Journal of Computational Physics 286 (2015) 172–193

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

# Low-storage implicit/explicit Runge–Kutta schemes for the simulation of stiff high-dimensional ODE systems

Daniele Cavaglieri [*], Thomas Bewley

ARTICLE INFO

*Article history:*
Received 14 December 2013
Received in revised form 19 January 2015
Accepted 21 January 2015
Available online 26 January 2015

*Keywords:*
Stiff ODE/PDE/DAE time marching
Low-storage IMEXRK methods
SSP/TVD methods
*L* stability

ABSTRACT

*Implicit/explicit* (IMEX) Runge–Kutta (RK) schemes are effective for time-marching ODE systems with both stiff and nonstiff terms on the RHS; such schemes implement an (often *A*-stable or better) implicit RK scheme for the stiff part of the ODE, which is often linear, and, simultaneously, a (more convenient) explicit RK scheme for the nonstiff part of the ODE, which is often nonlinear. *Low-storage* RK schemes are especially effective for time-marching high-dimensional ODE discretizations of PDE systems on modern (cache-based) computational hardware, in which memory management is often the most significant computational bottleneck. In this paper, we develop and characterize eight new *low-storage implicit/explicit* RK schemes which have higher accuracy and better stability properties than the only low-storage implicit/explicit RK scheme available previously, the venerable second-order Crank–Nicolson/Runge–Kutta–Wray (CN/RKW3) algorithm that has dominated the DNS/LES literature for the last 25 years, while requiring similar storage (two, three, or four registers of length *N*) and comparable floating-point operations per timestep.

Published by Elsevier Inc.

## 1. Introduction

Although a wide variety of methods have been used for spatial discretization and subgrid-scale modeling in the Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES) of turbulent flows, time marching schemes for such systems have relied, in most cases, on an implicit scheme for the advancement of the stiff terms and an explicit scheme for the advancement of the nonstiff terms. Among these so-called IMEX schemes, an approach that gained favor due to [11] and [12] coupled the (implicit, second-order) Crank–Nicolson (CN) scheme for the stiff terms with the (explicit) second-order Adams–Bashforth (AB2) scheme for the nonstiff terms. This approach was refined in [13], which used the (implicit) CN scheme for the stiff terms, at each RK substep, together with the (explicit) third-order low-storage Runge–Kutta–Wray (RKW3) scheme [22] for the nonstiff terms. This venerable IMEX algorithm, dubbed CN/RKW3, still enjoys extensive use today, and is particularly appealing, as only two registers are required for advancing the ODE in time, though if three registers are used, the number of flops required by the algorithm may be significantly reduced. In high-dimensional discretizations of 3D PDE systems on modern computational hardware, the reduced memory footprint of this time marching algorithm, in its two-register or three-register form, can significantly reduce the execution time of a simulation. However, the CN/RKW3 scheme has the considerable disadvantage of being only second-order accurate, and its implicit part is only *A*-stable. In recent years, there have been relatively few attempts to refine the CN/RKW3 time-marching scheme for turbulence simulations, perhaps due to a mistaken notion that modifying it to achieve higher order might result in either increased storage requirements,

[*] Corresponding author.
*E-mail addresses:* dcavagli@ucsd.edu (D. Cavaglieri), bewley@ucsd.edu (T. Bewley).

http://dx.doi.org/10.1016/j.jcp.2015.01.031
0021-9991/Published by Elsevier Inc.

significantly more computation per timestep, or the loss of $A$ stability of the implicit part. It turns out that this is untrue; in fact, there is much to be gained by revising this algorithm.

When using an IMEX scheme, such as those described above, to march the incompressible Navier–Stokes equation, one natural choice is to treat the (linear) diffusion terms as the "stiff terms" and the (nonlinear) convective terms as the "nonstiff terms". Note that a better choice for discretizations with significant grid clustering implemented in one or more spatial directions, as usually present when simulating wall-bounded turbulent flows, is to treat the diffusion and linearized convection terms with derivatives in the direction of most significant grid clustering (e.g., in the direction normal to the nearest wall) as the "stiff" terms, and the remaining terms as the "nonstiff" terms, as suggested by [1]. Note further that so-called fractional step methods are often combined with such IMEX schemes in order to enforce the incompressibility constraint (see, e.g., [13]). The present paper focuses exclusively on the IMEXRK part of such time-advancement algorithms; various creative choices for which terms to take implicitly at different points in the physical domain of interest, and various methods for implementing fractional step techniques for enforcing exactly the divergence-free constraint, may subsequently be addressed in an identical manner as discussed in [1] and [13], and elsewhere in the literature.

Over the last 30 years, there has been significant development of (full-storage) IMEXRK algorithms. A comprehensive review of this literature is given in [9], and a brief summary of this subject is given in Section 1.1 below, including the general structure of full-storage IMEXRK schemes, their general implementation, conditions on their parameters for second-, third-, and fourth-order accuracy, and characterizations of their stability.

Further, in the years since the development of RKW3 in [22], there has been significant development of alternative low-storage explicit RK schemes; a comprehensive review of this literature is given in [10], and a brief summary of this subject is given in Section 1.2 below, including the extension to implicit RK schemes, the introduction of a general 2-register IMEXRK form, efficient 3-register and 2-register implementations of this form, as well as the introduction of a general 3-register IMEXRK form, and efficient 4-register and 3-register implementations of this form.

We then develop eight new low-storage IMEXRK schemes well suited for turbulent flow simulations, and other computational grand challenge applications, using two, three, or four registers of length $N$ (the dimension of the ODE under consideration). With an eye on the computational cost of their implementation, we focus on schemes with the smallest number of stages possible for a given order, stability, and storage requirement. A comprehensive summary of the schemes developed in this paper is given in Table 1. In short:

- Section 2 presents two second-order, 2-register IMEXRK schemes:
  - the classic 3-stage, $A$-stable, CN/RKW3 scheme, and
  - a new, $(2,3)$-stage [that is, a scheme with 2 implicit stages and 3 explicit stages], $L$-stable, strong-stability-preserving scheme, dubbed **IMEXRKCB2**.
- Section 3 presents five new third-order, 2-register IMEXRK schemes:
  - a $(2,3)$-stage, strongly $A$-stable scheme, dubbed **IMEXRKCB3a**,
  - a $(3,4)$-stage, strongly $A$-stable scheme with ESDIRK implicit part, dubbed **IMEXRKCB3b**, and
  - three $(3,4)$-stage, $L$-stable schemes:
    - one with coefficients selected to maximize stability of the ERK part on the negative real axis while being strong stability preserving, dubbed **IMEXRKCB3c**,
    - one with coefficients selected to be strong stability preserving for the maximum possible timestep, dubbed **IMEXRKCB3d**, and
    - one with coefficients selected to maximize accuracy of the ERK part, dubbed **IMEXRKCB3e**.
- Section 4 presents a new third-order, 3-register, 4-stage, $L$-stable, stage-order-2 scheme dubbed **IMEXRKCB3f**.
- Section 5 presents a new fourth-order, 3-register, 6-stage, $L$-stable, stage-order-2 scheme dubbed **IMEXRKCB4**.

In Section 6, we provide an analysis of the well-known order reduction phenomenon arising during the integration of very stiff ODEs using these IMEXRK schemes. Finally, Section 7 considers the application of all of these low-storage IMEXRK schemes, and some of their full-storage IMEXRK competitors, to a representative test problem in order to compare their computational efficiency.

### *1.1. Full-storage IMEXRK schemes and their Butcher tableaux*

A comprehensive review of (full-storage) IMEXRK schemes is given by Kennedy, Carpenter, and Lewis [9]. In short, IMEXRK schemes incorporate a coordinated pair of Diagonally Implicit Runge–Kutta (DIRK, with lower-triangular $A$) and Explicit Runge–Kutta (ERK, with strictly lower-triangular $A$) schemes, with parameters as summarized in the standard Butcher tableaux

**Table 1**
At a glance: summary of the properties of the eight IMEXRK schemes developed in this paper (top) and eight of the leading IMEXRK competitors (bottom), including the leading-order computational cost per timestep for efficient finite-difference (FD) and pseudospectral (PS) implementation of each scheme on the 1D Kuramoto–Sivashinsky (KS) equation. "SSP" means that the scheme is strong stability preserving under the appropriate timestep restriction, "embedded" means that a lower-order embedded scheme following the guidelines listed in Section 1.2 is provided, "ESDIRK" means that all diagonal components of the $A$ matrix of the associated DIRK scheme are equal (facilitating storage and reuse of an LU decomposition during the implicit solves), and "SO2" means the scheme is stage order 2.

