mathrm{s}}^{(j)}\right), \quad \theta_i = 1 + r c_i, \quad r = \frac{(\Delta t)^{(n+1)}}{(\Delta t)^{(n)}}. \tag{29}$$

Shortcomings of this approach include order-of-accuracy reduction when an interpolation formula is used in extrapolation mode and the introduction of instabilities into the extrapolated guess. As is sometimes done with the implicit error control estimate when substantial stiffness is present, one may wish to smooth the predicted stage value by multiplying it by the iteration matrix [32,59]. More sophisticated predictors have been derived relevant to DIRK methods [20,29,45,56]. We do not consider these, in part, because computer memory management may become a problem. To conserve memory usage during extrapolation, all s-stages may be estimated at the beginning of the step and function values might then be overwritten by stage-value guesses, one equation at a time. For large r, the trivial guess may be more prudent.

3.2.3. Error and step-size control

Step-size control is a means by which accuracy, iteration, and to a lesser extent stability are controlled. The choice of the (Δt) may be chosen from many criteria, among those are the (Δt) from the accuracy based step controller, the $(\Delta t)_{\text{inviscid}}$ and $(\Delta t)_{\text{viscous}}$ associated with the inviscid and viscous stability limits of the ERK, and the $(\Delta t)_{\text{iter}}$ associated with iteration convergence [23]. If error control reliability is sufficient, CFL numbers may be removed as their function would be superfluous. For $q = p + 1$ pairs, one could consider timestep control of the IMEX schemes using I-, PI-, and PC-controllers [27] or a PID-controller [22,63,64]

$$(\Delta t)^{(n+1)}_{\text{PID}} = \kappa (\Delta t)^{(n)} \left\{ \frac{\epsilon}{\|\delta^{(n+1)}\|_\infty} \right\}^{\alpha} \left\{ \frac{\|\delta^{(n)}\|_\infty}{\epsilon} \right\}^{\beta} \left\{ \frac{\epsilon}{\|\delta^{(n-1)}\|_\infty} \right\}^{\gamma}, \tag{30}$$

with error estimate $\delta^{(n+1)} = U^{(n+1)} - \widehat{U}^{(n+1)}$, $\kappa \approx 0.9$, ϵ is a user specified tolerance, and p is the order-of-accuracy of the embedded method. The I-, PI-, and PID-controllers are appropriate to explicit methods. Implicit methods use either I- or PC-controllers. A PID-controller is considered because many stability optimized ERK methods are SC-unstable with a PI-controller for eigenvalues on the real axis [41]. Its characteristic roots are those of the equation $\zeta^3 + (p\alpha - 1)\zeta^2 - p\beta\zeta + p\gamma = 0$. Individual controller gains are obtained via $k_I = p(\alpha - \beta + \gamma)$, $k_P = p(\beta - 2\gamma)$, and $k_D = p\gamma$. One may wish to keep controller gains fixed, independent of stepsize changes. This may be done by preselecting gains, then using $\omega^{(n)} = (\Delta t)^{(n)}/(\Delta t)^{(n-1)}$ and

$$p\alpha = \left[k_I + k_P + \left(\frac{2\omega^{(n)}}{1+\omega^{(n)}} \right) k_D \right], \qquad p\beta = \left[k_P + 2\omega^{(n)} k_D \right], \qquad p\gamma = \left(\frac{2\omega^{2(n)}}{1+\omega^{(n)}} \right) k_D. \tag{31}$$

In the present context, we have selected $k_I = 0.25$, $k_P = 0.14$, $k_D = 0.10$, or $\alpha = 0.49/p$, $\beta = 0.34/p$, $\gamma = 0.10/p$ when $\omega^{(n)} = 1$, giving characteristic roots of $\{-0.518, 0.247, 0.781\}$. The controller is SC-stable at all stability boundary points of the ERK in the complex left-half-plane for each of the three proposed methods. Inclusion of second-derivative gain, a PIDD^2-controller, was not found to enhance control. This strategy may not behave well if substantial order-reduction occurs due to extreme stiffness because, at a minimum, p no longer reflects the actual order of the embedded method. Stability based time step limits involving the inviscid and viscous CFL numbers, (λ, λ_v), are given by $(\Delta t)_{\text{inviscid}} \approx \lambda(\Delta x)/(u + a)$ and $(\Delta t)_{\text{viscous}} \approx \lambda_v(\Delta x)^2/\nu$ where a is the local speed of sound, u is the magnitude of local fluid convection speed, and ν is an appropriate diffusivity of either mass, momentum, or energy [41]. For implicit–explicit methods, we select

$$(\Delta t)^{(n+1)} = \text{Min}\left\{ (\Delta t)^{(n+1)}_{\text{error}}, (\Delta t)_{\text{inviscid}}, (\Delta t)_{\text{viscous}}, (\Delta t)_{\text{iteration}} \right\}. \tag{32}$$

It remains to be determined which controller(s) are best suited to IMEX methods.

4. Third-order methods

A third-order, 3(2) pair, ARK$_2$ scheme is designed in four stages using $b_i^{[I]} = b_i^{[E]}$, $c_i^{[I]} = c_i^{[E]}$, and simplifying assumption $C^{[I]}(2, i)$. Stiff accuracy and a stage-order of two are incorporated into the four-

Table 2
ARK3(2)4L[2]SA dense output method

Property	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$	$\theta = 5$
$A^{*(3)}(\theta)$	0	1.098	4.809	12.87	27.02
$A^{*(4)}(\theta)$	0.07217	2.280	14.18	48.23	121.9
$R^{*}(z^{[E]}, -\infty, \theta)$	0	5.789	18.37	37.74	63.89

stage ESDIRK method. The main method is obtained by solving for the $s^2 - 2s + 2 = 10$ DOF ($s = 4$) using

$$\begin{aligned} &\tau_1^{(k)} = 0, \quad k = 1, 2, 3, \qquad p_{30} = p_{31} = 0, \qquad c_3 = 3/5, \\ &{}^{[E]}\tau_2^{(3)} = 0, \qquad {}^{[E]}\Phi_4^{(4)} = 1/35, \qquad \sum_{j=1}^{s} a_{ij}^{[I]} c_j = c_i^2/2, \quad i = 2, 3, \end{aligned} \tag{33}$$

with c_3 and ${}^{[E]}\Phi_4^{(4)}$ being used for optimization. Total fourth-order principal error is $A^{(4)} = 0.07217$. Linear stability of the ERK is given by the CFL numbers $(\lambda, \lambda_v) = (1.24, 0.92)$. Solving p_{30} for γ results in a cubic equation, $6\gamma^3 - 18\gamma^2 + 9\gamma - 1 = 0$, having three roots. Only one of these gives I-stability, $\gamma \approx 0.4358665215084589994160 19$. An embedded method is found by enforcing

$$\hat{\tau}_1^{(k)} = 0, \quad k = 1, 2, \qquad \hat{p}_{40} = 0, \qquad \hat{p}_{30} = -3q_3/40, \tag{34}$$

yielding $\widehat{R}(-\infty) = -0.075 - 0.087z^{[E]}$ and $(\hat{\lambda}, \hat{\lambda}_v) = (1.12, 0.87)$. Second-order dense output is achieved by solving for $b_i^* = \sum_{j=1}^{2} b_{ij}^* \theta^j$ and selected $p_{ij}^* = \sum_{k=1}^{2} p_{ijk}^* \theta^k$ using

$$\tau_1^{*(k)} = 0, \quad k = 1, 2, \qquad \sum_{j=1}^{2} b_{4j}^* = b_4, \qquad p_{40}^* = p_{312}^* = 0. \tag{35}$$

Its properties are summarized in Table 2. Coefficients of the scheme, ARK3(2)4L[2]SA, are given in Appendix C. Characteristics of the method are listed in Appendix B.

5. Fourth-order methods

A five-stage ARK_2 method using a stiffly-accurate, L-stable, stage-order 2, ESDIRK method must satisfy $24\gamma^4 - 96\gamma^3 + 72\gamma^2 - 16\gamma + 1 = 0$. Of the four roots to this equation, only one leads to an L-stable method resulting in $c_2 = 2\gamma \approx 1.14563212496426971081600277$. Further, the minimum principal error associated with the two free-parameter family of five-stage, fourth-order, stage-order 2, stiffly-accurate, L-stable ESDIRKs is $A^{(5)} = 0.03855$, approximately 15 times greater than SDIRK4 [27]. In spite of these shortcomings, one may construct a fourth-order, 4(2), ARK_2 pair using identical root and canopy nodes as well as row simplifying assumption $C^{[I]}(2, i)$. With 17 main and 5 embedded available DOF, one may solve

$$\tau_1^{(k)} = 0, \quad k = 1, 2, 3, 4, \qquad \sum_{j=1}^{s} a_{ij}^{[I]} c_j = c_i^2/2, \quad i = 2, 3, 4,$$

$$p_{40} = p_{41} = {}^{[E]}\tau_2^{(3)} = {}^{[E]}\tau_{2,3,4}^{(4)} = {}^{[I]}\tau_3^{(4)} = 0, \qquad \sum_{i,j=1}^{s} b_i a_{ij}^{[I]} a_{ij}^{[E]} c_k = 1/4!, \tag{36}$$
$$c_3 = 50/100, \qquad c_4 = 95/100, \qquad \hat{\tau}_1^{(k)} = 0, \quad k = 1, 2, \qquad \hat{\tau}_2^{(3)} = \hat{p}_{50} = 0, \qquad \hat{p}_{40} = -q_4/10,$$

and obtain a leading order principal error norm of $A^{(5)} = 0.07664$. Linear stability limits for the explicit method, in terms of the inviscid and viscous CFL numbers, are $(\lambda, \lambda_v) = (1.38, 0.67)$. An optimization of the embedded method could follow that of Tsitouras and Papakostas [67].

For the design of a fourth-order, 4(3), ARK$_2$ pair, we again use the simplifying assumptions $b_i^{[I]} = b_i^{[E]}$ and $c_i^{[I]} = c_i^{[E]}$. Only stiffly accurate, stage-order 2, ESDIRK methods are employed. Using six stages permits $s^2 - 2s + 2 = 26$ DOF ($s = 6$). The value of γ must be

$$0.24799463621274745516799 10 \leqslant \gamma \leqslant 0.67604239322628132887238 63, \tag{37}$$

for I-stability [27]. Besides facilitating better iterative convergence of the modified Newton method, smaller values of γ tend to result in lower truncation error in the implicit method. For simplicity, we use $\gamma = 1/4$. With assumptions $C^{[I]}(2, i)$ and $C^{[E]}(2, i \neq 2)$, ARK4(3)6L[2]SA satisfies

$$\begin{aligned}
&\tau_1^{(k)} = 0, \quad k = 1, 2, 3, 4, \qquad \sum_{j=1}^{s} a_{ij}^{[I]} c_j = c_i^2/2, \quad i = 2, 3, 4, 5,\\
&\sum_{j=1}^{s} a_{ij}^{[E]} c_j = c_i^2/2, \quad i = 3, 4, 5, 6,\\
&b_2 = p_{50} = p_{51} = {}^{[E]}\tau_3^{(4)} = {}^{[I]}\tau_3^{(4)} = \sum_{i=1}^{s} b_i a_{i2}^{[I]} = \sum_{i=1}^{s} b_i a_{i2}^{[E]} = 0,\\
&c_3 = 332/1000, \qquad c_4 = 62/100, \qquad c_5 = 85/100, \qquad \gamma = 1/4,\\
&{}^{[E]}\tau_5^{(5)} = 1/8000, \qquad {}^{[E]}\Phi_9^{(5)} = 1/135, \qquad {}^{[E]}\Phi_{20}^{(6)} = 1/1250,
\end{aligned} \tag{38}$$

where $A^{(5)} = 0.01224$ and $(\lambda, \lambda_v) = (2.01, 1.06)$. The embedded method for this scheme is found by solving

$$\hat{\tau}_1^{(k)} = 0, \quad k = 1, 2, 3, \qquad \hat{b}_2 = \hat{p}_{60} = 0, \qquad \hat{p}_{50} = -3q_5/20, \tag{39}$$

to find $\widehat{R}(-\infty) = -0.150 - 0.040z^{[E]}$ and $(\hat{\lambda}, \hat{\lambda}_v) = (1.94, 1.10)$. Dense output may be approached by either maximizing accuracy or stability. If the method is to be used for interpolation, then a third-order method is appropriate. For extrapolation, stability may be more important and a second-order method is constructed. Third-order dense, or continuous, output is achieved by solving for $b_i^* = \sum_{j=1}^{3} b_{ij}^* \theta^j$ and certain $p_{ij}^* = \sum_{k=1}^{3} p_{ijk}^* \theta^k$ with

$$\begin{aligned}
&\tau_1^{*(k)} = 0, \quad k = 1, 2, 3, \qquad b_2^* = p_{60}^* = 0,\\
&p_{513}^* = -3q_5, \qquad p_{502}^* = -3q_5, \qquad \sum_{j=1}^{3} b_{6j}^* = b_6.
\end{aligned} \tag{40}$$

Second-order dense output is achieved with $p^* = 2$ by solving

$$\tau_1^{*(k)} = 0, \quad k = 1, 2, \qquad b_2^* = p_{60}^* = p_{51}^* = p_{502}^* = 0, \qquad \sum_{j=1}^{2} b_{6j}^* = b_6. \tag{41}$$

Table 3
ARK4(3)6L[2]SA dense output method

Property	$\theta=1$	$\theta=2$	$\theta=3$	$\theta=4$	$\theta=5$
2nd-order					
$A^{*(3)}(\theta)$	0	1.106	5.049	13.56	28.38
$A^{*(4)}(\theta)$	0	2.459	14.79	49.52	124.1
$R^{*}(z^{[E]},-\infty,\theta)$	0	−1	−2	−3	−4
3rd-order					
$A^{*(4)}(\theta)$	0	2.206	12.86	42.48	106.5
$A^{*(5)}(\theta)$	0.01224	3.852	30.61	132.1	410.2
$R^{*}(z^{[E]},-\infty,\theta)$	0	21.2–10.4 $z^{[E]}$	92.6–49.1 $z^{[E]}$	242–134 $z^{[E]}$	499–284 $z^{[E]}$

Properties of these two methods are given in Table 3. Characteristics and coefficients of the scheme ARK4(3)6L[2]SA are given in Appendices B and C.