| Scheme | Order | Registers | Stages $(s^{\mathrm{IM}}, s^{\mathrm{EX}})$ | Stability of DIRK part $[\sigma(z^{\mathrm{IM}} \to \infty; z^{\mathrm{EX}})]$ | Stability of ERK part on negative real axis | Truncation error | Other properties | FD cost for 1D KS | PS cost |
|---|---|---|---|---|---|---|---|---|---|
| IMEXRKCB2 | second | [2R] | $(2, 3)$ | $L$-stable [0] | $-5.81 \le z^{\mathrm{EX}} \le 0$ | $A^{(3)} = 0.114$ | embedded,<br>SSP ($c = 1.0$) | $90N$ flops (3-reg),<br>$101N$ flops (2-reg) | 6 FFTs<br>(3-reg) |
| IMEXRKCB3a | third | [2R] | $(2, 3)$ | strongly $A$-stable<br>$[-0.738]$ | $-2.51 \le z^{\mathrm{EX}} \le 0$ | $A^{(4)} = 0.226$ | | $90N$ flops (3-reg),<br>$101N$ flops (2-reg) | 6 FFTs<br>(3-reg) |
| IMEXRKCB3b | | | $(3, 4)$ | strongly $A$-stable<br>$[-0.732 - 0.366z^{\mathrm{EX}}]$ | $-2.21 \le z^{\mathrm{EX}} \le 0$ | $A^{(4)} = 0.186$ | ESDIRK | $130N$ flops (3-reg),<br>$139N$ flops (2-reg) | 8 FFTs<br>(3-reg) |
| IMEXRKCB3c | | | | $L$-stable [0] | $-6.00 \le z^{\mathrm{EX}} \le 0$ | $A^{(4)} = 0.113$ | embedded,<br>SSP ($c = 0.70$) | $133N$ flops (3-reg),<br>$157N$ flops (2-reg) | 8 FFTs<br>(3-reg) |
| IMEXRKCB3d | | | | | $-2.52 \le z^{\mathrm{EX}} \le 0$ | $A^{(4)} = 0.207$ | embedded,<br>SSP ($c = 0.77$) | | |
| IMEXRKCB3e | | | | | $-2.79 \le z^{\mathrm{EX}} \le 0$ | $A^{(4)} = 0.0824$ | | | |
| IMEXRKCB3f | | [3R] | $(4, 4)$ | $L$-stable [0] | $-6.00 \le z^{\mathrm{EX}} \le 0$ | $A^{(4)} = 0.107$ | embedded,<br>$\mathrm{SO}_2$ | $162N$ flops (4-reg),<br>$266N$ flops (3-reg) | 8 FFTs<br>(4-reg) |
| IMEXRKCB4 | fourth | [3R] | $(6, 6)$ | $L$-stable [0] | $-6.32 \le z^{\mathrm{EX}} \le 0$ | $A^{(5)} = 0.0157$ | embedded,<br>$\mathrm{SO}_2$ | $253N$ flops (4-reg),<br>$458N$ flops (3-reg) | 12 FFTs<br>(4-reg) |
| CN/RKW3 | second | [2R] | $(3, 3)$ | $A$-stable $[-1]$ | $-2.51 \le z^{\mathrm{EX}} \le 0$ | $A^{(3)} = 0.0387$ | | $115N$ flops (3-reg),<br>$127N$ flops (2-reg) | 6 FFTs<br>(3-reg) |
| Ascher(2, 3, 3)<br>{see [2]} | third | 7 | $(2, 3)$ | strongly $A$-stable<br>$[-0.732 - 0.732z^{\mathrm{EX}}]$ | $-2.51 \le z^{\mathrm{EX}} \le 0$ | $A^{(4)} = 0.206$ | | $92N$ flops | 6 FFTs |
| Ascher(3, 4, 3)<br>{see [2]} | | 9 | $(3, 4)$ | $L$-stable $[0.106z^{\mathrm{EX}}]$ | $-2.78 \le z^{\mathrm{EX}} \le 0$ | $A^{(4)} = 0.103$ | | $141N$ flops | 8 FFTs |
| Ascher(4, 4, 3)<br>{see [2]} | | 10 | $(4, 4)$ | $L$-stable [0] | $-2.14 \le z^{\mathrm{EX}} \le 0$ | $A^{(4)} = 0.163$ | | $190N$ flops | 8 FFTs |
| LIRK3<br>{see [4]} | | 9 | $(3, 4)$ | | $-2.21 \le z^{\mathrm{EX}} \le 0$ | $A^{(4)} = 0.100$ | | $139N$ flops | 8 FFTs |
| ARK3(2)4L[2]SA<br>{see [9]} | | 10 | $(4, 4)$ | | $-3.66 \le z^{\mathrm{EX}} \le 0$ | $A^{(4)} = 0.0722$ | embedded | $159N$ flops | 8 FFTs |
| LIRK4<br>{see [4]} | fourth | 13 | $(5, 6)$ | $L$-stable [0] | $-3.41 \le z^{\mathrm{EX}} \le 0$ | $A^{(5)} = 0.0404$ | | $249N$ flops | 12 FFTs |
| ARK4(3)6L[2]SA<br>{see [9]} | | 14 | $(6, 6)$ | | $-4.23 \le z^{\mathrm{EX}} \le 0$ | $A^{(5)} = 0.0122$ | embedded | $270N$ flops | 12 FFTs |

$$
\begin{array}{c|ccccc}
c_1^{\mathrm{IM}} & a_{1,1}^{\mathrm{IM}} & & & & \\
c_2^{\mathrm{IM}} & a_{2,1}^{\mathrm{IM}} & a_{2,2}^{\mathrm{IM}} & & & \\
\vdots & \vdots & \ddots & \ddots & & \\
c_s^{\mathrm{IM}} & a_{s,1}^{\mathrm{IM}} & \cdots & a_{s,s-1}^{\mathrm{IM}} & a_{s,s}^{\mathrm{IM}} & \\
\hline
 & b_1^{\mathrm{IM}} & \cdots & b_{s-1}^{\mathrm{IM}} & b_s^{\mathrm{IM}} & \\
\hline
 & \hat{b}_1^{\mathrm{IM}} & \cdots & \hat{b}_{s-1}^{\mathrm{IM}} & \hat{b}_s^{\mathrm{IM}} &
\end{array}
\qquad
\begin{array}{c|ccccc}
c_1^{\mathrm{EX}} & 0 & & & & \\
c_2^{\mathrm{EX}} & a_{2,1}^{\mathrm{EX}} & 0 & & & \\
\vdots & \vdots & \ddots & \ddots & & \\
c_s^{\mathrm{EX}} & a_{s,1}^{\mathrm{EX}} & \cdots & a_{s,s-1}^{\mathrm{EX}} & 0 & \\
\hline
 & b_1^{\mathrm{EX}} & \cdots & b_{s-1}^{\mathrm{EX}} & b_s^{\mathrm{EX}} & \\
\hline
 & \hat{b}_1^{\mathrm{EX}} & \cdots & \hat{b}_{s-1}^{\mathrm{EX}} & \hat{b}_s^{\mathrm{EX}} &
\end{array}
\tag{1}
$$

for the time advancement of an ODE of the form

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x}, t), \tag{2}$$

where $\mathbf{f}(\mathbf{x}, t)$ represents the stiff part of the RHS [advanced with the DIRK method at left in (1)], and $\mathbf{g}(\mathbf{x}, t)$ represents the nonstiff part of the RHS [simultaneously advanced with the ERK method at right in (1)].

If the stiff part of the ODE is linear [that is, if $\mathbf{f}(\mathbf{x}, t) = A\mathbf{x}$] then, denoting the efficient solution of $A\mathbf{x} = \mathbf{b}$ as $A^{-1}\mathbf{b}$, a full-storage implementation of the IMEXRK scheme in (1) to advance from $\mathbf{x} = \mathbf{x}_n$ to $\mathbf{x} = \mathbf{x}_{n+1}$ proceeds as follows

```
for  $k = 1 : s$                                                                    (3a)
  if  $k == 1$,  $\mathbf{y} = \mathbf{x}$,  else,  $\mathbf{y} = \mathbf{x} + \sum_{i=1}^{k-1} a_{k,i}^{\mathrm{IM}}\, \Delta t\, \mathbf{f}_i + \sum_{j=1}^{k-1} a_{k,j}^{\mathrm{EX}}\, \Delta t\, \mathbf{g}_j$,  end      (3b)
  $\mathbf{f}_k = A\left(I - a_{k,k}^{\mathrm{IM}}\, \Delta t\, A\right)^{-1}\mathbf{y}$   $\left[\text{equivalently, } \mathbf{f}_k = \left(I - a_{k,k}^{\mathrm{IM}}\, \Delta t\, A\right)^{-1} A\,\mathbf{y}\right]$      (3c)
  $\mathbf{g}_k = \mathbf{g}\big(\mathbf{y} + a_{kk}^{\mathrm{IM}}\, \Delta t\, \mathbf{f}_k,\ t_n + c_k^{\mathrm{EX}}\, \Delta t\big)$                                   (3d)
end                                                                                  (3e)
$\mathbf{x} \leftarrow \mathbf{x} + \sum_{i=1}^{s} b_i^{\mathrm{IM}}\, \Delta t\, \mathbf{f}_i + \sum_{j=1}^{s} b_j^{\mathrm{EX}}\, \Delta t\, \mathbf{g}_j$                                  (3f)
$\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \sum_{i=1}^{s} \hat{b}_i^{\mathrm{IM}}\, \Delta t\, \mathbf{f}_i + \sum_{j=1}^{s} \hat{b}_j^{\mathrm{EX}}\, \Delta t\, \mathbf{g}_j$                                  (3g)
```

Line (3c) above is simply $\mathbf{f}_k = \mathbf{f}(\mathbf{z}, t_n + c_k^{\mathrm{IM}}\Delta t)$, where $\mathbf{z}$ is the solution of $\mathbf{e}(\mathbf{z}) = \mathbf{z} - \mathbf{y} - a_{kk}^{\mathrm{IM}}\, \Delta t\, \mathbf{f}(\mathbf{z}, t_n + c_k^{\mathrm{IM}}\Delta t) = 0$ [that is, where $\mathbf{z} = \mathbf{y} + a_{kk}^{\mathrm{IM}}\, \Delta t\, \mathbf{f}(\mathbf{z}, t_n + c_k^{\mathrm{IM}}\Delta t)$], in the special case that $\mathbf{f}(\mathbf{x}, t) = A\mathbf{x}$. More generally, if the stiff part $\mathbf{f}(\mathbf{x}, t)$ is nonlinear, then line (3c) is replaced by a Newton–Raphson iteration (see [16]) to find the $\mathbf{z}$ such that $\mathbf{e}(\mathbf{z}) = 0$:

$$
\left.
\begin{array}{ll}
\text{Initialize:} & \mathbf{z}_0 = \mathbf{y} + a_{kk}^{\mathrm{IM}}\, \Delta t\, \mathbf{f}\big(\mathbf{y}, t_n + c_k^{\mathrm{IM}}\Delta t\big) \\
\text{Iterate:} & \left(I - a_{kk}^{\mathrm{IM}}\, \Delta t\, \left.\dfrac{\partial \mathbf{f}(\mathbf{x},\, t_n + c_k^{\mathrm{IM}}\Delta t)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{z}_m}\right)(\mathbf{z}_{m+1} - \mathbf{z}_m) = -\mathbf{z}_m + \mathbf{y} + a_{kk}^{\mathrm{IM}}\, \Delta t\, \mathbf{f}\big(\mathbf{z}_m,\ t_n + c_k^{\mathrm{IM}}\Delta t\big) \\
\text{Upon exit:} & \mathbf{f}_k = \mathbf{f}\big(\mathbf{z}_{\text{converged}},\ t_n + c_k^{\mathrm{IM}}\Delta t\big)
\end{array}
\right\}
\tag{3c$'$}
$$

The Jacobian used in this iteration may be computed analytically or approximated numerically. The low-storage IMEXRK algorithms developed in this work may be applied in the linear or nonlinear setting, mutatis mutandis; Sections 1.2.1–1.2.4 provide low-storage pseudocode implementations in the case in which the stiff part of the ODE is linear.

Finally, note that the $\hat{b}_i^{\mathrm{IM}}$ and $\hat{b}_i^{\mathrm{EX}}$ coefficients in the Butcher tableaux, if provided, are used to form a so-called embedded scheme to advance the solution at each timestep with an order of accuracy reduced by one with respect to the main scheme. Using this embedded scheme, one may estimate the error of the simulation at each timestep, and adjust the stepsize at the next iteration accordingly.

As is well known (see, e.g., [3]), for the DIRK and ERK components in (1), when used in isolation, to be first-order accurate, it is required that

$$\tau_1^{\mathrm{IM}(1)} = \sum_i b_i^{\mathrm{IM}} - 1 = 0 \qquad \tau_1^{\mathrm{EX}(1)} = \sum_i b_i^{\mathrm{EX}} - 1 = 0, \tag{4a}$$

for these schemes, when used in isolation, to be second-order accurate, it is additionally required that

$$\tau_1^{\mathrm{IM}(2)} = \sum_i b_i^{\mathrm{IM}} c_i^{\mathrm{IM}} - 1/2 = 0 \qquad \tau_1^{\mathrm{EX}(2)} = \sum_i b_i^{\mathrm{EX}} c_i^{\mathrm{EX}} - 1/2 = 0, \tag{4b}$$

for these schemes, when used in isolation, to be third-order accurate, it is additionally required that

$$\tau_1^{\mathrm{IM}(3)} = (1/2)\sum_i b_i^{\mathrm{IM}} c_i^{\mathrm{IM}} c_i^{\mathrm{IM}} - 1/6 = 0 \qquad \tau_1^{\mathrm{EX}(3)} = (1/2)\sum_i b_i^{\mathrm{EX}} c_i^{\mathrm{EX}} c_i^{\mathrm{EX}} - 1/6 = 0 \tag{4c}$$

$$\tau_2^{\mathrm{IM}(3)} = \sum_{i,\,j} b_i^{\mathrm{IM}} a_{ij}^{\mathrm{IM}} c_j^{\mathrm{IM}} - 1/6 = 0 \qquad \tau_2^{\mathrm{EX}(3)} = \sum_{i,\,j} b_i^{\mathrm{EX}} a_{ij}^{\mathrm{EX}} c_j^{\mathrm{EX}} - 1/6 = 0, \tag{4d}$$

and for these schemes, when used in isolation, to be fourth-order accurate, it is additionally required that

$$\tau_1^{\mathrm{IM}(4)} = (1/6)\sum_i b_i^{\mathrm{IM}} c_i^{\mathrm{IM}} c_i^{\mathrm{IM}} c_i^{\mathrm{IM}} - 1/24 = 0 \qquad \tau_1^{\mathrm{EX}(4)} = (1/6)\sum_i b_i^{\mathrm{EX}} c_i^{\mathrm{EX}} c_i^{\mathrm{EX}} c_i^{\mathrm{EX}} - 1/24 = 0 \tag{4e}$$

$$\tau_2^{\mathrm{IM}(4)} = (1/3)\sum_{i,j} b_i^{\mathrm{IM}} c_i^{\mathrm{IM}} a_{ij}^{\mathrm{IM}} c_j^{\mathrm{IM}} - 1/24 = 0 \qquad \tau_2^{\mathrm{EX}(4)} = (1/3)\sum_{i,j} b_i^{\mathrm{EX}} c_i^{\mathrm{EX}} a_{ij}^{\mathrm{EX}} c_j^{\mathrm{EX}} - 1/24 = 0 \tag{4f}$$

$$\tau_3^{\mathrm{IM}(4)} = (1/2)\sum_{i,j} b_i^{\mathrm{IM}} a_{ij}^{\mathrm{IM}} c_j^{\mathrm{IM}} c_j^{\mathrm{IM}} - 1/24 = 0 \qquad \tau_3^{\mathrm{EX}(4)} = (1/2)\sum_{i,j} b_i^{\mathrm{EX}} a_{ij}^{\mathrm{EX}} c_j^{\mathrm{EX}} c_j^{\mathrm{EX}} - 1/24 = 0 \tag{4g}$$

$$\tau_4^{\mathrm{IM}(4)} = \sum_{i,j,k} b_i^{\mathrm{IM}} a_{ij}^{\mathrm{IM}} a_{jk}^{\mathrm{IM}} c_k^{\mathrm{IM}} - 1/24 = 0 \qquad \tau_4^{\mathrm{EX}(4)} = \sum_{i,j,k} b_i^{\mathrm{EX}} a_{ij}^{\mathrm{EX}} a_{jk}^{\mathrm{EX}} c_k^{\mathrm{EX}} - 1/24 = 0. \tag{4h}$$

Recall that, in the scalar case, the exact solution of $x' = f(x) + g(x)$ has the following terms:

$$\begin{aligned} x_{n+1} &= x_n + \Delta t\, x_n' + (\Delta t)^2 x_n''/2! + (\Delta t)^3 x_n'''/3! + O\big((\Delta t)^4\big) \\ &= x_n + \Delta t\{f+g\}_{(x_n,t_n)} + \frac{(\Delta t)^2}{2!}\big\{f'f + f'g + g'f + g'g\big\}_{(x_n,t_n)} + \frac{(\Delta t)^3}{3!}\big\{f''ff + 2f''fg + f''gg + g''ff \\ &\quad + 2g''fg + g''gg + f'f'f + f'g'f + g'f'f + g'g'f + f'f'g + f'g'g + g'f'g + g'g'g\big\}_{(x_n,t_n)} + O\big((\Delta t)^4\big); \end{aligned}$$

note in particular that there are 2 terms at second order and 10 terms at third order that involve both $f$ and $g$. For the DIRK and ERK components in (1), when used together in an IMEX fashion, to be second-order accurate, it is thus additionally required that

$$\tau_1^{\mathrm{IMEX}(2)} = \sum_i b_i^{\mathrm{IM}} c_i^{\mathrm{EX}} - 1/2 = 0 \qquad \tau_2^{\mathrm{IMEX}(2)} = \sum_i b_i^{\mathrm{EX}} c_i^{\mathrm{IM}} - 1/2 = 0, \tag{4i}$$

for these schemes, when used together in an IMEX fashion, to be third-order accurate, it is additionally required that

$$\tau_1^{\mathrm{IMEX}(3)} = (1/2)\sum_i b_i^{\mathrm{IM}} c_i^{\mathrm{EX}} c_i^{\mathrm{EX}} - 1/6 = 0 \qquad \tau_2^{\mathrm{IMEX}(3)} = (1/2)\sum_i b_i^{\mathrm{EX}} c_i^{\mathrm{IM}} c_i^{\mathrm{IM}} - 1/6 = 0 \tag{4j}$$

$$\tau_3^{\mathrm{IMEX}(3)} = (1/2)\sum_i b_i^{\mathrm{IM}} c_i^{\mathrm{IM}} c_i^{\mathrm{EX}} - 1/6 = 0 \qquad \tau_4^{\mathrm{IMEX}(3)} = (1/2)\sum_i b_i^{\mathrm{EX}} c_i^{\mathrm{IM}} c_i^{\mathrm{EX}} - 1/6 = 0 \tag{4k}$$

$$\tau_5^{\mathrm{IMEX}(3)} = \sum_{i,\,j} b_i^{\mathrm{IM}} a_{ij}^{\mathrm{EX}} c_j^{\mathrm{EX}} - 1/6 = 0 \qquad \tau_6^{\mathrm{IMEX}(3)} = \sum_{i,\,j} b_i^{\mathrm{EX}} a_{ij}^{\mathrm{IM}} c_j^{\mathrm{IM}} - 1/6 = 0 \tag{4l}$$

$$\tau_7^{\mathrm{IMEX}(3)} = \sum_{i,\,j} b_i^{\mathrm{EX}} a_{ij}^{\mathrm{EX}} c_j^{\mathrm{IM}} - 1/6 = 0 \qquad \tau_8^{\mathrm{IMEX}(3)} = \sum_{i,\,j} b_i^{\mathrm{IM}} a_{ij}^{\mathrm{IM}} c_j^{\mathrm{EX}} - 1/6 = 0 \tag{4m}$$

$$\tau_9^{\mathrm{IMEX}(3)} = \sum_{i,\,j} b_i^{\mathrm{IM}} a_{ij}^{\mathrm{EX}} c_j^{\mathrm{IM}} - 1/6 = 0 \qquad \tau_{10}^{\mathrm{IMEX}(3)} = \sum_{i,\,j} b_i^{\mathrm{EX}} a_{ij}^{\mathrm{IM}} c_j^{\mathrm{EX}} - 1/6 = 0, \tag{4n}$$

and for these schemes, when used together in an IMEX fashion, to be fourth-order accurate, 44 additional constraints are required (see [9]), which for brevity aren't listed here.

#### *1.1.1. Stability*

The stability of an RK scheme may be characterized by considering the model problem $dx/dt = \lambda x$ and defining $z = \lambda\,\Delta t$, $\sigma(z) = x_{n+1}/x_n$, and $\sigma(\infty) \triangleq \lim_{|z|\to\infty}\sigma(z)$. The stability function of an RK scheme with Butcher tableau parameters $A$ and $\mathbf{b}$ is then given by $\sigma(z) = 1 + z\mathbf{b}^T(I - zA)^{-1}\mathbf{1}$, where $\mathbf{1}$ denotes a vector of ones; the RK scheme is said to be stable for any $z$ such that $|\sigma(z)| \le 1$. Further, considering its application to stiff systems, an RK scheme is said to be

- $A$-stable if $|\sigma(z)| \le 1$ over the entire LHP of $z$,
- strongly $A$-stable if it is $A$-stable and $|\sigma(\infty)| < 1$, and
- $L$-stable if it is $A$-stable and $\sigma(\infty) = 0$.