6. Fifth-order methods

In principle, one may construct a fifth-order ARK$_2$ method in seven stages using simplifying assumptions $b_i^{[I]}=b_i^{[E]}$, $c_i^{[I]}=c_i^{[E]}$, $C^{[E]}(2,i\neq 2)$, $D^{[E]}(1,j)$, and $C^{[I]}(2,i)$. L-stable methods require that

$$0.18391465367517517516323214436 \leqslant \gamma \leqslant 0.33414236706805043595403 01. \tag{42}$$

With the 37 DOF available, minimally, the following 33 equations must be solved

$$\begin{aligned}
&\tau_1^{(k)}=0,\quad k=1,2,\ldots,5, \qquad \sum_{i=1}^{s} b_i a_{ij}^{[E]}=b_j(1-c_j),\quad j=2,3,\ldots,6,\\
&\sum_{j=1}^{s} a_{ij}^{[I]}c_j=c_i^2/2,\quad i=2,3,\ldots,6, \qquad \sum_{j=1}^{s} a_{ij}^{[E]}c_j=c_i^2/2,\quad i=3,4,\ldots,6,\\
&b_2=p_{60}=p_{61}={}^{[E]}\tau_4^{(5)}={}^{[I]}\tau_3^{(4)}={}^{[I]}\tau_{4,5,8}^{(5)}=\frac{1}{2}\sum_{i,j=1}^{s} b_i a_{ij}^{[I]}a_{jk}^{[E]}c_k^2-1/5!=0,\\
&\sum_{i=1}^{s} b_i a_{i2}^{[I]}=\sum_{i=1}^{s} b_i c_i a_{i2}^{[E]}=\sum_{i=1}^{s} b_i c_i a_{i2}^{[I]}=\sum_{i,j=1}^{s} b_i a_{ij}^{[I]}a_{j2}^{[E]}=\sum_{i,j=1}^{s} b_i a_{ij}^{[I]}a_{j2}^{[I]}=0.
\end{aligned} \tag{43}$$

To solve this system of equations, one must solve for at least one abscissa directly. Given the size of the system, it is more fruitful to temporarily ignore $\sum_{i=1}^{s} b_i c_i a_{i2}^{[I]}=0$ and $\sum_{i,j=1}^{s} b_i a_{ij}^{[I]}a_{j2}^{[I]}=0$, include ${}^{[E]}\tau_{20}^{(6)}=0$, specify c_3, c_4, c_5, and c_6, and investigate if the scheme merits further effort. With these changes, both implicit and explicit methods are fifth-order but the coupling method is only fourth-order. We have been unable to find any promising solutions to this method.

Adding a stage, we now consider an eight-stage 5(4) pair. Eight stages permit $s^2-2s+2=50$ DOF in the main method and 8 in the embedded method. The primary difficulty in designing a 5(4) pair is reducing the number of embedded order conditions while simultaneously keeping the implicit portion L-stable and keeping the tall trees of the explicit method well placed. We select simplifying assumptions $b_i^{[I]} = b_i^{[E]}$, $c_i^{[I]} = c_i^{[E]}$, $C^{[I]}(3, i \neq 2)$, $D^{[E]}(1, j)$, and $C^{[E]}(2, i \neq 2)$. In addition, we set the lower elements within the second column of each a_{ij} matrix to zero. ARK5(4)8L[2]SA is constructed according to the following conditions

$$\begin{aligned}
&\tau_1^{(k)} = 0, \quad k = 1, 2, \ldots, 5, \qquad \sum_{i=1}^{s} b_i a_{ij}^{[E]} = b_j(1 - c_j), \quad j = 3, 4, \ldots, 7, \\
&\sum_{j=1}^{s} a_{ij}^{[I]} c_j = c_i^2/2, \quad i = 2, 3, \ldots, 7, \qquad \sum_{j=1}^{s} a_{ij}^{[E]} c_j = c_i^2/2, \quad i = 3, 4, \ldots, 7, \\
&\sum_{j=1}^{s} a_{ij}^{[I]} c_j^2 = c_i^3/3, \quad i = 3, 4, \ldots, 7, \\
&a_{i2}^{[I]} = 0, \quad i = 4, 5, \ldots, 7, \qquad a_{i2}^{[E]} = 0, \quad i = 4, 5, \ldots, 8, \\
&b_2 = b_3 = p_{70} = p_{71} = {}^{[E]}\tau_4^{(5)} = {}^{[I]}\tau_5^{(5)} = \sum_{i=1}^{s} b_i a_{i3}^{[I]} = \frac{1}{2} \sum_{i,j=1}^{s} b_i a_{ij}^{[I]} a_{jk}^{[E]} c_k^2 - 1/5! = 0, \\
&\hat{\tau}_1^{(k)} = 0, \quad k = 1, 2, \ldots, 4, \qquad \hat{b}_2 = \hat{b}_3 = \hat{p}_{80} = \hat{\tau}_3^{(4)} = 0, \qquad \hat{p}_{70}/q_7 = 1/5, \\
&c_5 = 92/100, \qquad c_6 = 24/100, \qquad c_7 = 60/100, \qquad \gamma = 41/200, \\
&a_{75}^{[E]} = -1/8, \qquad a_{76}^{[E]} = -1/8,
\end{aligned} \tag{44}$$

where $A^{(6)} = 0.006988$ but $(\lambda, \lambda_v) = (0.43, 0.96)$. The embedded method is characterized by $\widehat{R}(-\infty) = +0.200 + 0.286z^{[E]}$ and $(\hat{\lambda}, \hat{\lambda}_v) = (0.22, 1.07)$. Better 5(4) pairs probably require nine-stages.

A fourth-order dense output formula is not possible. Third-order continuous output, however, is achieved by solving for $b_i^* = \sum_{j=1}^{3} b_{ij}^* \theta^j$ and specific $p_{ij}^* = \sum_{k=1}^{3} p_{ijk}^* \theta^k$ with

$$\begin{aligned}
&\sum_{i=1}^{s} b_i^* c_i^{k-1} = 1/k!, \quad k = 1, 2, 3, \qquad b_2^* = b_3^* = p_{80}^* = p_{71}^* = p_{703}^* = p_{702}^* = 0, \\
&\sum_{j=1}^{3} b_{8j}^* = b_8.
\end{aligned} \tag{45}$$

For ARK5(4)8L[2]SA, the dense output method is characterized in Table 4. Coefficients of the scheme ARK5(4)8L[2]SA are given in Appendix C while method properties are given in Appendix B.

Table 4
ARK5(4)8L[2]SA dense output method

Property	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$	$\theta = 5$
$A^{*(4)}(\theta)$	0	0.295	4.434	21.01	63.59
$A^{*(5)}(\theta)$	0	1.581	21.03	107.8	361.2
$R^*(z^{[E]}, -\infty, \theta)$	0	-1	-2	-3	-4

7. Test problems

To test the IMEX ARK_2 methods that have just been presented, four separate test problems having adjustable stiffness are considered. Ultimately, all equations are singular perturbation problems and each may be evaluated with either unperturbed or the more troublesome perturbed initial conditions.

7.1. Kaps' problem

Dekker and Verwer [15] investigate a nonlinear problem (experiment 7.5.2) originally given by Kaps,

$$\dot{y}_1(t) = -\big(\varepsilon^{-1}+2\big)y_1(t) + \varepsilon^{-1}y_2^2(t), \qquad \dot{y}_2(t) = y_1(t) - y_2(t) - y_2^2(t), \tag{46}$$

where $0 \leqslant t \leqslant 1$ and whose exact solution is $y_1(t) = y_2^2(t),\ y_2(t) = \exp(-t)$. Equilibrium (unperturbed) initial conditions are given by $y_1(0) = y_2(0) = 1$. The equations exhibit increasing stiffness as $\varepsilon \to 0$ and, in the limit of $\varepsilon = 0$, the system becomes an index-1 differential algebraic equation system. This may be easily seen by multiplying the first equation by ε to obtain $\varepsilon\dot{y}_1(t) = -(1-2\varepsilon)y_1(t) + y_2^2(t)$. Upon setting $\varepsilon = 0$, it reduces to the algebraic equation $y_1(t) = y_2^2(t)$. In an IMEX formulation, terms multiplied by ε^{-1} are integrated implicitly while all other terms are integrated explicitly.

7.2. Van der Pol's equation

Van der Pol's (vdP) equation is an equation describing nonlinear oscillations where the solutions are damped (amplified) for large (small) values of y_1 [26,27],

$$\dot{y}_1(t) = y_2(t), \qquad \dot{y}_2(t) = \varepsilon^{-1}\big(\big(1-y_1(t)^2\big)y_2(t) - y_1(t)\big). \tag{47}$$

Unperturbed initial conditions are given by $y_1(0) = 2,\ y_2(0) = -0.6666654321121172$. For partitioned integration, the first equation is integrated explicitly while the second is integrated implicitly. Van der Pol's equation develops a very challenging boundary layer at time $t \approx 0.8$ based on these initial conditions. Two test cases are chosen involving different time intervals: (1) $0 \leqslant t \leqslant 0.5$, and (2) $0 \leqslant t \leqslant 1.5$. The first is used to study order reduction while the second tests the error prediction capabilities of the schemes and of the robustness of the error controllers.

7.3. Pareschi and Russo's problem

Pareshi and Russo [51] have constructed a simple test equation which contains both stiff and nonstiff terms,

$$\dot{y}_1(t) = -y_2(t), \qquad \dot{y}_2(t) = y_1(t) + \varepsilon^{-1}\big(\sin\big(y_1(t)\big) - y_2(t)\big). \tag{48}$$

Partitioning for an IMEX scheme, terms multiplied by ε^{-1} are integrated with the implicit method while other terms are integrated explicitly. Initial conditions may be considered in two different forms. Equilibrium initial conditions remove any contribution of the stiff term in the initial conditions. This is accomplished with $y_1(0) = \pi/2,\ y_2(0) = 1$. Nonequilibrium, or perturbed, data is specified by replacing the condition on y_2 with $y_2(0) = 1/2$.

7.4. One-dimensional convection–diffusion–reaction problem

A simplified one-dimensional version of the gas-phase, multicomponent, compressible Navier–Stokes equations with chemical reaction is tested. The simplification assumes no bulk viscosity, no thermal diffusion or its cross effect, no spatial gradients of the transport coefficients (μ, λ, ρD_i), no barodiffusion, ordinary diffusion is representable by an effective Fickian diffusion coefficient, and no body forces are present. With these assumptions, one must solve the system

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u\\ \rho e_0\\ \rho Y_i\end{bmatrix} = -\frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2+p\\ (\rho e_0+p)\,u\\ \rho u Y_i\end{bmatrix} + \begin{bmatrix}0\\ \frac{4\mu}{3}\frac{\partial^2 u}{\partial x^2}\\ \lambda\frac{\partial^2 T}{\partial x^2}+\frac{4u\mu}{3}\frac{\partial^2 u}{\partial x^2}+\frac{4\mu}{3}\left(\frac{\partial u}{\partial x}\right)^2+\frac{\partial}{\partial x}\left(\sum_{i=1}^{\mathrm{ncs}}\rho D_i h_i\frac{\partial Y_i}{\partial x}\right)\\ \rho D_i\frac{\partial^2 Y_i}{\partial x^2}\end{bmatrix} + \begin{bmatrix}0\\0\\0\\ \dot{\omega}_i\end{bmatrix} \tag{49}$$

where the three righthand side terms are F_{C}, F_{D}, and F_{R} with the subscripts C, D, and R denoting convection, diffusion, and reaction, respectively. Also, u is the fluid velocity, T is the temperature, Y_i are the species mass fraction, i is the species index that runs from 1 to ncs (number of chemical species), ρ is the fluid density, p is the pressure, t is time, x is the spatial direction, e_0 is the total specific internal energy, μ is the molecular viscosity, λ is the thermal conductivity, D_i is the effective Fickian diffusion coefficient, h_i is the partial specific enthalpy of species i, and $\dot{\omega}_i$ is the reaction rate of species i. We consider the reaction rate only in modified Arrhenius form without pressure correction terms. Supplementary relations that are needed to solve this system are given in the extended paper [42]. This constitutes $(ncs+2)\times(nx)$ equations that must be solved where nx is the number of grid points. In this work both convection and diffusion are deemed nonstiff, $F_{\mathrm{ns}} = F_{\mathrm{C}} + F_{\mathrm{D}}$, and are integrated using the ERK method while reaction terms are treated as stiff, $F_{\mathrm{s}} = F_{\mathrm{R}}$, and are integrated using the ESDIRK method.

For the purposes of this test problem, five species are included: H_2, O_2, OH, H_2O, and N_2. U^{T} is then $\{\rho, \rho u, \rho e_0, \rho Y_{\mathrm{H_2}}, \rho Y_{\mathrm{O_2}}, \rho Y_{\mathrm{OH_2}}, \rho Y_{\mathrm{H_2O}}\}$. Wherever possible, an attempt is made to match the thermophysical properties of these molecules. A two-step reaction mechanism is employed having one reversible and one irreversible step

$$\mathrm{H_2}+\mathrm{O_2}\overset{k_1}{\underset{k_2}{\rightleftharpoons}}2\mathrm{OH},\qquad \frac{1}{2}\mathrm{H_2}+\mathrm{OH}\xrightarrow[k_3]{}\mathrm{H_2O}, \tag{50}$$

where $k_1 = k_2 \gg k_3$. Values of specific reaction rate constants, k_i, used in this test problem are not those of the supplementary relations but have fixed temperature prefactors, temperature exponents, and activation energies. In this way, one may simply adjust stiffness via the ratio k_2/k_3, yet retain some temperature dependence for the purposes of ignition. A parametric study identified the maximum stiffness (k_2/k_3) for which the convection–diffusion–reaction system is stable with explicit time advancement. This value of k_2/k_3 is defined as a stiffness of $\varepsilon = 10^0$. A stiffness of 10^x implies that $\varepsilon^{-1} = k_2/k_3$ is 10^x times larger than its baseline stiffness value. Two levels of heat release were used in the study. The isoenthalpic case assumed that the enthalpy of formation of all the species was identical. The exothermic case assumed H_2,

O_2, N_2 and OH to be identical, but H_2O was adjusted to yield approximately realistic flame temperatures for a hydrogen-air system. Derivatives are evaluated using sixth-order explicit stencils on a grid having 401 points. Approximately 20 grid points are contained within one shock thickness. Spatial boundary conditions for the integration variables were specified using supersonic Euler conditions at the inflow and extrapolation conditions at the outflow. As gradients of flow variables were extremely small at the boundaries, specification error of conditions was deemed negligible.

Initial condition for the hydrodynamic variables, $\{\rho, u, T\}$, are specified by using a precomputed normal-shock profile of air travelling at Mach 5. Each variable is nondimensionalized by its upstream value and integration is performed on the vector $U = \{\rho, \rho u, \rho e_0, \rho Y_i\}$. The nonequilibrium aspect of the initial condition consists of specifying a constant spatial distribution for species mass fractions. Stoichiometric reactant mass fractions are used. Initial values for reaction products are zero. Upon starting the simulation, isoenthalpic or exothermic chemical reactions are abruptly activated. Integrating the reacting shock wave through the spatial domain ten times with this initial species profile rapidly results in a consumption of O_2 and H_2 behind the shock wave, an increase in H_2O behind the shock wave, and a small region of high OH concentration just behind the shock wave. This new profile for all integration variables is the equilibrium profile. Although testing integration methods with nonequilibrium initial conditions may seem somewhat contrived, fluid dynamicists rarely know their initial condition exactly. Oftentimes, the initial condition amounts to an educated guess, particularly inside a flame.