The stability of an IMEXRK scheme is a bit more difficult to characterize. Of course, one may start by characterizing the stability of the implicit and explicit parts considered in isolation. To evaluate the stability of the implicit and explicit components of an IMEX scheme working together, we consider the model problem $dx/dt = \lambda_f x + \lambda_g x$, where the first term on the RHS is handled implicitly, and the second term on the RHS is handled explicitly. Defining $z^{\mathrm{IM}} = \lambda_f \Delta t$, $z^{\mathrm{EX}} = \lambda_g \Delta t$, and $\sigma(z^{\mathrm{IM}}; z^{\mathrm{EX}}) = x_{n+1}/x_n$, we may write (see [9])

$$\sigma\left(z^{\mathrm{IM}}; z^{\mathrm{EX}}\right) = \frac{\det[I - z^{\mathrm{IM}} A^{\mathrm{IM}} - z^{\mathrm{EX}} A^{\mathrm{EX}} + z^{\mathrm{IM}} \mathbf{1} (\mathbf{b}^{\mathrm{IM}})^T + z^{\mathrm{EX}} \mathbf{1} (\mathbf{b}^{\mathrm{EX}})^T]}{\det[I - z^{\mathrm{IM}} A^{\mathrm{IM}}]}. \tag{5}$$

We may then characterize the stability of the implicit and explicit parts of an IMEXRK scheme working in concert, when the implicit part of the problem is stiff, by looking at $\sigma(z^{\mathrm{IM}}; z^{\mathrm{EX}})$ as $z^{\mathrm{IM}} \to \infty$ for finite $z^{\mathrm{EX}}$.

### *1.1.2. Strong-stability preserving (SSP) schemes*

Consider the 1D hyperbolic PDE

$$\partial u/\partial t = -\partial f(u)/\partial x; \tag{6}$$

denoting by $u_i(t)$ the discretization of $u(x,t)$ on $N$ spatial grid points $x_i$, and by $\mathbf{u}(t)$ a vector containing all of the $u_i(t)$, we write the spatial discretization of this PDE as the ODE

$$d\mathbf{u}/dt = L(\mathbf{u}). \tag{7}$$

If a TVD spatial discretization is used, such as a Godunov or MUSCL scheme with an appropriate flux limiter incorporated (see [14]), then applying a simple Explicit Euler time discretization to (7),

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, L\big(\mathbf{u}^n\big), \tag{8}$$

under the appropriate CFL condition on the timestep, $\Delta t \le \Delta t_{CFL}$, results in a simulation exhibiting a total variation of the discrete solution which does not increase in time, that is,

$$TV\big(\mathbf{u}^{n+1}\big) \le TV\big(\mathbf{u}^n\big), \quad \text{where } TV\big(\mathbf{u}^n\big) = \sum_j \big|u_{j+1}^n - u_j^n\big|. \tag{9}$$

Strong-stability preserving (SSP) explicit time-discretization methods (see [17] and [18]) are simply higher-order time discretization methods that conserve this total variation diminishing property with a modified CFL condition on the timestep, $\Delta t \le c\,\Delta t_{CFL}$.

In [18] (see also [6]), a condition for an explicit Runge–Kutta scheme to be SSP has been developed. This condition states that if an $s$-stage explicit Runge–Kutta scheme is written in incremental form, that is,

$$\begin{aligned} &\mathbf{u}^{(0)} = \mathbf{u}^n \\ &\mathbf{u}^{(i)} = \sum_{j=0}^{i-1} \big(\alpha_{ij}\mathbf{u}^{(j)} + \Delta t \beta_{ij} \mathbf{L}\big(\mathbf{u}^{(j)}\big)\big) \quad \text{for } i = 1, \dots, s \\ &\mathbf{u}^{n+1} = \mathbf{u}^{(s)}, \end{aligned}$$

where all of the $\alpha_{ij} \ge 0$, and if the forward Euler method applied to the ODE (7) arising from a TVD spatial discretization of the hyperbolic PDE (6) is strongly stable under the appropriate CFL restriction, then such an explicit Runge–Kutta method is SSP provided that all of the $\beta_{ij} \ge 0$ and that the following CFL restriction is fulfilled:

$$\Delta t \le c\,\Delta t_{CFL}, \quad c = \min_{i,j} \frac{\alpha_{ij}}{\beta_{ij}}. \tag{10}$$

In case an explicit scheme is coupled with an implicit scheme, as in an IMEXRK formulation, then, provided the implicit scheme used to integrate the stiff part of the ODE is $L$-stable, in the stiff limit the time integration scheme becomes the explicit Runge–Kutta scheme, and the order of accuracy of the limiting scheme is greater than or equal to the order of accuracy of the IMEXRK scheme itself. Hence, as stated in [15], if the explicit part of the IMEXRK scheme is SSP, then the IMEXRK scheme will also be SSP in the stiff limit.

In [15], three full-storage second-order and two full-storage third-order IMEXRK schemes are presented which are SSP in the stiff limit; no other IMEXRK schemes with this SSP property were found in our review of the IMEXRK literature. The present paper derives three new IMEXRK schemes which are SSP in the stiff limit (one which is second-order and two which are third-order); unlike the schemes in [15], the IMEXRK schemes derived here are of the low-storage variety.

### *1.2. Low-storage IMEXRK schemes*

The existing literature on low-storage RK schemes to date appears to focus exclusively on explicit schemes. Note that a cavalier implementation of a full-storage ERK scheme [see the explicit part of (3)] requires storage of the state vector [$\mathbf{x}$], the intermediate vector [$\mathbf{y}$], and $s$ values of the RHS vectors [$\mathbf{g}_k$]; that is, $s+2$ vectors of length $N$, where $\mathbf{x} = \mathbf{x}_{N\times 1}$. We now summarize the two main classes of low-storage ERK schemes,[1] a comprehensive review of which is given in Kennedy, Carpenter, and Lewis [10].

The two-register Williamson class of ERK schemes [20], denoted "[2$N$]" schemes, may be written to advance from $\mathbf{x} = \mathbf{x}_n$ to $\mathbf{x} = \mathbf{x}_{n+1}$ as

```
for  $k = 1 : s$
  if  $k == 1$,   $\Delta\mathbf{x} \leftarrow \Delta t\,\mathbf{g}(\mathbf{x},\, t_n + c_k \Delta t)$,   else
    $\Delta\mathbf{x} \leftarrow \alpha_k\,\Delta\mathbf{x} + \Delta t\,\mathbf{g}(\mathbf{x},\, t_n + c_k \Delta t)$
  end
  $\mathbf{x} \leftarrow \mathbf{x} + \beta_k\,\Delta\mathbf{x}$
end                                                                    (11)
```

If handled with care, such schemes can often be implemented efficiently in two registers of length $N$, $\mathbf{x}$ and $\Delta\mathbf{x}$.

The two-register van der Houwen class of schemes [19], denoted "[2R]" schemes, restrict the parameters $a_{ij}$ below the first subdiagonal in the Butcher tableau of the ERK scheme to be equal to the parameters $b_j$ of the corresponding column, and may thus be written to advance from $\mathbf{x} = \mathbf{x}_n$ to $\mathbf{x} = \mathbf{x}_{n+1}$ as

```
for  $k = 1 : s$
  if  $k == 1$,   $\mathbf{y} \leftarrow \mathbf{x}$,   else
    $\mathbf{y} \leftarrow \mathbf{x} + (a_{k,k-1} - b_{k-1})\,\Delta t\,\mathbf{g}(\mathbf{y},\, t_n + c_{k-1}\Delta t)$
  end
  $\mathbf{x} \leftarrow \mathbf{x} + b_k\,\Delta t\,\mathbf{g}(\mathbf{y},\, t_n + c_k \Delta t)$
end                                                                    (12)
```

Such schemes can often be implemented efficiently in two registers of length $N$ (namely, $\mathbf{x}$ and $\mathbf{y}$). If implemented with three registers, however, the function $\mathbf{g}(\mathbf{y},\, t_n + c_k \Delta t)$ can be computed just once per timestep (instead of twice). RKW3 [22] is a commonly-used example of a two-register, three-stage, third-order van der Houwen ERK scheme, with a Butcher tableau of

$$\begin{array}{c|ccc} 0 & 0 & & \\ 8/15 & 8/15 & 0 & \\ 2/3 & 1/4 & 5/12 & 0 \\ \hline & 1/4 & 0 & 3/4 \end{array} \qquad (13)$$

In the three-register van der Houwen class of schemes, denoted "[3R]" schemes, only the parameters $a_{ij}$ below the *second* subdiagonal of the Butcher tableau of the ERK scheme must equal the parameters $b_j$ of the corresponding column. An effective implementation of such [3R] schemes that uses only three registers of length $N$ (namely, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$) is given by

```
for  $k = 1 : s$
  if  $k == 1$,   $\mathbf{y} \leftarrow \mathbf{x}$,   $\mathbf{z} \leftarrow \mathbf{x}$,   else,
    $\mathbf{z} \leftarrow \mathbf{y} + a_{k,k-1}\,\Delta t\,\mathbf{g}(\mathbf{y},\, t_n + c_{k-1}\Delta t)$
    if  $k < s$,   $\mathbf{y} \leftarrow \mathbf{x} + (a_{k+1,k-1} - b_{k-1})\,\mathbf{g}(\mathbf{y},\, t_n + c_{k-1}\Delta t)$,   end
  end
  $\mathbf{x} \leftarrow \mathbf{x} + b_k\,\Delta t\,\mathbf{g}(\mathbf{y},\, t_n + c_k \Delta t)$
end                                                                    (14)
```

Again, if implemented with four registers, the function $\mathbf{g}(\mathbf{y},\, t_n + c_k \Delta t)$ can be computed just once per timestep (instead of thrice). In the present work, we extend the two- and three-register van der Houwen classes of ERK schemes to the DIRK case, which can be accomplished with precisely the same restrictions on the (lower triangular) DIRK Butcher tableau as in the (strictly lower triangular) ERK case, as specified above. Further, we will develop coordinated pairs of such [2R] and [3R] DIRK and ERK schemes in the IMEX setting described in Section 1.1. In particular, we will develop a [2R] second-order IMEX scheme, [2R] and [3R] third-order IMEX schemes, and a [3R] fourth-order IMEX scheme.