8. Discussion

Numerical tests of the new schemes are conducted on a chemical reaction inducing propagating shock wave and three two-equation singularly-perturbed initial-value problems. The performance of these methods are compared to many existing ARK_2 methods: the (1,2,1), (2,2,2), (2,3,2), (2,3,3), (3,4,3), and (4,4,3) methods of Ascher et al. [5], LIRK3 and LIRK4 due to Calvo et al. [10], a five-stage, 3(2) pair of Fritzen and Wittekindt (FW53) [18], ASIRK–3A from Shen and Zhong [61], the LSSIRK–3A and LSSIRK–4A methods of Yoh and Zhong [76], and ASIRK–3A by Zhong [78] as well as the SBDF methods of Ascher et al. [4]. Tests are conducted to determine stiffness leakage, efficiency, order reduction, error control quality, and dense output performance. A method by Driscoll [16], CRK43, recently appeared and was not numerically tested. No attempt is made to numerically assess conservation properties of any of the methods. Characteristics of the various ARK_2 schemes are listed in Appendix B.

8.1. Stiffness leakage

An essential requirement for the viability of stiff/nonstiff IMEX schemes is that the stiffness remains truely separable. If this were not the case then stiffness would leak out of the stiff terms and stiffen the nonstiff terms. It would manifest itself as a loss in stability or a forced reduction in stepsize of the nonstiff terms. A more expensive fully implicit approach might then be required, and hence, methods that leak substantial stiffness might best be avoided. We test for leakage on the reacting shock wave problem. There are two primary affronts that can be made to the integrator on this problem. The first is simply a very stiff reaction rate describing isoenthalpic or exothermic reactions. Secondly, one could provide an initial condition to the flowfield that is substantially different from the quasi steady-state solution. This nonequilibrium initial condition is accompanied by a strong equilibration process of the flowfield during the initial time steps. Ideally, this initial perturbation is damped during subsequent time steps.

Thirty-seven existing and new IMEX ARK$_2$ and three SBDF schemes are considered for testing. A mild test for stiffness leakage is to provide the integrator with an equilibrium initial condition and an exothermic reaction rate. Time steps are specified as CFL numbers where CFL $= u(\Delta t)/(\Delta x)$ and the upstream velocity is Mach 5. Leakage generally falls into three catagories: insignificant, moderate, and catastrophic. Table 5 shows that for several methods the decrease in stepsize for the nonstiff method is in direct proportion to the increase in reaction rate stiffness. This constitutes, in our opinion, catastrophic leakage and a failure of the methods. They do not possess sufficient stability to be useful in the contexts that they might reasonably be expected to apply. We do not consider these methods further.

A more severe test of leakage is a nonequilibrium initial condition with isoenthalpic reactions. In this case, Table 6 shows that the two methods of Griepentrog [21] can be broken in this rather severe environment. Although much less severe than the leakage displayed in Table 5, both methods of Griepentrog may be inappropriate for stiff computations. More reluctantly than above, we do not further consider these methods. It is interesting to ask why these methods have failed while other methods have not. One may inspect the internal stability function magnitude of the implicit method at stage i for $z^{[I]} \to -\infty$; $R_{\text{int}}^{(i)}(-\infty)$. Unlike SDIRK methods for which $R_{\text{int}}^{(i)}(-\infty) = 0$, ESDIRK methods generally have nonzero values of $R_{\text{int}}^{(i)}(-\infty)$. Table 7 shows that the final stages of both Griepentrog's methods (as well as Driscoll's untested CRK43) are noticably unstable. We would expect that (untested) CRK43 might

Table 5
Examples of catastrophic leakage. Maximum CFL as a function of reaction rate stiffness using equilibrium initial conditions

Method	$\varepsilon = 10^0$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$
Zhong, ASIRK–3A	0.51	0.01	0.0001	0.0
Yoh, SIRK–3A	0.51	0.01	0.0001	0.0
Shen, ASIRK–3A	0.57	0.05	0.0005	0.0
Yoh LSSIRK–3A	0.52	0.05	0.0005	0.0
Yoh LSSIRK–4A	0.36	0.06	0.0006	0.0

Table 6
Maximum CFL as a function of reaction rate stiffness using nonequilibrium initial conditions and a nonexothermic reaction rate

Method	$\varepsilon = 10^0$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-8}$
Griepentrog (3-stage)	0.37	0.37	0.37	0.0007	0.0
Griepentrog (4-stage)	0.44	0.44	0.05	0.001	0.0

Table 7
$R_{\text{int}}^{(i)}(-\infty)$ for the implicit methods of Griepentrog and Driscoll

Stage	1	2	3	4	Step
Griepentrog (3-stage)	+1.00	+0.366	−2.464	–	−0.732
Griepentrog (4-stage)	+1.00	+0.235	+1.068	−4.909	+0.000
Driscoll CRK43	+1.00	−0.500	−1.000	−2.000	+0.000

leak stiffness because it is substantially unstable on its fourth-stage. The IMEX Runge–Kutta methods of Ascher et al. [5], Calvo et al. [10], Fritzen and Wittekindt [18], and the present methods exhibited little to no leakage on either of these problems. If internal stability of the implicit method is a principal contributor to the breakdown of these methods, it is not surprising that the zero-padded A-matrices found in the methods of Ascher et al. [5] and Calvo et al. [10] do not leak stiffness badly because $R_{\text{int}}^{(i)}(-\infty) = 0$ for all stages. The primary concessions of this approach are that the implicit method cannot have a stage-order of two because $a_{21} \neq a_{22}$ and that higher-order embedded methods become problematic.

Our most severe leakage test for the reacting shock wave problem is to use a nonequilibrium initial condition and exothermic chemistry. This initial condition is severe enough to cause stiffness leakage for all of the Runge–Kutta methods when used in a fixed stepsize mode. Table 8 documents the progressive failure of several methods as stiffness is increased. Comparing this to the internal stability characteristics of the implicit methods in Table 9, one may surmise that the ARK5(4)8L[2]SA is failing, in part, because of marginal damping at each stage. It is also interesting that ARK3(2)4L[2]SA and Ascher (2,3,3) wither similarly. One has strong damping at each stage while the other has strong damping at the end of the step. LIRK4, with strong damping at each stage and the step, arguably, appears to endure the best. The

Table 8
Maximum CFL as a function of reaction rate stiffness using nonequilibrium initial conditions and exothermic chemistry

Method	$\varepsilon = 10^0$	$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-5}$	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-7}$
Ascher (2,3,3)	0.38	0.22	0.17	0.16	0.12	0.07	0.01	0.001
ARK3(2)4L[2]SA	0.57	0.28	0.20	0.19	0.14	0.07	0.06	0.001
Calvo LIRK4	0.44	0.44	0.44	0.44	0.40	0.15	0.03	0.008
ARK4(3)6L[2]SA	0.67	0.49	0.36	0.34	0.33	0.16	0.01	0.001
ARK5(4)8L[2]SA	0.43	0.41	0.10	0.05	0.03	0.001	0.00	0.000

Table 9
$R_{\text{int}}^{(i)}(-\infty)$ for IMEX implicit methods

Stage	2	3	4	5	6	7	8	Step
Ascher (2,3,3)	+0.000	+0.000	–	–	–	–	–	−0.732
ARK3(2)4L[2]SA	−1.000	−0.806	0.000	–	–	–	–	+0.000
Calvo LIRK4	+0.000	+0.000	+0.000	+0.000	+0.000	–	–	+0.000
ARK4(3)6L[2]SA	−1.000	−0.774	−0.083	−0.157	+0.000	–	–	+0.000
ARK5(4)8L[2]SA	−1.000	−0.732	−0.649	+0.856	−0.967	−0.353	+0.000	+0.000

Table 10
Maximum CFL of SBDF methods as a function of reaction rate stiffness using nonequilibrium initial conditions and exothermic chemistry

Method	$\varepsilon = 10^0$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$
SBDF2	0.20	0.17	0.10	0.09
SBDF3	0.14	0.13	0.10	0.09
SBDF4	0.11	0.10	0.10	0.09

only IMEX methods that we are aware of that can integrate this problem in constant stepsize mode are the SBDF methods of Ascher et al. [5]. In Table 10, they show insignificant stiffness leakage on our most severe case. Notice that as the order of accuracy increases, the relative leakage decreases. If ARK3(2)4L[2]SA, ARK4(3)6L[2]SA, and ARK5(4)8L[2]SA are used in conjunction with stepsize control, they are able to navigate through the strong initial transient of this problem. Although we cannot say conclusively why the SBDF methods remain stable while all of the IMEX ARK$_2$ methods did not, it seems that both the order and stability matter. Runge–Kutta schemes may be able to satisfy $R_{\mathrm{int}}^{(i)}(-\infty) = 0$ at each stage but always have an overall stage order of one. The SBDF methods are $L(\alpha)$-stable to stiff eigenvalues but each step value is of design order.

In practice, simulations of chemical systems often use exothermic reaction mechanisms and moderately nonequilibrium initial conditions. Once the computation is under way, each step would likely begin with nearly unperturbed initial conditions. For DNS of hydrocarbon flames using a compressible NSE formulation, spatial grid spacings may be of order ten microns at atmospheric pressure [11]. Corresponding convective time step limits based on $z^{[E]} \approx 1$ are of order ten nanoseconds. Under these conditions, detailed methane-air chemical mechanisms do not introduce time scales appreciably faster than those dictated by convective stability, implying $\varepsilon \approx 10^0$. Larger hydrocarbon mechanisms such as heptane-air may introduce timescales of order one femtosecond. Choosing to integrate at the convective limit, the stiffness would be approximately $\varepsilon \approx 10^{-7}$.

8.2. *Accuracy and efficiency*

Beyond avoiding stiffness leakage, one would like accurate and efficient methods. Although our focus is principally on methods of third-order accuracy and higher, we will occasionally use a first-order method, Ascher $(1, 2, 1)$, and a second-order method, Ascher $(2, 3, 2)$, for comparison purposes. These are chosen because their explicit methods have both nonvanishing CFL and viscous CFL numbers. At third-order, the first matter is to verify the accuracy of the methods: ARK3(2)4L[2]SA, Ascher $(2, 3, 3)$, Ascher $(3, 4, 3)$, Ascher $(4, 4, 3)$, Calvo LIRK3, and Fritzen FW53. From Appendix B, all methods are formally third-order. Ascher $(4, 4, 3)$ and Fritzen FW53 use five-stages overall, four of them implicit, while others use one or two less; hence, they might be expected to be relatively less efficient. Using only two implicit stages, Ascher $(2, 3, 3)$ might be expected to be quite efficient. Ascher $(2, 3, 3)$ and Ascher $(3, 4, 3)$ have ARK$_2$ stability functions that depend on $z^{[E]}$. All schemes of Calvo et al. and Ascher et al. have vanishing internal stability functions for the implicit method when $z^{[I]} \to -\infty$. Only ARK3(2)4L[2]SA and Fritzen FW53 have essentially cost-free embedded methods (Calvo LIRK3 and LIRK4 require extra work), only Ascher $(2, 3, 3)$ is not stiffly accurate, and only ARK3(2)4L[2]SA uses a stage-order two implicit method. Fritzen FW53 may not conserve linear first integrals well. Finally, values of γ range from approximately 0.4359 to 1.0.

Accuracy and efficiency tests are conducted using equilibrium initial conditions and exothermic chemistry on the CDR problem. All methods exhibit third-order accuracy in the absence of stiffness where error is given by the L_2 norm, over all grid points, of the difference between the computed and “exact” solution at some final time. The final time corresponds to the movement of the shockwave approximately 100 shock thicknesses. A quasi-exact solution is found by running ARK4(3)6L[2]SA at an order of magnitude finer time step than any used in the grid refinement study. At $\epsilon = 10^{-6}$, error increases in all methods but the order of accuracy remains nearly three. Two seperate measures of efficiency may be considered: accuracy and stability efficiency. Accuracy efficiency determines the work required to obtain

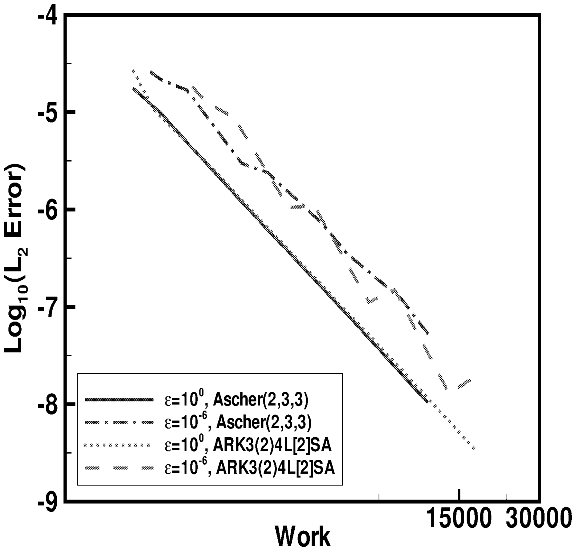

Fig. 2. Error versus work for ARK3(2)4L[2]SA and Ascher $(2, 3, 3)$ in the presence and absence of stiffness.

some chosen error tolerance. We define work as the number of implicit solves required for the integration without regard to Newton iteration count. On error versus work plots, the five stage methods of Ascher $(4, 4, 3)$ and Fritzen FW53 are least efficient. LIRK3 is the least efficient of the four remaining methods on this particular problem, followed by Ascher $(3, 4, 3)$. The most accuracy efficient methods are ARK3(2)4L[2]SA and Ascher $(2, 3, 3)$, shown in Fig. 2. There was no evidence of the coupling stability term causing a problem with Ascher $(2, 3, 3)$. When accuracy is sufficient, one simply seeks the largest stable time step. The limiting time step might be due to ERK linear stability boundary or iterative convergence problems with the Newton's method. In the absence of any stiffness leakage or convergence difficulties, one may compute a theoretical stability based efficiency of the methods by considering the inviscid and viscous CFL numbers normalized by the number of implicit stages. This minimum work point would correspond to the upper left limit of the lines on an error versus work plot. Table 11 compares the maximum time step per unit work for the first- and second-order schemes of Ascher et al. [5], six third-order schemes, two fourth-order schemes, and the one fifth-order method. ARK5(4)8L[2]SA has marginal stability efficiency for both real and imaginary nonstiff eigenvalues. Both LIRK3 and LIRK4 may not be efficient for relatively stiffer convective eigenvalues. In compressible flows, it is likely that the inviscid limit is more relevant to observed stability efficiency. Which of these two limits is more important during stiffness leakage is not clear. Using equilibrium initial conditions and exothermic chemistry, Numerical tests revealed Ascher $(4, 4, 3)$ and Fritzen FW53 were least stability efficient. LIRK3 performed better but not as well as Ascher $(3, 4, 3)$. ARK3(2)4L[2]SA and Ascher $(2, 3, 3)$ were the two most stability efficient. Ascher $(2, 3, 3)$ was found to be slightly more stability efficient than ARK3(2)4L[2]SA at $\varepsilon = 10^{-6}$; however, ARK3(2)4L[2]SA may possess some intrinsic efficiency advantage over Ascher $(2, 3, 3)$ in having a smaller value of γ: 0.4359 versus 0.7887. Modified Newton iteration would presumably converge faster. These experimental results correlate with the viscous stability efficiencies given in table 11 better than the inviscid ones. This might imply that the characteristics of the stiff scaled eigenvalue determine where the ERK stability domain should be fortified.

Table 11
Idealized stability based efficiencies of first- through fifth-order methods in the absence of stiffness leakage

Method	$\mathrm{CFL}_{\mathrm{inviscid}}/s_I$	$\mathrm{CFL}_{\mathrm{viscous}}/s_I$
Ascher (1, 2, 1)	0.435	0.315
Ascher (2, 3, 2)	0.500	0.250
Ascher (2, 3, 3)	0.435	0.315
Ascher (3, 4, 3)	0.473	0.233
Ascher (4, 4, 3)	0.195	0.135
Calvo LIRK3	0.023	0.183
Fritzen FW53	0.218	0.158
ARK3(2)4L[2]SA	0.413	0.307
Calvo LIRK4	0.032	0.176
ARK4(3)6L[2]SA	0.402	0.212
ARK5(4)8L[2]SA	0.061	0.096

At fourth-order and above, for DIRK-based IMEX methods, we are only aware of the LIRK4 method of to Calvo et al. [10] and the methods that we have generated. Calvo LIRK4 was constructed by adding an ERK to the zero-padded SDIRK4 [27]. Both the implicit and additive methods fully damp stiff scaled eigenvalues. Large leading-order error of the ERK dominates the leading-order error of the IMEX method, as shown in Appendix B. The method may conserve quadratic first integrals poorly and has a relatively small inviscid stability limit for the ERK. ARK4(3)6L[2]SA results from an extensive examination of possible approaches to fourth-order methods.

Interestingly, in the presence of stiffness, methods using $C(2, i \neq 2)^{[E]}$ exhibit a smooth reduction in error as work is increased. Those using no simplifying assumption on their explicit method appear rather jagged, e.g., LIRK4 and the five-stage, 4(2) pair that was constructed. One may also compare the degree of error increase with the addition of stiffness. LIRK4 has $A^{(5)} = 0.03919$ and uses no simplifying assumptions whereas one of the test methods has $A^{(5)} = 0.03542$ but makes use of $C(2, i \neq 2)^{[E]}$ and $C(2, i)^{[I]}$. Their nonstiff accuracy efficiencies are quite similar but in the presence of stiffness, LIRK4 shows not only a more dramatic increase in error but more order reduction. Since LIRK4 does not use $C(2, i \neq 2)^{[E]}$ or $C(2, i)^{[I]}$, it is not unreasonable to attribute this difference to the stage-order of the implicit method. Finally, results of various methods on this test problem often correlated with the leading-order error term of the entire method, $A^{(5)}$. Fig. 3 compares Calvo LIRK4 and ARK4(3)6L[2]SA at stiffness extremes showing that ARK4(3)6L[2]SA is not only more accurate but increasingly so as the stiffness is increased.

The only fifth-order method that we are aware of is ARK5(4)8L[2]SA. It has a relatively small linear stability region for its ERK. In the absence of stiffness it is found to be fifth-order, but in the presence of strong stiffness it order reduces in the same manner as ARK4(3)6L[2]SA. Given the behavior of ARK5(4)8L[2]SA on these tests, it is probably the best choice for situations where both mild stiffness and tight error tolerances are present. Finally, comparing efficiency of methods of all orders at $\varepsilon = 10^0$ in Fig. 4 and at $\varepsilon = 10^{-6}$ in Fig. 5, one may conclude that first-order methods are ill-advised. Ascher (2, 3, 2) can provide a much more accurate solution for an identical cost. At any level of stiffness, Ascher (2, 3, 2) is very stability efficient. At tighter error tolerances, ARK4(3)6L[2]SA would appear to be the most

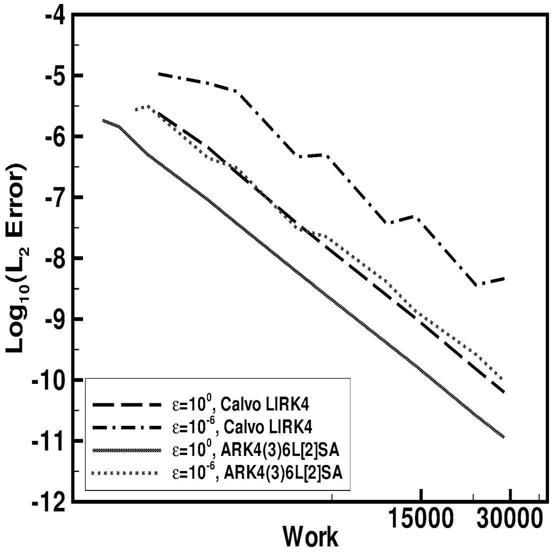

Fig. 3. Error versus work for ARK4(3)6L[2]SA and Calvo LIRK4 in the presence and absence of stiffness.

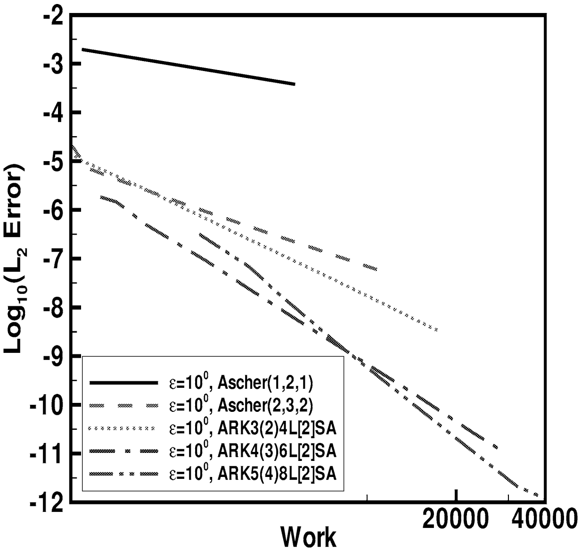

Fig. 4. Error versus work for first- through fifth-order methods in the nonstiff case.

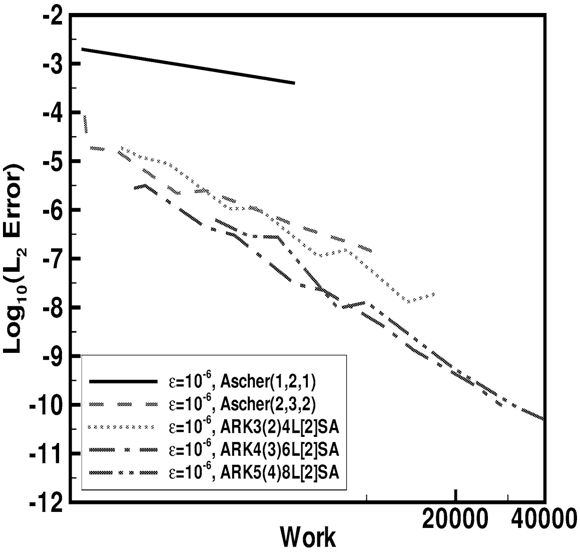

Fig. 5. Error versus work for first- through fifth-order methods in the stiff case.

efficient high-order IMEX additive Runge–Kutta method of which we are aware for this problem. What is somewhat surprising is the marginal improvement in efficiency of higher-order methods at practical tolerances. It likely reflects the fact that ARK_2 order conditions accumulate much faster with increasing order than those of traditional Runge–Kutta methods adversely effecting the leading order error constants.

A limited attempt is made here to quantify the behavior and efficiency of the multistep SBDF schemes as they are the chief alternative to the IMEX Runge–Kutta schemes for coupled integration. Multistep

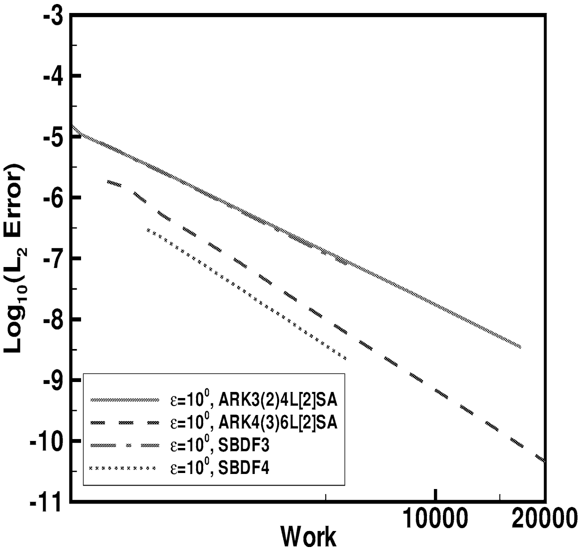

Fig. 6. Efficiency comparison of third- and fourth-order methods with $\varepsilon = 10^0$.

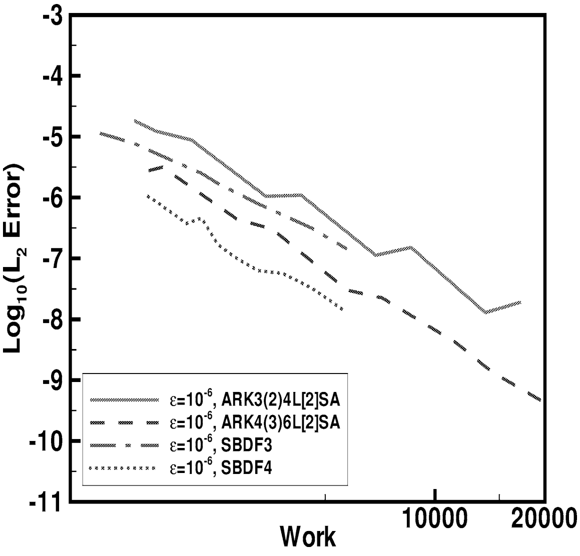

Fig. 7. Efficiency comparison of third- and fourth-order methods with $\varepsilon = 10^{-6}$.

schemes solve only one nonlinear set of equations per time-step. As a result, they can potentially achieve great efficiency compared with multistage Runge–Kutta formulations. Like most ARK$_2$ methods in this study, the SBDF methods lack error control. Additionally, SBDF methods are not self starting, require fixed time steps, and the implicit formulations do not possess A-stability beyond second order accuracy. Second-, third- and fourth-order SBDF schemes of Ascher et al. [5] achieve design accuracy on the reacting shock wave problem independent of the stiffness level, although the leading order error constant appears to depend on stiffness. One may compare the various Runge–Kutta and SBDF methods in terms of either accuracy or stability efficiency. In terms of accuracy, Ascher $(2, 3, 2)$ is substantially more efficient than SBDF2; however, at higher-order, the SBDF methods are more efficient than existing Runge–Kutta methods on this problem. Figs. 6 and 7 compare third- and fourth-order multistep and Runge–Kutta methods with and without stiffness. It should be remembered that chemical reaction rate terms generally give rise to real eigenvalues. The maximum time step for the SBDF formulations is strongly dependent on the location of the scaled stiff eigenvalues, and in particular, whether they fall in the unstable lobes of the implicit BDF3 and BDF4 operators. Scaled eigenvalues on or near the negative real axis are well suited for implicit BDF operators, while eigenvalues near the imaginary axis are not. Conversely, RK schemes degrade in accuracy with increased stiffness due to their lower stage-order.

8.3. *Order-reduction*

Each of the four test problems in this paper are examples of singular perturbation problems. They are ODEs characterized by a stiffness parameter, ε. As ε decreases, the ODE problems gradually transition in behavior toward index-1 DAEs. For Runge–Kutta methods, accompanying this transition is an order-reduction phenomena where the observed convergence rates of the methods fall below the classical order of accuracy. Some differential variables transition to algebraic variables, displaying different convergence rates. Hairer et al. [25,27] determine the convergence rates of SDIRK methods with and without the stiffly accurate assumption. Global error for both differential and algebraic variables are of the form $\epsilon_{\text{global}} = c_1(\Delta t)^\alpha + c_2\varepsilon(\Delta t)^\beta$ for $\varepsilon \leqslant$ Const. (Δt). Independent of stiff accuracy, SDIRK methods have $\alpha = q$ and $\beta = q_{\text{so}} + 1$ for the differential variable, where q and q_{so} are the classical and stage-orders of the method. In the case of algebraic variables, $\alpha = q_{\text{so}} + 1$ and $c_2 = 0$ for nonstiffly accurate methods but $\alpha = q$ and $\beta = q_{\text{so}}$ in the stiffly accurate case. Practical experience shows that the order reduction is problem dependent and may not be as severe as the theoretical estimate.

Although theoretical bounds exist for many implicit Runge–Kutta methods applied to singular perturbation problems, little exists for IMEX methods [74]. No attempt is made to theoretically predict the form of the global error of IMEX ARK_2 methods, rather we shall use the values and form articulated above for SDIRK methods as guidance in empirically estimating the values of the respective exponents. The ultimate goal of this order-reduction study is to establish its severity for CDR problems using stiff chemical kinetic mechanisms. Both the vdP and CDR problems will be used to this end. A cursory examination of the findings from the previous section might lead one to the incorrect conclusion that no order-reduction exists in the CDR problem. Order reduction is more easily identified in simple model problems. Thus, we begin our study with the stiff singular-perturbation model problems. We establish the accuracy of the new methods on these problems and compare them to existing IMEX ARK_2 schemes. Attention is then focused on the reacting shock wave problem where its order-reduction characteristics are demonstrated.

8.3a Order-reduction on model problems

All previously mentioned numerical schemes were run on Kaps' problem, van der Pol's equation, and Pareschi and Russo's problem. In each case, fully implicit and IMEX formulations were compared to assess the effects of partitioning. Order reduction is observed for all ARK_2 schemes whose classical order is greater than two, but is not observed for the SBDF formulations. The general nature of the order reduction is similar for all three problems although the degree of reduction varies between problems. Since van der Pol's equation exhibits the greatest order-reduction, it is chosen as the testbed to compare the accuracy of all schemes. The time interval chosen for these studies is $0 \leqslant t \leqslant 0.5$. Figs. 8 and 9 show representative results of a temporal refinement study at various levels of the stiffness parameter ϵ for van der Pol's equation obtained using ARK4(3)6L[2]SA. From Eq. (47), y_1 is the differential variable, while y_2 transitions from a differential variable to an algebraic variable as the stiffness is increased. The convergence rate of the differential variable is nearly fourth-order, its classical order, and is reasonably smooth. Convergence behavior of the algebraic variable, however, is jagged and departs significantly from the expected design accuracy. This departure from the design accuracy occurs most dramatically for intermediate values of the stiffness parameter ε.