As shown in Section 1.1, six constraints on the parameters of the IMEX Butcher tableaux (1) must be satisfied for second-order accuracy, fourteen additional constraints must be satisfied for third-order accuracy, and fifty-two additional constraints must be satisfied for fourth-order accuracy. Before proceeding, we thus introduce some significant simplifying assumptions. Following [15] and [9] and the CN/RKW3 scheme of [13], we synchronize the stages of DIRK and ERK components by imposing $c_i^{\mathrm{IM}} = c_i^{\mathrm{EX}} = c_i$ for $i = 1, \ldots, s$. We also coordinate the constituent DIRK and ERK components such that $b_i^{\mathrm{IM}} = b_i^{\mathrm{EX}} = b_i$

[1] Both the Williamson class and the van der Houwen class of ERK schemes extend to ERK variants that require, at minimum, three, four, or more registers for their implementation; with an eye on the computational cost of their implementation, we focus in this paper on schemes which admit a two- or three-register implementation.

for $i = 1, \dots, s$, as also done in [15] and [9], but which is not satisfied by CN/RKW3. Finally, for each stage, a stage-order of one is also imposed such that

$$\sum_{j=1}^{i} a_{ij}^{\text{IM}} = \sum_{j=1}^{i-1} a_{ij}^{\text{EX}} = c_i \quad \text{for } i = 1, \dots, s; \tag{15}$$

it follows that $c_1 = a_{11}^{\text{IM}} = a_{11}^{\text{EX}} = 0$. As a result of these assumptions, the number of constraints on the IMEX parameters [see (4)] for second-order accuracy is reduced to just two, the number of constraints for third-order accuracy is reduced to five, and the number of constraints for fourth-order accuracy is reduced to fourteen.

For several of the IMEXRK schemes developed in this paper, a lower-order embedded scheme is also developed, relaxing the $\hat{b}_i^{\text{IM}} = \hat{b}_i^{\text{EX}}$ restriction to provide increased freedom during the design phase. As a general guideline, none of the leading-order truncation terms of an embedded scheme should vanish, so that each of these terms will contribute to the error estimate (subject to this restriction, the remaining free parameters of the embedded scheme are then optimized to maximize the magnitude of the leading-order truncation terms). Unfortunately, this is not always achievable; as a result, not all of the schemes developed in this paper are listed with embedded schemes. For all of the embedded schemes we do report, the DIRK part of the embedded scheme is at least $A$-stable, which is a property of the embedded scheme recommended by [8]; note, however, that the embedded scheme is not used for time marching, it is only used to adjust the timestep.

The IMEX Butcher tableaux in (1) for coordinated pairs of [2R] DIRK and ERK schemes are thus simplified to

$$\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a_{2,1}^{\text{IM}} & a_{2,2}^{\text{IM}} & & & & \\
c_3 & b_1 & a_{3,2}^{\text{IM}} & a_{3,3}^{\text{IM}} & & & \\
c_4 & b_1 & b_2 & a_{4,3}^{\text{IM}} & a_{4,4}^{\text{IM}} & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\
c_s & b_1 & b_2 & \cdots & b_{s-2} & a_{s,s-1}^{\text{IM}} & a_{s,s}^{\text{IM}} \\
\hline
 & b_1 & b_2 & \cdots & b_{s-2} & b_{s-1} & b_s \\
\hline
 & \hat{b}_1^{\text{IM}} & \hat{b}_2^{\text{IM}} & \cdots & \hat{b}_{s-2}^{\text{IM}} & \hat{b}_{s-1}^{\text{IM}} & \hat{b}_s^{\text{IM}}
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a_{2,1}^{\text{EX}} & 0 & & & & \\
c_3 & b_1 & a_{3,2}^{\text{EX}} & 0 & & & \\
c_4 & b_1 & b_2 & a_{4,3}^{\text{EX}} & 0 & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\
c_s & b_1 & b_2 & \cdots & b_{s-2} & a_{s,s-1}^{\text{EX}} & 0 \\
\hline
 & b_1 & b_2 & \cdots & b_{s-2} & b_{s-1} & b_s \\
\hline
 & \hat{b}_1^{\text{EX}} & \hat{b}_2^{\text{EX}} & \cdots & \hat{b}_{s-2}^{\text{EX}} & \hat{b}_{s-1}^{\text{EX}} & \hat{b}_s^{\text{EX}}
\end{array} \tag{16}$$

and the IMEX Butcher tableaux for coordinated pairs of [3R] DIRK and ERK schemes are simplified to

$$\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a_{2,1}^{\text{IM}} & a_{2,2}^{\text{IM}} & & & & \\
c_3 & a_{3,1}^{\text{IM}} & a_{3,2}^{\text{IM}} & a_{3,3}^{\text{IM}} & & & \\
c_4 & b_1 & a_{4,2}^{\text{IM}} & a_{4,3}^{\text{IM}} & a_{4,4}^{\text{IM}} & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\
c_s & b_1 & b_2 & \cdots & a_{s,s-2}^{\text{IM}} & a_{s,s-1}^{\text{IM}} & a_{s,s}^{\text{IM}} \\
\hline
 & b_1 & b_2 & \cdots & b_{s-2} & b_{s-1} & b_s \\
\hline
 & \hat{b}_1^{\text{IM}} & \hat{b}_2^{\text{IM}} & \cdots & \hat{b}_{s-2}^{\text{IM}} & \hat{b}_{s-1}^{\text{IM}} & \hat{b}_s^{\text{IM}}
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a_{2,1}^{\text{EX}} & 0 & & & & \\
c_3 & a_{3,1}^{\text{EX}} & a_{3,2}^{\text{EX}} & 0 & & & \\
c_4 & b_1 & a_{4,2}^{\text{EX}} & a_{4,3}^{\text{EX}} & 0 & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\
c_s & b_1 & b_2 & \cdots & a_{s,s-2}^{\text{IM}} & a_{s,s-1}^{\text{EX}} & 0 \\
\hline
 & b_1 & b_2 & \cdots & b_{s-2} & b_{s-1} & b_s \\
\hline
 & \hat{b}_1^{\text{EX}} & \hat{b}_2^{\text{EX}} & \cdots & \hat{b}_{s-2}^{\text{EX}} & \hat{b}_{s-1}^{\text{EX}} & \hat{b}_s^{\text{EX}}
\end{array} \tag{17}$$

Note also that, as the DIRK component, the IMEXRK form considered above has an explicit first stage, its stability function (5) may be written

$$\sigma\left(z^{\text{IM}}; z^{\text{EX}}\right) = \frac{1 + \sum_{i=1}^{s} p_i(z^{\text{EX}})\,[z^{\text{IM}}]^i}{1 + \sum_{i=1}^{s-1} q_i\,[z^{\text{IM}}]^i} \quad \text{where } p_i\left(z^{\text{EX}}\right) = \sum_{j=0}^{s-i} \hat{p}_{ij}\,[z^{\text{EX}}]^j. \tag{18}$$

### *1.2.1. General three-register implementation of [2R] IMEXRK schemes*

Note that, if the stiff part of the ODE is linear [that is, if $\mathbf{f}(\mathbf{x}, t) = A\mathbf{x}$] then, denoting the efficient solution of $A\mathbf{x} = \mathbf{b}$ as $A^{-1}\mathbf{b}$, a straightforward implementation of the low-storage IMEXRK scheme in (16) that uses three registers[2] of length $N$ to advance from $\mathbf{x} = \mathbf{x}_n$ to $\mathbf{x} = \mathbf{x}_{n+1}$ proceeds as follows:

[2] That is, in addition to any extra memory required to solve the linear system, which is problem dependent, plus an additional register of length $N$ for the embedded scheme, if adaptive timestepping is implemented.

```
for  $k = 1:s$
  if  $k == 1$,  $\mathbf{y} \leftarrow \mathbf{x}$,  else
    $\mathbf{y} \leftarrow \mathbf{x} + \big(a^{\mathrm{IM}}_{k,k-1} - b^{\mathrm{IM}}_{k-1}\big)\,\Delta t\,\mathbf{z} + \big(a^{\mathrm{EX}}_{k,k-1} - b^{\mathrm{EX}}_{k-1}\big)\,\Delta t\,\mathbf{y}$
  end
  $\mathbf{z} = \big(I - a^{\mathrm{IM}}_{k,k}\,\Delta t\,A\big)^{-1} A\,\mathbf{y}$                                        (19)
  $\mathbf{y} \leftarrow \mathbf{g}\big(\mathbf{y} + a^{\mathrm{IM}}_{k,k}\,\Delta t\,\mathbf{z},\ t_n + c^{\mathrm{EX}}_k \Delta t\big)$
  $\mathbf{x} \leftarrow \mathbf{x} + b^{\mathrm{IM}}_k\,\Delta t\,\mathbf{z} + b^{\mathrm{EX}}_k\,\Delta t\,\mathbf{y}$
  $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \hat{b}^{\mathrm{IM}}_k\,\Delta t\,\mathbf{z} + \hat{b}^{\mathrm{EX}}_k\,\Delta t\,\mathbf{y}$
end
```

where $\mathbf{z}$ and $\mathbf{y}$ store the implicit and explicit parts of the RHS at each stage, $\mathbf{x}$ is used to advance the solution of the main scheme,[3] and $\hat{\mathbf{x}}$ stores the solution of the embedded scheme if adaptive time stepping is implemented. Note that one linear solve of the form $(I - c\,A)^{-1}\mathbf{b}$, one matrix/vector product $A\mathbf{y}$, and one nonlinear function evaluation $\mathbf{g}(\mathbf{y}, t)$ are computed per stage, in addition to various level-1 BLAS (basic linear algebra subroutine) operations. As discussed in Section 1.1, implementation in the case of a nonlinear stiff part is a straightforward extension.

*1.2.2. General two-register implementation of [2R] IMEXRK schemes*

By applying the matrix inversion lemma $(\hat{A} + \hat{B}\hat{C}\hat{D})^{-1} = \hat{A}^{-1} - \hat{A}^{-1}\hat{B}(\hat{C}^{-1} + \hat{D}\hat{A}^{-1}\hat{B})^{-1}\hat{D}\hat{A}^{-1}$ with $\hat{A} = \hat{C} = I$, $\hat{D} = A$, and $B = -a^{\mathrm{IM}}_{k,k}\,\Delta t$, the algorithm laid out in Section 1.2.1 may be rewritten in a form that only requires two registers[2] of length $N$:

```
for  $k = 1:s$
  if  $k == 1$,  $\mathbf{y} \leftarrow \mathbf{x}$,  else
    $\mathbf{y} \leftarrow \mathbf{x} + \big(a^{\mathrm{IM}}_{k,k-1} - b^{\mathrm{IM}}_{k-1}\big)\,\Delta t\,A\,\mathbf{y} + \big(a^{\mathrm{EX}}_{k,k-1} - b^{\mathrm{EX}}_{k-1}\big)\,\Delta t\,\mathbf{g}\big(\mathbf{y},\ t_n + c^{\mathrm{EX}}_{k-1}\Delta t\big)$
  end
  $\mathbf{y} \leftarrow \big(I - a^{\mathrm{IM}}_{k,k}\,\Delta t\,A\big)^{-1}\mathbf{y}$                                        (20)
  $\mathbf{x} \leftarrow \mathbf{x} + b^{\mathrm{IM}}_k\,\Delta t\,A\,\mathbf{y} + b^{\mathrm{EX}}_k\,\Delta t\,\mathbf{g}\big(\mathbf{y},\ t_n + c^{\mathrm{EX}}_k\Delta t\big)$
  $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \hat{b}^{\mathrm{IM}}_k\,\Delta t\,A\,\mathbf{y} + \hat{b}^{\mathrm{EX}}_k\,\Delta t\,\mathbf{g}\big(\mathbf{y},\ t_n + c^{\mathrm{EX}}_k\Delta t\big)$
end
```

In this case, one linear solve of the form $(I - c\,A)^{-1}\mathbf{b}$ and two operations of the form[4] $\mathbf{x} + c\,A\,\mathbf{y} + d\,\mathbf{g}(\mathbf{y}, t)$ are computed per stage, in addition to various level-1 BLAS operations. However, the storage requirement is reduced from three registers of length $N$ to only two, which is quite significant. In many cases, some of the coefficients in the above algorithm turn out to be zero, so the increased computational cost associated with the extra nonlinear function evaluations and matrix/vector products in this implementation is not as bad as one might initially anticipate, as quantified in Section 7.