In Figs. 10, 11, and 12, the convergence rates of the ARK3(2)4L[2]SA, ARK4(3)6L[2]SA, and ARK5(4)8L[2]SA methods on van der Pol's equation are plotted versus the stiffness parameter ε. Convergence rates are calculated by a least-squares fit of the data at each ε such as that presented in Figs. 8 and 9. This procedure is at some variance with the fact that error versus stepsize lines are generally composed of two lines of differing slopes. It is adopted, nonetheless, because the dual slope lines are not discernable from the jagged data. ESDIRK method results, for which theoretical estimates of order reduction exist, are included for comparison purposes along with the IMEX values. Certain general trends appeared across each test problem. At values of $\varepsilon \approx 10^0$, the observed convergence rates were

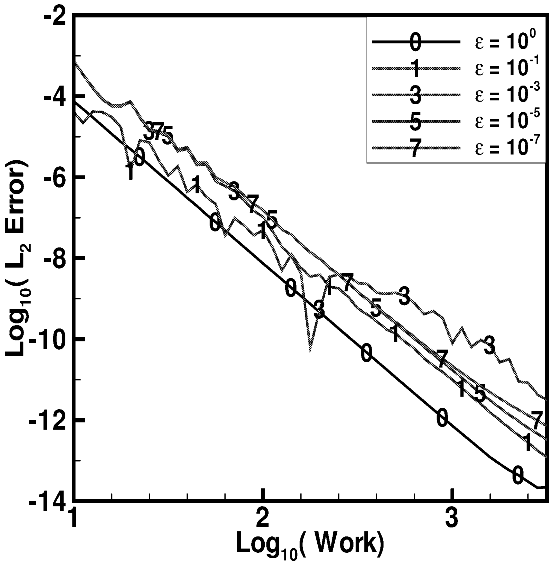

Fig. 8. Nonstiff (differential) error in the van der Pol equation as calculated with the ARK4(3)6L[2]SA scheme.

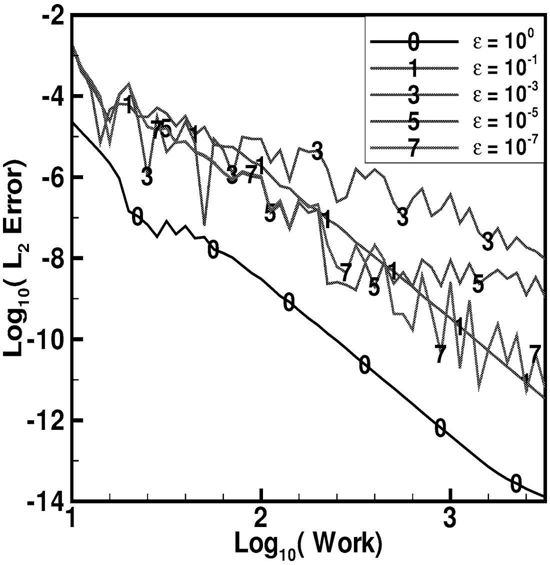

Fig. 9. Stiff (algebraic) error in the van der Pol equation as calculated with the ARK4(3)6L[2]SA scheme.

equal to the classical order for ARK3(2)4L[2]SA and ARK4(3)6L[2]SA methods but somewhat lower for ARK5(4)8L[2]SA on the vdP equation.

All curves show order reduction to some degree for intermediate values of the parameter ε. Both formulations, the ESDIRK alone and the IMEX, order-reduce for the algebraic variables considerably more than for the differential variables. The general trends follow those for the vdP equation presented in Figs. 10, 11, and 12 are summarized in Table 12 along with the Ascher $(2,3,3)$ scheme, Fritzen FW53, and Calvo LIRK3 and LIRK4. Based on results from the three model test problems, we conservatively estimate the leading order truncation terms for each method for the cases where $\varepsilon \leqslant$ Const. (Δt). In practice, this is an awkward and inexact procedure due to the extreme jaggedness of the convergence plots. With this in mind, some methods showed problem dependent convergence behavior. The exact nature of a method's order reduction depends on the relative size of the parameters ε and Δt. The $\varepsilon \ll \Delta t$ limit yields the $\varepsilon(\Delta t)^{\beta}$ contribution to the convergence rate for the method (see Fig. 10 where $\varepsilon = 10^{-8}$). The $\varepsilon \gg \Delta t$ limit gives the classical order for all methods as ε is not a small parameter in this case. Onset of order reduction is observed for the cases where $\varepsilon \approx \Delta t$. By fixing ε and varying Δt in the $\varepsilon \approx \Delta t$ limit, a careful study of order-reduction may be conducted. Typically the convergence rate for these studies has a slope discontinuity, that can be used to identify the leading order error terms. For the IMEX operators, convergence behavior is erratic in this limit, often not monitonically decreasing with time step. Convergence behavior for the third- and fourth-order implicit schemes, presented in Table 12, agrees with the theoretical estimates for SDIRK schemes given above. Incorporating a stage-order of two in the implicit scheme does not translate into higher order-of-accuracy for the differential variables for any of the IMEX schemes, but it increases the convergence rate for the ESDIRK alone. Also, the increased stage order increases the accuracy of the algebraic variable in the fourth- and fifth-order IMEX cases. The 4th-order asymptotic convergence of scheme ARK5(4)8L[2]SA and its ESDIRK makes the overall behavior of the fifth-order scheme very similar to that of the 4th-order scheme. We offer no explanation

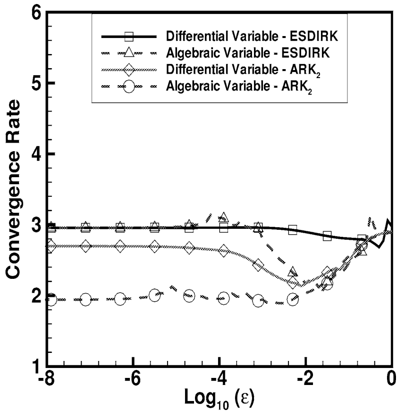

Fig. 10. Convergence rates of ARK3(2)4L[2]SA on the vdP equation as a function of ε.

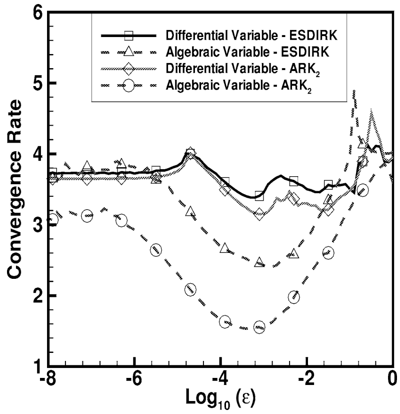

Fig. 11. Convergence rates of ARK4(3)6L[2]SA on the vdP equation as a function of ε.

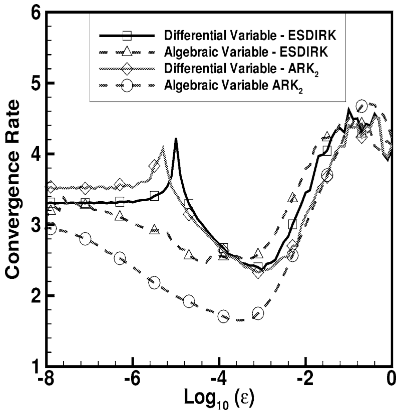

Fig. 12. Convergence rates of ARK5(4)8L[2]SA on the vdP equation as a function of ε.

for this behavior aside from the common observation that van der Pol's equation is a particularly severe test equation. It is possible that this is another manifestation of insufficient internal stability.

8.3b Order-reduction on CDR problems

We now extend the previous study on order reduction to CDR problems using the reaction inducing, propagating shock wave problem. Computations use exothermic chemistry and equilibrium initial conditions. As the reaction rate is introducing the stiffness and only the species continuity equations explicitly contain the reaction rate, we focus on these equations. From Eq. (49), the species equations are given by

$$\frac{\partial(\rho Y_i)}{\partial t} = -\frac{\partial(\rho u Y_i)}{\partial x} + \rho D_i \frac{\partial^2 Y_i}{\partial x^2} + \dot{\omega}_i = F_{\mathrm{ns},i} + \dot{\omega}_i, \tag{51}$$

where $Y_i = \{Y_{\mathrm{H}_2}, Y_{\mathrm{O}_2}, Y_{\mathrm{H}_2\mathrm{O}}, Y_{\mathrm{OH}}, Y_{\mathrm{N}_2}\}$ and $F_{\mathrm{ns},i}$ denotes the sum of convective and diffusive terms for species i. Because of overall continuity, it is not necessary to solve a differential equation for Y_{N_2}. Using

Table 12
Estimated convergence rates for differential and algebraic variables in $\epsilon_{\text{global}} = c_1(\Delta t)^{\alpha} + c_2\varepsilon(\Delta t)^{\beta}$ form using several ESDIRK and IMEX ARK$_2$ methods based on numerical results for $\varepsilon \leqslant$ Const. (Δt).

Method	ESDIRK differential	ESDIRK algebraic	IMEX differential	IMEX algebraic
Ascher (2,3,3)	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$
Calvo LIRK3	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^3 + \varepsilon(\Delta t)^1$	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$
Fritzen FW53	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^3 + \varepsilon(\Delta t)^1$	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^3 + \varepsilon(\Delta t)^1$
ARK3(2)4L[2]SA	$(\Delta t)^3 + \varepsilon(\Delta t)^3$	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$
Calvo LIRK4	$(\Delta t)^4 + \varepsilon(\Delta t)^2$	$(\Delta t)^4 + \varepsilon(\Delta t)^1$	$(\Delta t)^4 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$
ARK4(3)6L[2]SA	$(\Delta t)^4 + \varepsilon(\Delta t)^3$	$(\Delta t)^4 + \varepsilon(\Delta t)^2$	$(\Delta t)^4 + \varepsilon(\Delta t)^2$	$(\Delta t)^3 + \varepsilon(\Delta t)^1$
ARK5(4)8L[2]SA	$(\Delta t)^4 + \varepsilon(\Delta t)^3$	$(\Delta t)^4 + \varepsilon(\Delta t)^2$	$(\Delta t)^4 + \varepsilon(\Delta t)^2$	$(\Delta t)^3 + \varepsilon(\Delta t)^1$

concentrations, $C_i = \rho Y_i / W_i$, and our simplified reaction mechanism, the full species equations appear as

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho Y_{\mathrm{H_2}} \\ \rho Y_{\mathrm{O_2}} \\ \rho Y_{\mathrm{H_2O}} \\ \rho Y_{\mathrm{OH}} \end{bmatrix} = \begin{bmatrix} F_{\mathrm{ns,H_2}} \\ F_{\mathrm{ns,O_2}} \\ F_{\mathrm{ns,H_2}} \\ F_{\mathrm{ns,OH}} \end{bmatrix} + \begin{bmatrix} W_{\mathrm{H_2}}\alpha \exp^{(-T_0/T)^2}\left[\frac{1}{\varepsilon}\left(-C_{\mathrm{H_2}}C_{\mathrm{O_2}} + C_{\mathrm{OH}}^2\right) - C_{\mathrm{H_2}}C_{\mathrm{OH}}^2\right] \\ W_{\mathrm{O_2}}\alpha \exp^{(-T_0/T)^2}\left[\frac{1}{\varepsilon}\left(-C_{\mathrm{H_2}}C_{\mathrm{O_2}} + C_{\mathrm{OH}}^2\right)\right] \\ 2W_{\mathrm{H_2O}}\alpha \exp^{(-T_0/T)^2}\left[C_{\mathrm{H_2}}C_{\mathrm{OH}}^2\right] \\ 2W_{\mathrm{OH}}\alpha \exp^{(-T_0/T)^2}\left[\frac{1}{\varepsilon}\left(-C_{\mathrm{H_2}}C_{\mathrm{O_2}} + C_{\mathrm{OH}}^2\right) - C_{\mathrm{H_2}}C_{\mathrm{OH}}^2\right] \end{bmatrix},$$

where $\varepsilon = k_3/k_2$ and α is some real constant. Fast reactions involve the term $1/\varepsilon$ and are present in the reaction terms for the species H_2, O_2 and OH. Variable O_2 is purely a fast reaction. The other two variables, H_2 and OH, involve both fast and slow reactions, while the variable H_2O is entirely a slow variable. In general, it is difficult to identify slow (differential) and fast (algebraic) variables and hence treat each appropriately. For this simplified reaction system, however, the differential and algebraic variables can readily be identified. Defining the new variables $\xi_1 = W_{\mathrm{O_2}}Y_{\mathrm{H_2}} - W_{\mathrm{H_2}}Y_{\mathrm{O_2}}$ and $\xi_2 = W_{\mathrm{O_2}}Y_{\mathrm{OH}} + 2W_{\mathrm{OH}}Y_{\mathrm{O_2}}$, the system may now be written as

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \xi_1 \\ \rho Y_{\mathrm{O_2}} \\ \rho Y_{\mathrm{H_2O}} \\ \rho \xi_2 \end{bmatrix} = \begin{bmatrix} W_{\mathrm{O_2}}F_{\mathrm{ns,H_2}} - W_{\mathrm{H_2}}F_{\mathrm{ns,O_2}} \\ F_{\mathrm{ns},O_2} \\ F_{\mathrm{ns},H_2} \\ W_{\mathrm{O_2}}F_{\mathrm{ns,OH}} - 2W_{\mathrm{OH}}F_{\mathrm{ns,O_2}} \end{bmatrix} + \begin{bmatrix} W_{\mathrm{H_2}}W_{\mathrm{O_2}}\alpha \exp^{(-T_0/T)^2}\left[-C_{\mathrm{H_2}}C_{\mathrm{OH}}^2\right] \\ W_{O_2}\alpha \exp^{(-T_0/T)^2}\left[\frac{1}{\varepsilon}\left(-C_{\mathrm{H_2}}C_{\mathrm{O_2}} + C_{\mathrm{OH}}^2\right)\right] \\ 2W_{\mathrm{H_2O}}\alpha \exp^{(-T_0/T)^2}\left[C_{\mathrm{H_2}}C_{\mathrm{OH}}^2\right] \\ 2W_{\mathrm{OH}}W_{\mathrm{O_2}}\alpha \exp^{(-T_0/T)^2}\left[-C_{\mathrm{H_2}}C_{\mathrm{OH}}^2\right] \end{bmatrix}.$$

With this new system, it is more clearly seen that at high levels of stiffness, $Y_{\mathrm{O_2}}$ will be an algebraic variable while $\{\xi_1, Y_{\mathrm{H_2O}}, \xi_2\}$ are the differential variables.