*1.2.3. General four-register implementation of [3R] IMEXRK schemes*

For the development of the stage-order-two schemes IMEXRKCB3f and IMEXRKCB4 in Section 4 and Section 5, the [3R] IMEXRK structure (17) will be used to provide increased freedom during the design phase. Such schemes admit the following four-register implementation:

```
for  $k = 1:s$
  if  $k == 1$,  $\mathbf{y} \leftarrow \mathbf{x}$,  $\mathbf{z}^{\mathrm{IM}} = \mathbf{x}$,  $\mathbf{z}^{\mathrm{EX}} \leftarrow \mathbf{x}$,  else
    $\mathbf{z}^{\mathrm{EX}} \leftarrow \mathbf{y} + a^{\mathrm{EX}}_{k,k-1}\,\Delta t\,\mathbf{z}^{\mathrm{EX}}$
    if  $k < s$,  $\mathbf{y} \leftarrow \mathbf{x} + \big(a^{\mathrm{IM}}_{k+1,k-1} - b^{\mathrm{IM}}_{k-1}\big)\,\Delta t\,\mathbf{z}^{\mathrm{IM}} + \big(a^{\mathrm{EX}}_{k+1,k-1} - b^{\mathrm{EX}}_{k-1}\big)\,\big(\mathbf{z}^{\mathrm{EX}} - \mathbf{y}\big)/a^{\mathrm{EX}}_{k,k-1}$,  end
    $\mathbf{z}^{\mathrm{EX}} \leftarrow \mathbf{z}^{\mathrm{EX}} + a^{\mathrm{IM}}_{k,k-1}\,\Delta t\,\mathbf{z}^{\mathrm{IM}}$
  end                                                                                      (21)
  $\mathbf{z}^{\mathrm{IM}} = \big(I - a^{\mathrm{IM}}_{k,k}\,\Delta t\,A\big)^{-1} A\,\mathbf{z}^{\mathrm{EX}}$
  $\mathbf{z}^{\mathrm{EX}} \leftarrow \mathbf{g}\big(\mathbf{z}^{\mathrm{EX}} + a^{\mathrm{IM}}_{k,k}\,\Delta t\,\mathbf{z}^{\mathrm{IM}},\ t_n + c^{\mathrm{EX}}_k\Delta t\big)$
  $\mathbf{x} \leftarrow \mathbf{x} + b^{\mathrm{IM}}_k\,\Delta t\,\mathbf{z}^{\mathrm{IM}} + b^{\mathrm{EX}}_k\,\Delta t\,\mathbf{z}^{\mathrm{EX}}$
  $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \hat{b}^{\mathrm{IM}}_k\,\Delta t\,\mathbf{z}^{\mathrm{IM}} + \hat{b}^{\mathrm{EX}}_k\,\Delta t\,\mathbf{z}^{\mathrm{EX}}$
end
```

[3] Note again that $b^{\mathrm{IM}}_i = b^{\mathrm{EX}}_i = b_i$ for $i = 1, \ldots, s$ for the schemes developed herein, though this property is not shared by CN/RKW3 (see Section 2).

[4] When using finite-difference methods, an operation of this form can, with care, usually be performed *in place* in the computer memory using $O(1)$ temporary storage variables; how this is best accomplished, of course, depends on the precise form of $A$ and $\mathbf{g}(\mathbf{y}, t)$. When using spectral methods, such a two-register implementation is generally not available.

where $\mathbf{z}^{\mathrm{IM}}$ and $\mathbf{z}^{\mathrm{EX}}$ store the implicit and explicit parts of the RHS at each stage, $\mathbf{y}$ is a temporary variable which contributes to advance the solution to the next stage, $\mathbf{x}$ is used to advance the solution of the main scheme, and $\hat{\mathbf{x}}$ stores the solution of the embedded scheme if adaptive timestepping is used. As in the three-register implementation of the [2R] scheme, only one linear solve of the form $(I - c\,A)^{-1}\mathbf{b}$, one matrix/vector product, and one nonlinear function evaluation are computed per stage.

### 1.2.4. General three-register implementation of [3R] IMEXRK schemes

Leveraging matrix inversion lemma as done in Section 1.2.2, we obtain a general three-register implementation of any [3R] IMEXRK scheme:

```
for $k = 1:s$
  if $k == 1$,   $\mathbf{y} \leftarrow \mathbf{x}$,   $\mathbf{z} \leftarrow \mathbf{x}$,   else
    if $k < s$
      $\mathbf{z} \leftarrow \mathbf{y} + a^{\mathrm{IM}}_{k,k-1}\,\Delta t\,A\,\mathbf{z}$
      $\mathbf{y} \leftarrow A^{-1}\,(\mathbf{z} - \mathbf{y})/(a^{\mathrm{IM}}_{k,k-1}\,\Delta t)$
      $\mathbf{z} \leftarrow \mathbf{z} + a^{\mathrm{EX}}_{k,k-1}\,\Delta t\,\mathbf{g}(\mathbf{y},\,t_n + c^{\mathrm{EX}}_{k-1}\Delta t)$
      $\mathbf{y} \leftarrow \mathbf{x} + (a^{\mathrm{IM}}_{k+1,k-1} - b^{\mathrm{IM}}_{k-1})\,\Delta t\,A\,\mathbf{y} + (a^{\mathrm{EX}}_{k+1,k-1} - b^{\mathrm{EX}}_{k-1})\,\Delta t\,\mathbf{g}(\mathbf{y},\,t_n + c^{\mathrm{EX}}_{k-1}\Delta t)$
    else                                                                                  (22)
      $\mathbf{z} \leftarrow \mathbf{y} + a^{\mathrm{IM}}_{k,k-1}\,\Delta t\,A\,\mathbf{z} + a^{\mathrm{EX}}_{k,k-1}\,\Delta t\,\mathbf{g}(\mathbf{y},\,t_n + c^{\mathrm{EX}}_{k-1}\Delta t)$
    end
  end
  $\mathbf{z} \leftarrow (I - a^{\mathrm{IM}}_{k,k}\,\Delta t\,A)^{-1}\,\mathbf{z}$
  $\mathbf{x} \leftarrow \mathbf{x} + b^{\mathrm{IM}}_{k}\,\Delta t\,A\,\mathbf{z} + b^{\mathrm{EX}}_{k}\,\Delta t\,\mathbf{g}(\mathbf{z},\,t_n + c^{\mathrm{EX}}_{k}\Delta t)$
  $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \hat{b}^{\mathrm{IM}}_{k}\,\Delta t\,A\,\mathbf{z} + \hat{b}^{\mathrm{EX}}_{k}\,\Delta t\,\mathbf{g}(\mathbf{z},\,t_n + c^{\mathrm{EX}}_{k}\Delta t)$
end
```

Note that this algorithm requires the invertibility of the matrix $A$, a condition that is often true when $A$ arises from the discretization of a PDE. In this case, two linear systems, three matrix/vector products, and three nonlinear function evaluations must be performed per stage (except for the last stage), plus an additional matrix/vector product and one nonlinear function evaluation if the embedded scheme is used for adaptive timestepping.

Finally, note that a (hardware-dependent) trade-off between flops and storage must ultimately be conducted to select between the two-register and three-register implementation of any [2R] scheme, or between the three-register and four-register implementation of any [3R] scheme.

## 2. Two second-order, 2-register IMEXRK schemes

The classical second-order, $A$-stable **CN/RKW3** method may easily be written in the low-storage IMEXRK Butcher tableaux form (16) (albeit with the $b^{\mathrm{IM}}_i = b^{\mathrm{EX}}_i = b_i$ constraint relaxed) with the four-stage IMEX Butcher tableaux

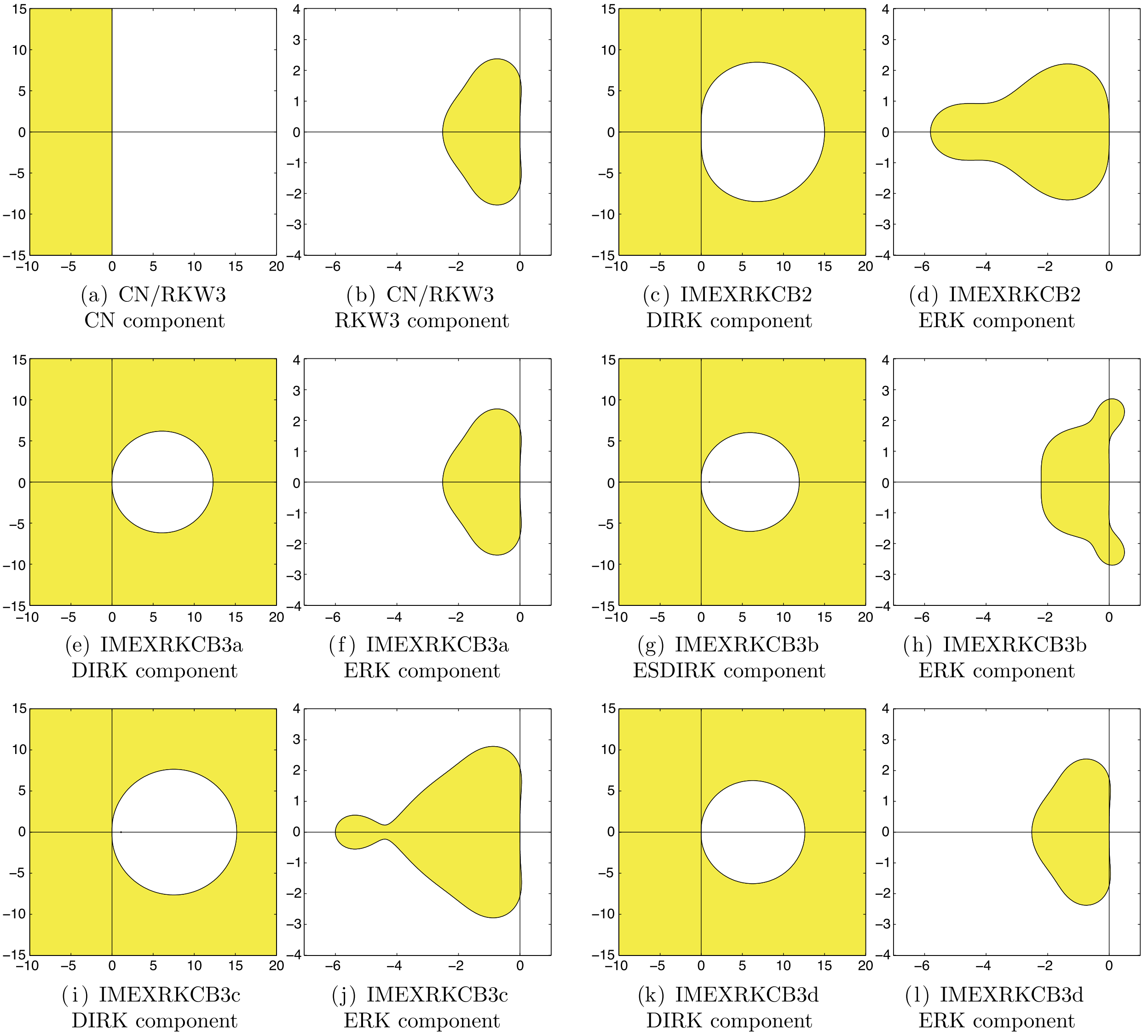
$$\boxed{\begin{array}{r|llll} 0 & 0 & & & \\ 8/15 & 4/15 & 4/15 & & \\ 2/3 & 4/15 & 1/3 & 1/15 & \\ 1 & 4/15 & 1/3 & 7/30 & 1/6 \\ \hline & 4/15 & 1/3 & 7/30 & 1/6 \end{array} \qquad \begin{array}{r|llll} 0 & 0 & & & \\ 8/15 & 8/15 & 0 & & \\ 2/3 & 1/4 & 5/12 & 0 & \\ 1 & 1/4 & 0 & 3/4 & 0 \\ \hline & 1/4 & 0 & 3/4 & 0 \end{array}} \tag{23}$$

A DIRK scheme with $c_1 = 0$ and $c_s = 1$ [such as that shown at left in (23)] is known as a first-same-as-last (FSAL) scheme. In such a scheme, the implicit part of the last stage of one timestep is precisely the implicit part of the first stage of the next timestep, and thus an FSAL scheme, such as the implicit part of the CN/RKW3 scheme shown above, actually incorporates only $s - 1$ implicit solves per timestep. Note also that, since $b^{\mathrm{EX}}_s = 0$ above, $\mathbf{g}_s$ actually never needs to be computed. Thus, though CN/RKW3 is written above as a four-stage IMEX Butcher tableaux, a careful implementation of CN/RKW3 actually incorporates only three implicit stages and three explicit stages per timestep.

The stability boundaries of the constituent CN and RKW3 schemes of (23) are shown in Figs. 1(a)–1(b); the CN scheme, applied over each of three stages, is $A$ stable, and the stability of the RKW3 scheme is that of any third-order, three-stage ERK scheme, with (denoting $z = z^{\mathrm{EX}}$) a stability function of

$$\sigma^{\mathrm{EX}}(z) = 1 + z\sum_{i=1}^{4} b_i + z^2\sum_{i=1}^{4} b_i\,c_i + z^3\sum_{i,j=1}^{4} b_i\,a^{\mathrm{EX}}_{ij}\,c_j + z^4\sum_{i,j,k=1}^{4} b_i\,a^{\mathrm{EX}}_{ij}\,a^{\mathrm{EX}}_{jk}\,c_k = 1 + z + z^2/2 + z^3/6,$$

where, again, $|\sigma^{\mathrm{EX}}(z)| \le 1$ indicates the stability region.



(a) CN/RKW3 CN component

(b) CN/RKW3 RKW3 component

(c) IMEXRKCB2 DIRK component

(d) IMEXRKCB2 ERK component

(e) IMEXRKCB3a DIRK component

(f) IMEXRKCB3a ERK component

(g) IMEXRKCB3b ESDIRK component

(h) IMEXRKCB3b ERK component

(i) IMEXRKCB3c DIRK component

(j) IMEXRKCB3c ERK component

(k) IMEXRKCB3d DIRK component

(l) IMEXRKCB3d ERK component

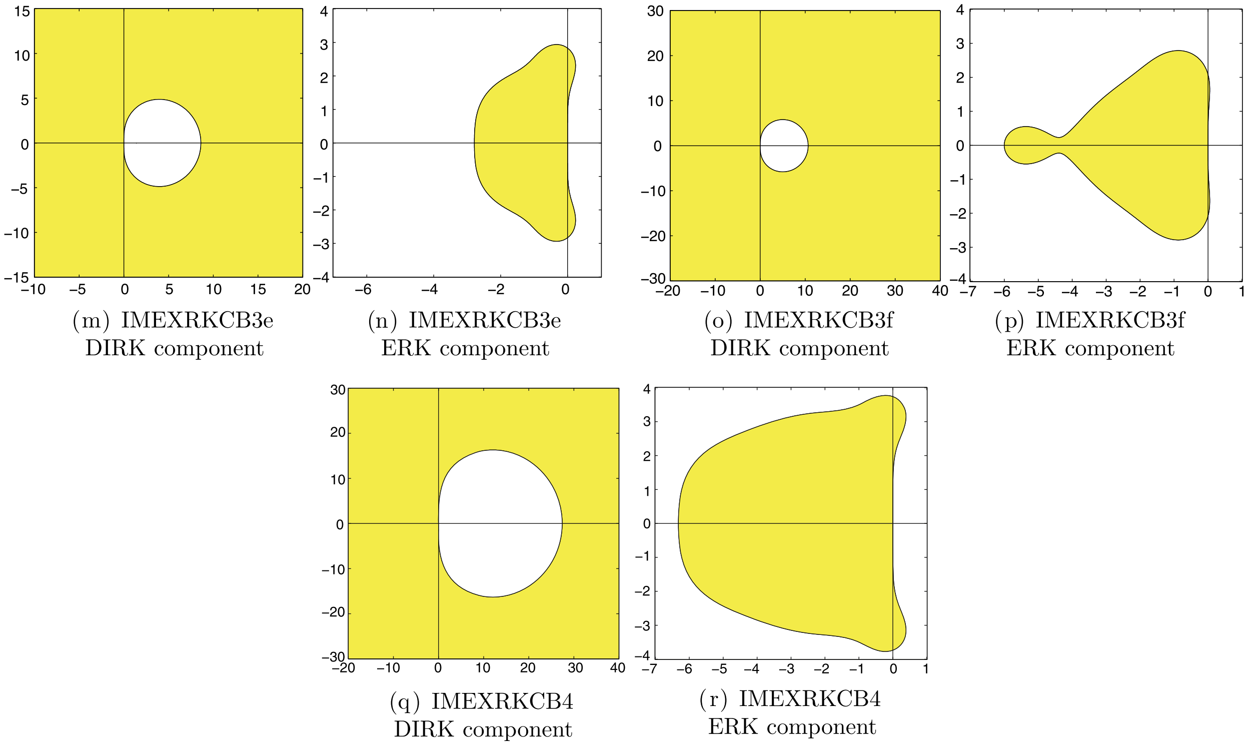
**Fig. 1.** Stability regions $|\sigma(z)| \leq 1$ for the low-storage IMEXRK schemes considered in this paper.

The CN/RKW3 scheme was initially developed simply by joining together two existing schemes, CN and RKW3, in an IMEXRK fashion. It was, e.g., not designed with the constraints (4i)–(4n) in mind, and thus leaves significant room for improvement. For example, a remarkably simple second-order [2R] alternative to CN/RKW3 which

(a) requires fewer flops per timestep to implement than CN/RKW3,
(b) comes with a first-order embedded scheme, following the guidelines listed in Section 1.2, for adaptive timestepping,
(c) whose implicit part is $L$-stable, and
(d) whose explicit part is both SSP and exhibits much improved stability on the negative real axis as compared to CN/RKW3,

dubbed **IMEXRKCB2**, is given by[5]

[5] For details on how this scheme was discovered, see Section 3.3, which applies the same techniques used to discover (24) to the third-order, 3-stage implicit, 4-stage explicit, $L$-stable case.