Fig. 13 compares the convergence behavior of the three new IMEX ARK$_2$ schemes on the reacting shock wave problem. A representative differential variable, H_2O, and the algebraic variable, O_2, are presented for ARK3(2)4L[2]SA, ARK4(3)6L[2]SA, and ARK5(4)8L[2]SA. Convergence rates are again determined from a least-squares fit of a convergence study at each value of the parameter ε. Intermediate values of the parameter ε, again, produce the most order-reduction in both the differential and algebraic variables.

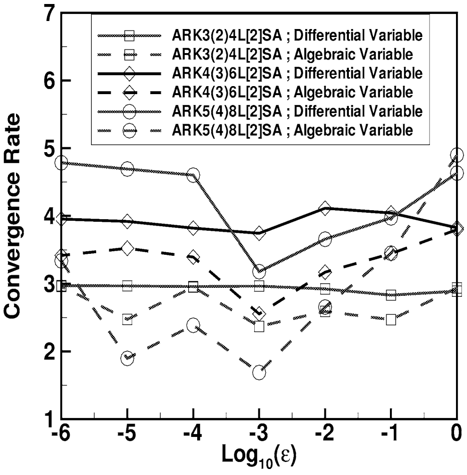

Fig. 13. Convergence rates of differential and algebraic variables on CDR problem using the new ARK_2 methods.

The algebraic variable order-reduction in this CDR problem is remarkably similar to that observed in all three singular perturbation model problems. This degradation in accuracy for the algebraic variable, however, does not dramatically degrade the overall accuracy of the CDR problem. Temperature, which is a combination of differential and algebraic variables, converged at a rate slightly lower than the differential variables. As stiff modes are unnecessary to resolve for accuracy purposes and algebraic variables arise from high stiffness, it may not be surprising that the lower convergence rates of algebraic variables appear to weakly affect temporal error. Another explanation for the benign role of order-reduction in the present CDR problem is that the one algebraic variable may be only weakly coupled to the rest of the system and does not greatly influence the solution accuracy of the other six variables. It is not clear if this may be generalized to all or most reacting flows, however, ARK_2 schemes are likely to experience significant order-reduction on problems where the algebraic component of the error plays a dominant role. In this scenerio, the SBDF schemes, which do not experience order-reduction, are likely to have a clear efficiency advantage over the IMEX Runge–Kutta schemes provided the stiff eigenvalues are predominately real. A case by case study is probably necessary to definitively answer whether order-reduction is an important issue.

8.4. Error control

Choosing a practical error controller for the current IMEX methods is problematic. Advanced controllers designed for explicit and implicit methods are constructed based on different criteria. IMEX schemes, being combinations of each, represent a new challenge for error controllers. Beyond this, with increasing stiffness, controllers additionally confront order-reduction as well as emerging algebraic variables. With this in mind, we test four general appoaches: the I-, PI-, PC, and PID-controllers. The I-controller is appropriate for either implicit or explicit methods. PI- and PID-controllers are advances over I-controllers for explicit methods. PC-controllers have been designed for the unique dynamics of an implicit method.

Van der Pol's equation provides a challenging test for the error control capabilities of the new additive schemes. Over one period, the vdP solution has two temporal boundary layers having thickness related to ϵ^{-1}. To maintain accuracy, the embedded method and controller must sense the layers and adjust the time step accordingly. Significant variations were observed between different controllers. For example, the PC-controller is ineffective. It produced a strong step-change instability characterized by large time-step changes in portions of the temporal cycle where no adjustments were necessary. Controllers designed for implicit methods appear inappropriate. Performance of the simple I-controller is better, yet marginal. Both the PI- and PID-controllers are able to guide the integration through the temporal boundary layer with reasonable efficiency. With better SC-stability properties and similar characteristic roots, the PID-controller behaved best. As the PI- and PID-controllers are designed for the dynamics of explicit methods, it appears that controlling the IMEX method on these problems is largely a function of controlling the explicit method. Further, it appears that the behavior of the controller at the stability boundary is most important. That these controllers worked well is surprising considering that in the presence of large stiffness, the algebraic variables dominate solution accuracy in the three model problems and order reduction is present. The generality of these findings is not clear.

In a second test of the error controllers, the propagating reacting shock wave problem is computed. Testing methods on this problem is rather difficult because the flow contains no transients, yet if one specifies an exothermic reaction system, a nonequilibrium initial condition, and $\varepsilon = 10^{-6}$, a highly transient problem ensues. This same problem is severe enough to break all ARK$_2$ schemes when used in fixed stepsize mode and large stiffness. Conclusions are similar to those drawn from van der Pol's equation. The only controllers capable of guiding the integration out of the flow equilibration phase at all stiffnesses are the PI- and PID-controllers. At low and high stiffness, requested and resultant error are well correlated. Stiffness affects the relation of predicted and actual error but the controller remains useful and is remarkably insensitive to the stiffness even in the case of order reduction. It is less surprising that the explicit-based controllers perform adequately on the CDR problem as the algebraic variables are of secondary importance to solution accuracy.

We do not offer any theoretical explanation why the PI- and PID-controllers work fairly well on these problems. To maintain constant controller gain during order-reduction, p should presumably be reduced. It is not reduced in these tests. Perhaps the essential feature of controlling IMEX methods is coping with scaled eigenvalues at the stability boundary of the explicit method, a task best suited to the PID-controller.

8.5. *Dense output*

The dense output for the three new schemes is tested on the reacting shockwave problem. An equilibrated solution is established at a time, $t = t_{\text{ref}}$ at $\varepsilon = 10^0$, and is used as the initial condition for the study. The initial condition is then advanced one time step to fill all function registers. Interpolation and extrapolation are done at points preceding and following $t_{\text{ref}} + \Delta t$. The dense output is then compared to an "exact" solution obtained with a separate run beginning with the initial condition using $\Delta t/10$ stepsizes and run to the dense output times. A refinement study is performed using a single timestep in the variable Δt to determine the local order of accuracy of the dense output. Note that the nature of the refinement study in the variable Δt returns the local error of the dense output or the global error plus one. Table 13 summarizes the observed logarithms of the local errors, $\delta_{\Delta t}$, and convergence rates from a study using the third-order ($p^* = 3$) formula associated with the ARK4(3)6L[2]SA scheme. The interpolated and extrapolation values are at $\frac{1}{2}(\Delta t)$ and $\frac{3}{2}(\Delta t)$, respectively.

Design order is asymptotically achieved in both modes. Note that the extrapolated data are one and one half orders less accurate than the interpolated data although their respective orders-of accuracy are

Table 13
Convergence rate and local error of the $p^* = 3$ dense output for interpolation and extrapolation as calculated with the ARK4(3)6L[2]SA scheme

Δt	$\delta_{(\Delta t),\mathrm{Int}}$	Order	$\delta_{(\Delta t),\mathrm{Ext}}$	Order
0.9	−5.218		−3.599	
0.6	−5.790	3.241	−4.229	3.579
0.5	−6.070	3.541	−4.525	3.745
0.4	−6.426	3.679	−4.896	3.823
0.3	−6.900	3.791	−5.382	3.891
0.2	−7.583	3.879	−6.077	3.945
0.1	−8.770	3.944	−7.275	3.981

similar. The efficacy of extrapolation decays rapidly with distance. Similar results showing design order dense output were obtained for the ARK3(2)4L[2]SA and ARK5(4)8L[2]SA schemes. A final test of the dense output was performed on both van der Pol's and the CDR equations. Extrapolation mode was used to predict the starting values of the newton iteration from data at the previous timestep. A uniform speedup of the iteration was observed at all levels of stiffness, $\varepsilon = 10^0$ through $\varepsilon = 10^{-6}$, indicating the efficacy of the extrapolation.

9. Conclusions

Additive Runge–Kutta (ARK) methods are investigated for application to the spatially discretized one-dimensional convection–diffusion–reaction (CDR) equations. First, accuracy, stability, conservation, and dense-output are considered for the general case when N different Runge–Kutta methods are grouped into a single composite method. Comparing the $N = 3$ and $N = 2$ cases for CDR applications, $N = 2$ methods are chosen. Then, implicit–explicit, $N = 2$, additive Runge–Kutta (ARK$_2$) methods from third- to fifth-order are presented. Each allows for integration of stiff reactive terms by an L-stable, stiffly-accurate ESDIRK method while the nonstiff convection and diffusion terms are integrated with a traditional ERK method. Coupling error terms are minimized by selecting identical abscissae and scheme weights for each method and are of equal order to those of the elemental methods. Both ARK$_2$ and ESDIRK methods have vanishing stability functions for very large values of the stiff scaled eigenvalue, $z^{[I]} \to -\infty$, and retain high stability efficiency in the absence of stiffness, $z^{[I]} \to 0$. Extrapolation-type stage-value predictors are provided based on dense-output formulae. Dense output stability functions have minimized values for $\theta > 1$ and $z^{[I]} \to -\infty$. Optimized methods minimize both leading order ARK$_2$ error terms and Butcher coefficient magnitudes as well as maximize conservation properties. Numerical tests of the new schemes on a CDR problem show negligible stiffness leakage and near classical order convergence rates. Third- and fourth-order SBDF methods are slightly more efficient than the IMEX ARK$_2$ schemes but do not include error estimation and step-size control. Tests on three simple singular perturbation problems reveal similar and predictable order reduction for the Runge–Kutta methods but no order reduction for the SBDF methods. Order reduction of ARK$_2$ schemes is worst at intermediate stiffness levels. Estimated convergence rates for differential and algebraic variables generally coincide with that predicted by theory. A reinspection of differential and algebraic variables on the CDR problem shows similar behavior. Error control is best managed with a PID-controller, indicating that ERK stability is the overriding issue in controlling time steps and error. Dense output is useful both in interpolation and extrapolation. While results for the fifth-order method are disappointing, both the new third- and fourth-order methods are at least as efficient as existing ARK$_2$ methods.

Acknowledgements

The first author acknowledges support from Sandia National Laboratories, Laboratory Directed Research and Development, and DOE Basic Energy Sciences, Chemical Sciences Division. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94-AL85000. Helpful and much appreciated insight into Runge–Kutta methods has been kindly rendered by Ernst Hairer. Special thanks to the late Dave French for his graphic illustrations, irreverence, and friendship. Discussions with several experts referenced herein is also greatly appreciated.

Appendix A. Runge–Kutta order conditions

Equations of conditions for 1-trees up to fifth-order accuracy as well as the sixth-order tall tree are given by

$$\tau_1^{(1)} = \sum_{i=1}^{s} b_i - \frac{1}{1!}, \qquad \tau_1^{(2)} = \sum_{i=1}^{s} b_i c_i - \frac{1}{2!}, \qquad \tau_1^{(3)} = \frac{1}{2}\sum_{i=1}^{s} b_i c_i^2 - \frac{1}{3!},$$

$$\tau_2^{(3)} = \sum_{i,j=1}^{s} b_i a_{ij} c_j - \frac{1}{3!}, \qquad \tau_1^{(4)} = \frac{1}{6}\sum_{i=1}^{s} b_i c_i^3 - \frac{1}{4!}, \qquad \tau_2^{(4)} = \sum_{i,j=1}^{s} b_i c_i a_{ij} c_j - \frac{3}{4!},$$

$$\tau_3^{(4)} = \frac{1}{2}\sum_{i,j=1}^{s} b_i a_{ij} c_j^2 - \frac{1}{4!}, \qquad \tau_4^{(4)} = \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k - \frac{1}{4!}, \qquad \tau_1^{(5)} = \frac{1}{24}\sum_{i=1}^{s} b_i c_i^4 - \frac{1}{5!},$$

$$\tau_2^{(5)} = \frac{1}{2}\sum_{i,j=1}^{s} b_i c_i^2 a_{ij} c_j - \frac{6}{5!}, \qquad \tau_3^{(5)} = \frac{1}{2}\sum_{i,j,k=1}^{s} b_i a_{ij} c_j a_{ik} c_k - \frac{3}{5!},$$

$$\tau_4^{(5)} = \frac{1}{2}\sum_{i,j=1}^{s} b_i c_i a_{ij} c_j^2 - \frac{4}{5!}, \qquad \tau_5^{(5)} = \frac{1}{6}\sum_{i,j=1}^{s} b_i a_{ij} c_j^3 - \frac{1}{5!}, \qquad \tau_6^{(5)} = \sum_{i,j,k=1}^{s} b_i c_i a_{ij} a_{jk} c_k - \frac{4}{5!},$$

$$\tau_7^{(5)} = \sum_{i,j,k=1}^{s} b_i a_{ij} c_j a_{jk} c_k - \frac{3}{5!}, \qquad \tau_8^{(5)} = \frac{1}{2}\sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k^2 - \frac{1}{5!},$$

$$\tau_9^{(5)} = \sum_{i,j,k,l=1}^{s} b_i a_{ij} a_{jk} a_{kl} c_l - \frac{1}{5!}, \qquad \tau_{20}^{(6)} = \sum_{i,j,k,l,m=1}^{s} b_i a_{ij} a_{jk} a_{kl} a_{lm} c_m - \frac{1}{6!}.$$

General equations of condition for 2-trees up to fifth-order for one root node type are provided in the extended paper [42].

Appendix B. ERK/IRK additive Runge–Kutta methods

Author/Scheme	Year	(s, s_I)	(q^E, q^I, q^C, q)	q^I_{so}	${}^E A(q^E{+}1)$ ${}^E A(q^E{+}2)$	${}^I A(q^I{+}1)$ ${}^I A(q^I{+}2)$	${}^C A(q^C{+}1)$ ${}^C A(q^C{+}2)$	$A^{(q+1)}$ $A^{(q+2)}$	A-Stable $a^{[I]}_{sj} = b^{[I]}_j$	$R(-\infty)$ (λ, λ_v)	$b^{[E]}_i = b^{[I]}_i$ $c^{[E]}_i = c^{[I]}_i$	$\lvert M^{[E,I]}\rvert$	$\mathrm{Max}\{a^{[I]}_{ii}\}$ D
ARK5(4)8L[2]SA	2001	(8, 7)	(5, 5, 5, 5)	2	0.002945 0.008705	0.001680 0.002770	0.006110 0.01229	0.006988 0.01531	yes yes	0.00 (0.43, 0.67)	yes yes	13.44	0.2050 14.69
ARK4(3)6L[2]SA	2001	(6, 5)	(4, 4, 4, 4)	2	0.004470 0.007414	0.003401 0.005405	0.01087 0.07397	0.01224 0.07454	yes yes	0.00 (2.01, 1.06)	yes yes	0.6684	0.2500 1.059
Calvo LIRK4	2000	(6, 5)	(4, 4, 4, 4)	1	0.03012 0.03031	0.002504 0.004511	0.02494 0.07619	0.03919 0.08213	yes yes	0.00 (0.16, 0.88)	yes yes	112.6	0.2500 14.17
ARK3(2)4L[2]SA	2001	(4, 3)	(3, 3, 3, 3)	2	0.02236 0.02373	0.03663 0.07870	0.05802 0.09737	0.07217 0.1274	yes yes	0.00 (1.24, 0.92)	yes yes	1.203	0.4359 1.038
Driscoll CRK43	2001	(4, 3)	(4, 3, 3, 3)	1	0.01450 0.01604	0.04606 0.08812	0.09821 0.1617	0.1085 0.1847	yes no	0.00 (1.42, 0.70)	yes yes	0.1969	1.000 1.000
Calvo LIRK3	2000	(4, 3)	(3, 3, 3, 3)	1	0.05691 0.07123	0.02970 0.06535	0.07704 0.1497	0.1003 0.1767	yes yes	0.00 (0.07, 0.55)	yes yes	1.599	0.4359 1.989
Ascher (2,3,3)	1997	(3, 2)	(3, 3, 3, 3)	1	0.1019 0.06695	0.1270 0.2443	0.1270 0.2171	0.2064 0.3336	yes no	$-0.732(1 + z^{[E]})$ (0.87, 0.63)	yes yes	0.2887	0.7887 0.7887
Ascher (3,4,3)	1997	(4, 3)	(3, 3, 3, 3)	1	0.07595 0.08921	0.02970 0.06535	0.06311 0.1248	0.1031 0.1667	yes yes	$0.106 z^{[E]}$ (1.42, 0.70)	yes yes	1.407	0.4359 1.208
Ascher (4,4,3)	1997	(5, 4)	(3, 3, 3, 3)	1	0.09053 0.08623	0.03165 0.06994	0.1318 0.2437	0.1630 0.2678	yes yes	0.00 (0.78, 0.54)	no yes	2.969	0.5000 1.750
Fritzen	1997	(5, 4)	(3, 3, 3, 3)	1	0.07217 0.05649	0.07082 0.1512	0.1707 0.3238	0.1984 0.3618	yes yes	0.00 (0.87, 0.63)	no yes	2.634×10^{-9}	1.0000 1.000
Griepentrog	1978	(3, 2)	(3, 3, 3, 3)	1	0.05893 0.08038	0.1312 0.2910	0.1270 0.2869	0.1918 0.4165	yes no	$-0.732(1 + z^{[E]})$ (0.87, 0.63)	yes yes	0.7571	0.7887 2.943
Griepentrog	1978	(4, 3)	(4, 3, 3, 3)	1	0.01267 0.01444	0.07426 0.1481	0.06926 0.09834	0.1015 0.1782	yes no	$-0.123 z^{[E]}$ (1.42, 0.70)	yes yes	0.6047	0.4359 3.615
Yoh SIRK-3A	1998	(3, 3)	(3, 3, 2, 2)	1	0.08184 0.06176	0.09220 0.1791	0.3853 0.6189	0.3853 0.6310	yes no	0.00 (0.87, 0.63)	yes no	0.3131	0.7500 1.179
Yoh LSSIRK-4A	1998	(4, 4)	(3, 3, 2, 2)	1	0.1319 0.1169	2.341 22.46	0.3770 4.617	0.3770 5.178	no no	−0.456 (0.71, 0.46)	yes no	2.926	5.656 5.656
Yoh LSSIRK-3A	1997	(3, 3)	(3, 3, 2, 2)	1	0.04398 0.04551	0.007427 0.008368	0.1039 0.1683	0.1039 0.1741	no no	2.45 (0.87, 0.63)	yes no	0.2940	0.6533 0.9375
Ascher (1,2,2)	1997	(2, 1)	(2, 2, 3, 2)	1	0.1718 0.1398	0.09317 0.08839	0.05893 0.08079	0.1954 0.1755	yes no	−1.0000 (0.00, 0.50)	yes yes	0.5000	0.5000 1.000
Ascher (2,2,2)	1997	(3, 2)	(2, 2, 2, 2)	1	0.1911 0.1424	0.04168 0.05790	0.02860 0.1150	0.1976 0.1920	yes yes	$-0.7071 z^{[E]}$ (0.00, 0.50)	no yes	0.7071	0.2929 1.707
Ascher (2,3,2)	1997	(3, 2)	(2, 2, 3, 2)	1	0.01011 0.06223	0.04168 0.05790	0.0 0.1011	0.04289 0.08560	yes yes	$0.943 z^{[E]}$ (0.87, 0.63)	yes yes	0.2929	0.2929 1.000
Zhong ASIRK-2A	1996	(2, 2)	(2, 2, 2, 2)	1	0.1863 0.1443	0.04295 0.04443	0.05893 0.2045	0.2001 0.2542	yes no	0.00 (0.00, 0.50)	yes no	0.2041	0.3333 1.000
Zhong ASIRK-3A	1996	(3, 3)	(3, 3, 2, 2)	1	0.08184 0.06176	0.1110 0.2423	0.2676 0.4289	0.2676 0.4506	yes no	0.00 (0.87, 0.63)	yes no	0.4716	0.9511 1.258
Shen ASIRK-3A	1996	(4, 4)	(3, 3, 2, 2)	1	0.03792 0.04231	0.1452 0.3471	0.1078 0.4338	0.1078 0.5572	yes no	0.00 (0.92, 0.69)	yes no	0.2871	1.175 1.781

Appendix C. Additive Runge–Kutta scheme coefficients

ARK3(2)4L[2]SA–ERK

$$
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\frac{1767732205903}{2027836641118} & \frac{1767732205903}{2027836641118} & 0 & 0 & 0 \\
\frac{3}{5} & \frac{5535828885825}{10492691773637} & \frac{788022342437}{10882634858940} & 0 & 0 \\
1 & \frac{6485989280629}{16251701735622} & \frac{-4246266847089}{9704473918619} & \frac{10755448449292}{10357097424841} & 0 \\
\hline
b_i & \frac{1471266399579}{7840856788654} & \frac{-4482444167858}{7529755066697} & \frac{11266239266428}{11593286722821} & \frac{1767732205903}{4055673282236} \\
\hline
\hat{b}_i & \frac{2756255671327}{12835298489170} & \frac{-10771552573575}{22201958757719} & \frac{9247589265047}{10645013368117} & \frac{2193209047091}{5459859503100}
\end{array}
$$

ARK3(2)4L[2]SA–ESDIRK

$$
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\frac{1767732205903}{2027836641118} & \frac{1767732205903}{4055673282236} & \frac{1767732205903}{4055673282236} & 0 & 0 \\
\frac{3}{5} & \frac{2746238789719}{10658868560708} & \frac{-640167445237}{6845629431997} & \frac{1767732205903}{4055673282236} & 0 \\
1 & \frac{1471266399579}{7840856788654} & \frac{-4482444167858}{7529755066697} & \frac{11266239266428}{11593286722821} & \frac{1767732205903}{4055673282236} \\
\hline
b_i & \frac{1471266399579}{7840856788654} & \frac{-4482444167858}{7529755066697} & \frac{11266239266428}{11593286722821} & \frac{1767732205903}{4055673282236} \\
\hline
\hat{b}_i & \frac{2756255671327}{12835298489170} & \frac{-10771552573575}{22201958757719} & \frac{9247589265047}{10645013368117} & \frac{2193209047091}{5459859503100}
\end{array}
$$

ARK3(2)4L[2]SA–second-order dense output

b^*_{ij}	$i=1$	$i=2$	$i=3$	$i=4$
$j=1$	$\frac{+4655552711362}{22874653954995}$	$\frac{-18682724506714}{9892148508045}$	$\frac{34259539580243}{13192909600954}$	$\frac{584795268549}{6622622206610}$
$j=2$	$\frac{-215264564351}{13552729205753}$	$\frac{17870216137069}{13817060693119}$	$\frac{-28141676662227}{17317692491321}$	$\frac{2508943948391}{7218656332882}$

ARK4(3)6L[2]SA–ERK

0	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0
$\frac{83}{250}$	$\frac{13861}{62500}$	$\frac{6889}{62500}$	0	0	0	0
$\frac{31}{50}$	$\frac{-116923316275}{2393684061468}$	$\frac{-2731218467317}{15368042101831}$	$\frac{9408046702089}{11113171139209}$	0	0	0
$\frac{17}{20}$	$\frac{-451086348788}{2902428689909}$	$\frac{-2682348792572}{7519795681897}$	$\frac{12662868775082}{11960479115383}$	$\frac{3355817975965}{11060851509271}$	0	0
1	$\frac{647845179188}{3216320057751}$	$\frac{73281519250}{8382639484533}$	$\frac{552539513391}{3454668386233}$	$\frac{3354512671639}{8306763924573}$	$\frac{4040}{17871}$	0
b_i	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$\frac{-2260}{8211}$	$\frac{1}{4}$
$\hat{b}_i$	$\frac{4586570599}{29645900160}$	0	$\frac{178811875}{945068544}$	$\frac{814220225}{1159782912}$	$\frac{-3700637}{11593932}$	$\frac{61727}{225920}$

ARK4(3)6L[2]SA–ESDIRK

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

ARK4(3)6L[2]SA–second-order dense output

b^*_{ij}	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$
$j=1$	$\frac{5701579834848}{6164663940925}$	0	$\frac{13131138058924}{17779730471019}$	$\frac{-28096677048929}{11161768239540}$	$\frac{42062433452849}{11720557422164}$	$\frac{-25841894007917}{14894670528776}$
$j=2$	$\frac{-7364557999481}{9602213853517}$	0	$\frac{-6355522249597}{11518083130066}$	$\frac{29755736407445}{9305094404071}$	$\frac{-38886896333129}{10063858340160}$	$\frac{22142945955077}{11155272088250}$

ARK5(4)8L[2]SA–ERK

0	0	0	0	0	0	0	0	0
$\frac{41}{100}$	$\frac{41}{100}$	0	0	0	0	0	0	0
$\frac{2935347310677}{11292855782101}$	$\frac{367902744464}{2072280473677}$	$\frac{677623207551}{8224143866563}$	0	0	0	0	0	0
$\frac{1426016391358}{7196633302097}$	$\frac{1268023523408}{10340822734521}$	0	$\frac{1029933939417}{13636558850479}$	0	0	0	0	0
$\frac{92}{100}$	$\frac{14463281900351}{6315353703477}$	0	$\frac{66114435211212}{5879490589093}$	$\frac{-54053170152839}{4284798021562}$	0	0	0	0
$\frac{24}{100}$	$\frac{14090043504691}{34967701212078}$	0	$\frac{15191511035443}{11219624916014}$	$\frac{-18461159152457}{12425892160975}$	$\frac{-281667163811}{9011619295870}$	0	0	0
$\frac{3}{5}$	$\frac{19230459214898}{13134317526959}$	0	$\frac{21275331358303}{2942455364971}$	$\frac{-38145345988419}{4862620318723}$	$\frac{-1}{8}$	$\frac{-1}{8}$	0	0
1	$\frac{-19977161125411}{11928030595625}$	0	$\frac{-40795976796054}{6384907823539}$	$\frac{177454434618887}{12078138498510}$	$\frac{782672205425}{8267701900261}$	$\frac{-69563011059811}{9646580694205}$	$\frac{7356628210526}{4942186776405}$	0
b_i	$\frac{-872700587467}{9133579230613}$	0	0	$\frac{22348218063261}{9555858737531}$	$\frac{-1143369518992}{8141816002931}$	$\frac{-39379526789629}{19018526304540}$	$\frac{32727382324388}{42900044865799}$	$\frac{41}{200}$
$\hat{b}_i$	$\frac{-975461918565}{9796059967033}$	0	0	$\frac{78070527104295}{32432590147079}$	$\frac{-548382580838}{3424219808633}$	$\frac{-33438840321285}{15594753105479}$	$\frac{3629800801594}{4656183773603}$	$\frac{4035322873751}{18575991585200}$

ARK5(4)8L[2]SA–ESDIRK

0	0	0	0	0	0	0	0	0
$\frac{41}{100}$	$\frac{41}{200}$	$\frac{41}{200}$	0	0	0	0	0	0
$\frac{2935347310677}{11292855782101}$	$\frac{41}{400}$	$\frac{-567603406766}{11931857230679}$	$\frac{41}{200}$	0	0	0	0	0
$\frac{1426016391358}{7196633302097}$	$\frac{683785636431}{9252920307686}$	0	$\frac{-110385047103}{1367015193373}$	$\frac{41}{200}$	0	0	0	0
$\frac{92}{100}$	$\frac{3016520224154}{10081342136671}$	0	$\frac{30586259806659}{12414158314087}$	$\frac{-22760509404356}{11113319521817}$	$\frac{41}{200}$	0	0	0
$\frac{24}{100}$	$\frac{218866479029}{1489978393911}$	0	$\frac{638256894668}{5436446318841}$	$\frac{-1179710474555}{5321154724896}$	$\frac{-60928119172}{8023461067671}$	$\frac{41}{200}$	0	0
$\frac{3}{5}$	$\frac{1020004230633}{5715676835656}$	0	$\frac{25762820946817}{25263940353407}$	$\frac{-2161375909145}{9755907335909}$	$\frac{-211217309593}{5846859502534}$	$\frac{-4269925059573}{7827059040749}$	$\frac{41}{200}$	0
1	$\frac{-872700587467}{9133579230613}$	0	0	$\frac{22348218063261}{9555858737531}$	$\frac{-1143369518992}{8141816002931}$	$\frac{-39379526789629}{19018526304540}$	$\frac{32727382324388}{42900044865799}$	$\frac{41}{200}$
b_i	$\frac{-872700587467}{9133579230613}$	0	0	$\frac{22348218063261}{9555858737531}$	$\frac{-1143369518992}{8141816002931}$	$\frac{-39379526789629}{19018526304540}$	$\frac{32727382324388}{42900044865799}$	$\frac{41}{200}$
$\hat{b}_i$	$\frac{-975461918565}{9796059967033}$	0	0	$\frac{78070527104295}{32432590147079}$	$\frac{-548382580838}{3424219808633}$	$\frac{-33438840321285}{15594753105479}$	$\frac{3629800801594}{4656183773603}$	$\frac{4035322873751}{18575991585200}$

ARK4(3)6L[2]SA–third-order dense output

b^*_{ij}	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$
$j=1$	$\frac{6943876665148}{7220017795957}$	0	$\frac{7640104374378}{9702883013639}$	$\frac{-20649996744609}{7521556579894}$	$\frac{8854892464581}{2390941311638}$	$\frac{-11397109935349}{6675773540249}$
$j=2$	$\frac{-54480133}{30881146}$	0	$\frac{-11436875}{14766696}$	$\frac{174696575}{18121608}$	$\frac{-12120380}{966161}$	$\frac{3843}{706}$
$j=3$	$\frac{6818779379841}{7100303317025}$	0	$\frac{2173542590792}{12501825683035}$	$\frac{-31592104683404}{5083833661969}$	$\frac{61146701046299}{7138195549469}$	$\frac{-17219254887155}{4939391667607}$

ARK5(4)8L[2]SA–third-order dense output

b^*_{ij}	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$i=7$	$i=8$
$j=1$	$\frac{-17674230611817}{10670229744614}$	0	0	$\frac{65168852399939}{7868540260826}$	$\frac{15494834004392}{5936557850923}$	$\frac{-99329723586156}{26959484932159}$	$\frac{-19024464361622}{5461577185407}$	$\frac{-6511271360970}{6095937251113}$
$j=2$	$\frac{43486358583215}{12773830924787}$	0	0	$\frac{-91478233927265}{11067650958493}$	$\frac{-79368583304911}{10890268929626}$	$\frac{-12239297817655}{9152339842473}$	$\frac{115839755401235}{10719374521269}$	$\frac{5843115559534}{2180450260947}$
$j=3$	$\frac{-9257016797708}{5021505065439}$	0	0	$\frac{26096422576131}{11239449250142}$	$\frac{92396832856987}{20362823103730}$	$\frac{30029262896817}{10175596800299}$	$\frac{-26136350496073}{3983972220547}$	$\frac{-5289405421727}{3760307252460}$

References

[1] R. Alexander, Diagonally implicit Runge–Kutta methods for stiff O.D.E.s, SIAM J. Numer. Anal. 14 (6) (1977) 1006–1021.

[2] R. Alexander, J.J. Coyle, Runge–Kutta methods and differential–algebraic systems, SIAM J. Numer. Anal. 27 (3) (1990) 736–752.

[3] A.L. Araújo, A. Murua, J.M. Sanz-Serna, Symplectic methods based on decompositions, SIAM J. Numer. Anal. 34 (5) (1997) 1926–1947.

[4] U.M. Ascher, S.J. Ruuth, B.T.R. Wetton, Implicit–explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (3) (1995) 797–823.

[5] U.M. Ascher, S.J. Ruuth, R.J. Spiteri, Implicit–explicit Runge–Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (2–3) (1997) 151–167.

[6] A. Bellen, M. Zennaro, Stability properties of interpolants for Runge–Kutta methods, SIAM J. Numer. Anal. 25 (2) (1988) 411–432.

[7] T. Belytschko, A review of recent developments in time integration, in: A.K. Noor, J.T. Oden (Eds.), State-of-the-Art Surveys on Computational Mechanics, American Society of Mechanical Engineers, New York, 1989, pp. 185–199.

[8] J.C. Butcher, The Numerical Analysis of Ordinary Differential Equations: Runge–Kutta and General Linear Methods, Wiley, Chichester, 1987.

[9] M.P. Calvo, A. Iserles, A. Zanna, Numerical solution of isospectral flows, Math. Comput. 66 (220) (1997) 1461–1486.

[10] M.P. Calvo, J. de Frutos, J. Novo, Linearly implicit Runge–Kutta methods for advection–reaction–diffusion equations, Appl. Numer. Math. 37 (4) (2001) 535–549.

[11] J.H. Chen, H. G Im, Correlation of the flame speed with stretch in turbulent premixed methane/air flames, in: Twenty-Seventh Symposium (International) on Combustion, The Combustion Institute, Pittsburg, PA, 1998, pp. 819–826.

[12] L.R. Chen, D.G. Liu, Combined RK-Rosenbrock methods and their stability, Math. Numer. Sinica 22 (3) (2000) 319–332 (in Chinese), ISSN 0254-7791.

[13] L.R. Chen, D.G. Liu, Parallel compound methods for solving partitioned stiff systems, J. Comput. Math. 19 (6) (2001) 639–650.

[14] G.J. Cooper, A.M.S. Sayfy, Additive Runge–Kutta methods for stiff ordinary differential equations, Math. Comput. 40 (161) (1983) 207–218.

[15] K. Dekker, J.G. Verwer, Stability of Runge–Kutta Methods for Stiff Nonlinear Differential Equations, North-Holland, Amsterdam, 1984.

[16] T.A. Driscoll, A composite Runge–Kutta method for the spectral solution of semilinear PDE (2001), unpublished; See http://www.math.udel.edu/~driscoll/research/.

[17] J. Frank, W. Hundsdorfer, J.G. Verwer, Stability of implicit–explicit linear multstep methods, Appl. Numer. Math. 25 (2–3) (1997) 193–205.

[18] P. Fritzen, J. Wittekindt, Numerical solution of viscoplastic constitutive equations with internal state variables, Part I: Algorithms and implementation, Math. Meth. Appl. Sci. 20 (16) (1997) 1411–1425.

[19] A. Gerische, J.G. Verwer, Operator splitting and approximate factorization for taxis–diffusion–reaction models, Appl. Numer. Math. 42 (2002) 159–176.

[20] S. González-Pinto, J.I. Montijano, S. Pérez-Rodríguez, On starting algorithms for implicit RK methods, BIT 40 (4) (2000) 685–714.

[21] E. Griepentrog, Gemischte Runge–Kutta-Verfahren für steife system (Mixed Runge–Kutta methods for stiff systems), in: Seminarberichte Nr. 11, 19–29, Sektion Mathematik, Humboldt Universität, Berlin, 1978 (in German).

[22] K. Gustafsson, M. Lundh, G. Söderlind, A PI-stepsize control for the numerical solution of ordinary differential equations, BIT 28 (2) (1988) 270–287.

[23] K. Gustafsson, G. Söderlind, Control theoretic techniques for the iterative solution of nonlinear equation in ODE solvers, SIAM J. Sci. Comput. 18 (1) (1997) 23–40.

[24] E. Hairer, Order conditions for numerical methods for partitioned ordinary differential equations, Numer. Math. 36 (4) (1981) 431–445.

[25] E. Hairer, Ch. Lubich, M. Roche, Error of Runge–Kutta methods for stiff problems studied via differential algebraic equations, BIT 28 (3) (1988) 678–700.

[26] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Nonstiff Problems, 2nd Edition, Springer-Verlag, Berlin, 1993.

[27] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Stiff and Differential–Algebraic Problems, 2nd Edition, Springer-Verlag, Berlin, 1996.

[28] E. Hairer, Numerical Geometric Integration, Unpublished Notes, Section de Mathématiques, Université de Genève, Genève, 1999.

[29] I. Higueras, T. Roldán, Starting algorithms for some DIRK methods, Numer. Algorithms 23 (4) (2000) 357–369.

[30] A.C. Hindmarsh, S.P. Nørsett, KRYSI, An ODE Solver combining a semi-implicit Runge–Kutta method with a preconditioned Krylov method, UCID-21422, Lawrence Livermore National Laboratories, Livermore, CA, 1988.

[31] E. Hofer, A partially implicit method for large stiff systems of ODEs with only few equations introducing small time-constants, SIAM J. Numer. Anal. 13 (5) (1976) 645–663.

[32] M.E. Hosea, L.F. Shampine, Analysis and implementation of TR–BDF2, Appl. Numer. Math. 20 (1) (1996) 21–37.

[33] N. Houbak, S.P. Norsett, P.G. Thomsen, Displacement or residual test in the application of implicit methods for stiff problems, IMA J. Numer. Anal. 5 (3) (1985) 297–305.

[34] W.H. Hundsdorfer, Numerical solution of advection–diffusion–reaction equations, CWI Report NM-N9603, Centrum voor Wiskunde en Informatica, Amsterdam, 1996.

[35] K.J. in 't Hout, On the contractivity of implicit–explicit linear multistep methods, Appl. Numer. Math. 42 (2002) 201–212.

[36] A. Iserles, A. Zanna, Preserving algebraic invariants with Runge–Kutta methods, J. Comput. Appl. Math. 125 (1) (2000) 69–81.

[37] Z. Jackiewicz, R. Vermiglio, Order conditions for partitioned Runge–Kutta methods, Appl. Math. 45 (4) (2000) 301–316.

[38] L. Jay, Convergence of a class of Runge–Kutta methods for differential–algebraic systems of index-2, BIT 33 (1) (1993) 137–150.

[39] J.C. Jorge, J.I. Montijano, L. Randez, A pair 4 (3) of additive Runge–Kutta methods for the integration of stiff differential equations, in: Actas del XI C.E.D.Y.A. (Congreso de Ecuaciones Diferenciales y Aplicaciones): I Congreso de Matematica Aplicada, 1989, pp. 295–299 (in Spanish).

[40] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, PA, 1995.

[41] C.A. Kennedy, M.H. Carpenter, R.H. Lewis, Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations, Appl. Numer. Math. 35 (3) (2000) 177–219.

[42] C.A. Kennedy, M.H. Carpenter, Additive Runge–Kutta schemes for convection–diffusion–reaction equations, NASA Technical Memorandum, NASA/TM-2001-211038, Langley Research Center, Hampton, VA, 2001.

[43] A. Kværnø, More, and to be hoped for, better DIRK methods for the solution of stiff ODEs, Technical Report, Mathematical Sciences Division, Norwegian Institute of Technology, Trondheim, Norway (1992).

[44] A. Kværnø, S.P. Nørsett, B. Owren, Runge–Kutta research in Trondheim, Appl. Numer. Math. 22 (1–3) (1996) 263–277.

[45] M.P. Laburta, Starting algorithms for IRK methods, J. Comput. Appl. Math. 83 (2) (1997) 269–288.

[46] J.D. Lambert, Numerical Methods for Ordinary Differential Systems. The Initial Value Problem, Wiley, Chichester, 1991.

[47] D. Lanser, J.G. Verwer, Analysis of operator splitting for advection–diffusion–reaction problems from air pollution modelling, J. Comput. Appl. Math. 111 (1) (1999) 201–216.

[48] A. Murua, Formal series and numerical integrators, Part I: Systems of ODEs and symplectic integrators, Appl. Numer. Math. 29 (2) (1999) 221–251.

[49] S.P. Nørsett, P.G. Thomsen, Local error control in SDIRK-methods, BIT 26 (1) (1986) 100–113.

[50] B. Owren, M. Zennaro, Order barriers for continuous explicit Runge–Kutta methods, Math. Comp. 56 (194) (1991) 645–661.

[51] L. Pareschi, G. Russo, Implicit–explicit Runge–Kutta schemes for stiff systems of differential equations, in: L. Brugnano, D. Trigiante (Eds.), Recent Trends in Numerical Analysis, Vol. 3, Nova Science, Huntington, NY, 2000, pp. 269–289.

[52] P. Rentrop, Partitioned Runge–Kutta methods with stiffness detection and stepsize control, Numer. Math. 47 (4) (1985) 545–564.

[53] W.C. Rheinboldt, Methods for Solving Nonlinear Equations, 2nd Edition, SIAM, Philadelphia, PA, 1998.

[54] S.J. Ruuth, Implicit–explicit methods for reaction–diffusion problems in pattern formation, J. Math. Biology 34 (2) (1995) 148–176.

[55] J. Sand, RK-predictors: Extrapolation methods for implicit Runge–Kutta formulae, DIKU Report Nr. 88/18, Datalogisk Institut, Københavns Universitet, Copenhagen, 1988.

[56] J. Sand, Methods for starting iteration schemes for implicit Runge–Kutta formulae, in: J.R. Cash, I. Gladwell (Eds.), Computational Ordinary Differential Equations, Clarendon Press, New York, 1992, pp. 115–126.

[57] A.M.S. Sayfy, A[α-]stable additive Runge–Kutta methods for stiff I.V.P.'s, J. Info. Opt. Sci. 16 (3) (1995) 471–480.

[58] L.F. Shampine, Implementation of implicit formulas for the solution of ODEs, SIAM J. Sci. Comput. 1 (1) (1980) 103–118.

[59] L.F. Shampine, L.S. Baca, Error estimators for stiff differential equations, J. Comput. Appl. Math. 11 (2) (1984) 197–207.

[60] L.F. Shampine, Numerical Solution of Ordinary Differential Equations, Chapman and Hall, New York, 1994.

[61] J.W. Shen, X. Zhong, Semi-implicit Runge–Kutta schemes for non-autonomous differential equations in reactive flow computations, AIAA Paper 96-1969, AIAA, Fluid Dynamics Conference, 27th, New Orleans, LA, June 17–20, 1996.

[62] N.J.A. Sloane, S. Plouffe, The Encyclopedia of Integer Sequences, Academic Press, San Diego, 1995, Also, see http://www.research.att.com/~njas/sequences/index.html.

[63] G. Söderlind, The automatic control of numerical integration, CWI Quarterly 11 (1) (1998) 55–74.

[64] G. Söderlind, Digital filters in adaptive time-stepping Part I: PID control, ACM Trans. Math. Software, accepted for publication.

[65] M.N. Spijker, Feasibility and contractivity in implicit Runge–Kutta methods, J. Comput. Appl. Math. 12/13 (1985) 563–578.

[66] M.N. Spijker, On the error committed by stopping the Newton iteration in implicit Runge–Kutta methods, Ann. Numer. Math. 1 (1994) 199–212.

[67] Ch. Tsitouras, S.N. Papakostas, Cheap error estimation for Runge–Kutta methods, SIAM J. Sci. Comput. 20 (6) (1999) 2067–2088.

[68] V. Van, A hybrid Implicit–explicit FDTD scheme for nonlinear optical waveguide modeling, IEEE. Trans. Microwave Theory and Tech. 47 (5) (1999) 540–545.

[69] J.H. Verner, High-order explicit Runge–Kutta pairs with low stage order, Appl. Numer. Math. 22 (1–3) (1996) 345–357.

[70] J.G. Verwer, An analysis of Rosenbrock methods for nonlinear stiff initial value problems, SIAM J. Numer. Anal. 19 (1) (1981) 155–170.

[71] R. Weiner, M. Arnold, P. Rentrop, K. Strehmel, Partitioning strategies in Runge–Kutta type methods, IMA J. Numer. Anal. 13 (2) (1993) 303–319.

[72] S. Wolfram, The Mathematica Book, 4th Edition, Cambridge University Press, Cambridge, 1999.

[73] Mathematica 4.0 Standard Add-On Packages, Wolfram Research, Champaing, IL, 1999.

[74] A. Xiao, S. Li, Error of partitioned Runge–Kutta methods for multiple stiff singular perturbation problems, Computing 64 (2) (2000) 183–189.

[75] J.J.-I. Yoh, X. Zhong, Semi-implicit Runge–Kutta schemes for stiff multi-dimensional reacting flows, AIAA Paper 97-0803, AIAA, Aerospace Sciences Meeting and Exhibit, 35th, Reno, NV, January 6–9, 1997.

[76] J.J.-I. Yoh, X. Zhong, Low-storage semi-implicit Runge–Kutta methods for reactive flow computations, AIAA Paper 98-0130, AIAA, Aerospace Sciences Meeting and Exhibit, 36th, Reno, NV, January 12–15, 1998.

[77] X. Zhong, New high-order semi-implicit Runge–Kutta schemes for computing transient nonequilibrium hypersonic flows, AIAA Paper 95-2007, AIAA, Thermophysics Conference, 30th, San Diego, CA, June 19–22, 1995.

[78] X. Zhong, Additive semi-implicit Runge–Kutta methods for computing high-speed nonequilibrium reactive flows, J. Comput. Phys. 128 (1) (1996) 19–31.