(m) IMEXRKCB3e DIRK component

(n) IMEXRKCB3e ERK component

(o) IMEXRKCB3f DIRK component

(p) IMEXRKCB3f ERK component

(q) IMEXRKCB4 DIRK component

(r) IMEXRKCB4 ERK component

**Fig. 1.** (*continued*)

$$
\boxed{
\begin{array}{c|ccc}
0 & 0 & & \\
2/5 & 0 & 2/5 & \\
1 & 0 & 5/6 & 1/6 \\
\hline
 & 0 & 5/6 & 1/6 \\
\hline
 & 0 & 4/5 & 1/5
\end{array}
\qquad
\begin{array}{c|ccc}
0 & 0 & & \\
2/5 & 2/5 & 0 & \\
1 & 0 & 1 & 0 \\
\hline
 & 0 & 5/6 & 1/6 \\
\hline
 & 0 & 4/5 & 1/5
\end{array}
}
\tag{24}
$$

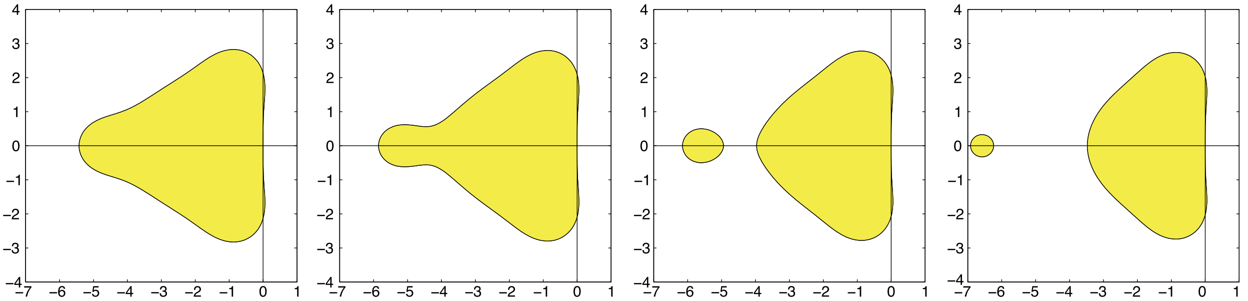
The coefficient for strong stability in (10) for this scheme is $c = 1$, which is the maximum possible, as proved in [6]. Note also that the so-called “stiff accuracy” conditions have been imposed on the implicit component of this scheme; that is, we have set $a^{\mathrm{IM}}_{s,i} = b_i$ for $i = 1, \ldots, s$. These conditions improve the convergence of such a scheme for the integration of stiff ODEs, as noted in [7] and [8] and described further in Section 6. Moreover, these conditions have the benefit of reducing by one the order of the polynomial in the numerator of the stability function, facilitating the attainment of $L$-stability [i.e., $\sigma(\infty) = 0$], as we will show in Section 3.3. Applying the stiff accuracy conditions to (4a) and (15), we obtain $c_s = 1$. Together with the condition $c_1 = 0$, it follows that all IMEX schemes developed herein with DIRK components achieving $L$-stability via the stiff accuracy conditions, such as (24), are FSAL, and thus require only $s - 1$ implicit solves per timestep. This is especially apparent in (24), in which the entire first column of the Butcher tableau of the implicit component equals zero. Since this IMEXRK scheme has two implicit stages and three explicit stages per timestep, as a shorthand, we report the scheme as requiring $(2, 3)$ stages per timestep in Table 1; the stage requirements of the other schemes developed in this paper are denoted similarly.

The stability boundaries of the constituent DIRK and ERK components of (24) are shown in Figs. 1(c)–1(d).

## 3. Five third-order, 2-register IMEXRK schemes

### 3.1. *A* $(2, 3)$*-stage, strongly* $A$*-stable scheme*

As suggested by (24), to streamline the implementation, we can suppress the first stage of the DIRK scheme by imposing $b_1 = a^{\mathrm{IM}}_{21} = 0$. Following this simplification, the entire first column of the DIRK scheme is zero, thus leading to a scheme with $s - 1$ implicit stages and $s$ explicit stages. In the $s = 3$ case, the IMEXRK Butcher tableaux take the general form



(a) $\delta = 1/50$ (b) $\delta = 1/53$ (c) $\delta = 1/55$ (d) $\delta = 1/60$

**Fig. 2.** Stability regions $|\sigma(z)| \le 1$ for $\sigma(z) = 1 + z + z^2/2 + z^3/6 + \delta z^4$ for various values of $\delta$; note that the case with $\delta = 1/24$ is given in Fig. 1(l), and the case with $\delta = 1/54$ is given in Fig. 1(j).

$$\begin{array}{c|ccc} 0 & 0 & & \\ c_2 & 0 & a_{22}^{\mathrm{IM}} & \\ c_3 & 0 & a_{32}^{\mathrm{IM}} & a_{33}^{\mathrm{IM}} \\ \hline & 0 & b_2 & b_3 \end{array} \qquad \begin{array}{c|ccc} 0 & 0 & & \\ c_2 & a_{21}^{\mathrm{EX}} & 0 & \\ c_3 & 0 & a_{32}^{\mathrm{EX}} & 0 \\ \hline & 0 & b_2 & b_3 \end{array} \tag{25a}$$

To achieve third-order accuracy, after imposing stage-order-one conditions on both implicit and explicit part, we arrive at five nonlinear equations in five parameters:

$$b_2 + b_3 - 1 = 0, \qquad b_2 c_2 + b_3 c_3 - 1/2 = 0, \qquad b_2 c_2^2 + b_3 c_3^2 - 1/3 = 0, \qquad b_3 c_2 c_3 - 1/6 = 0,$$
$$b_2 c_2^2 + b_3 a_{33}^{\mathrm{IM}} c_3 + b_3 \big(c_3 - a_{33}^{\mathrm{IM}}\big) c_2 - 1/6 = 0.$$

This system of nonlinear equations has a single closed-form solution among the real numbers. Defining $c_2$ as the sole real root of the polynomial $18c_2^3 - 27c_2^2 + 12c_2 - 2 = 0$, closed-form expressions for the parameters of this scheme, dubbed **IMEXRKCB3a**, are:

$$\begin{aligned} &c_2 = a_{22}^{\mathrm{IM}} = a_{21}^{\mathrm{EX}} = \big(27 + \sqrt[3]{2187 - 1458\sqrt{2}} + 9\sqrt[3]{3 + 2\sqrt{2}}\big)/54, \\ &c_3 = a_{32}^{\mathrm{EX}} = c_2 / \big(6c_2^2 - 3c_2 + 1\big), \qquad b_2 = (3c_2 - 1)/\big(6c_2^2\big), \qquad b_3 = \big(6c_2^2 - 3c_2 + 1\big)/\big(6c_2^2\big), \\ &a_{33}^{\mathrm{IM}} = \big(1/6 - b_2 c_2^2 - b_3 c_2 c_3\big) / \big[b_3 (c_3 - c_2)\big], \qquad a_{32}^{\mathrm{IM}} = a_{33}^{\mathrm{IM}} - c_3. \end{aligned} \tag{25b}$$

The stability boundaries of the constituent DIRK and ERK components of (25) are shown in Figs. 1(e)–1(f); note that the stability boundary of the 3-stage, third-order ERK component necessarily coincides with that of RKW3. As compared with (24), which has a Butcher tableaux of the same structure, the present scheme sacrifices *L*-stability of its DIRK component in order to achieve third-order accuracy.

It is instructive to note that, even after removing the assumption $b_1 = 0$, it is not possible to achieve *L*-stability of the DIRK component of a third-order IMEXRK scheme of the general form given in (16) using only three stages due to a conflict that arises in the $\tau^{\mathrm{IMEX}(3)} = 0$ constraints (4j)–(4n), as observed previously by [2]. For this reason, the remainder of this paper explores four-stage schemes of an analogous form for third-order accuracy.

### 3.2. *A* (3, 4)*-stage, strongly A-stable scheme with ESDIRK implicit part*

Extending the simplifying assumptions used in the previous section to a four-stage two-register scheme, by taking $b_1 = b_2 = 0$, and additionally imposing equal values for the diagonal terms of the implicit scheme (that is, $a_{i,i}^{\mathrm{IM}} = \gamma$ for $i = 2, 3, 4$), the Butcher tableaux (16) reduce to:

$$\begin{array}{c|cccc} 0 & 0 & & & \\ c_2 & 0 & \gamma & & \\ c_3 & 0 & a_{32}^{\mathrm{IM}} & \gamma & \\ c_4 & 0 & 0 & a_{43}^{\mathrm{IM}} & \gamma \\ \hline & 0 & 0 & b_3 & b_4 \end{array} \qquad \begin{array}{c|cccc} 0 & 0 & & & \\ c_2 & a_{21}^{\mathrm{EX}} & 0 & & \\ c_3 & 0 & a_{32}^{\mathrm{EX}} & 0 & \\ c_4 & 0 & 0 & a_{43}^{\mathrm{EX}} & 0 \\ \hline & 0 & 0 & b_3 & b_4 \end{array} \tag{26a}$$

After imposing stage-order-one conditions, determining all the parameters requires the solution of the following system of five nonlinear equations:

$$b_3 + b_4 - 1 = 0, \qquad b_3 c_3 + b_4 c_4 - 1/2 = 0, \qquad b_3 c_3^2 + b_4 c_4^2 - 1/3 = 0, \qquad b_3 c_2 c_3 + b_4 c_3 c_4 - 1/6 = 0,$$
$$b_3 c_2 c_3 + b_3 c_2 c_3 - b_3 c_2^2 + b_4 c_2 c_4 + b_4 c_3 c_4 - b_4 c_2 c_3 - 1/6 = 0.$$

This system of equations has two closed-form solutions, one of which does not lead to an $A$-stable scheme, and the other of which, dubbed **IMEXRKCB3b**, is given by

$$\gamma = c_2 = a_{21}^{\mathrm{EX}} = 1/2 + \sqrt{3}/6, \qquad c_3 = a_{32}^{\mathrm{EX}} = 1/2 - \sqrt{3}/6, \qquad c_4 = a_{43}^{\mathrm{EX}} = 1/2 + \sqrt{3}/6,$$
$$a_{32}^{\mathrm{IM}} = -\sqrt{3}/3, \qquad a_{43}^{\mathrm{IM}} = 0, \qquad b_3 = b_4 = 1/2. \tag{26b}$$

The stability boundaries of the constituent DIRK and ERK components of (26) are shown in Figs. 1(g)–1(h). This scheme again achieves strong $A$-stability of its DIRK component while, as compared with IMEXRKCB3a, slightly extending the limit of stability of the ERK component in the imaginary directions, and slightly reducing the limit of stability of the ERK component in the negative real direction.

Imposing the nonzero diagonal terms of the DIRK scheme to be equal [a simplification resulting in what is usually called an Explicit-first-stage Singly Diagonally Implicit Runge–Kutta (ESDIRK) method] facilitates use of the LU decomposition of the matrix $(I - c_2 \Delta t A)$ to simplify all of the implicit solves. This can significantly reduce the number of flops needed for the implicit solves, but may increase the number of registers required by the code; whether or not use of the LU decomposition in the implicit solves represents an overall speedup of the simulation depends both on the structure and size of $A$ and the computational hardware being used.

### 3.3. Three (3, 4)-stage, $L$-stable schemes

The simplifying assumptions considered in the previous section again facilitated a closed-form expression of the parameters, but prevented the DIRK component from achieving $L$-stability. In order to achieve $L$-stability of the DIRK component, as noted previously, a useful simplifying assumption is the "stiff accuracy" conditions $a_{s,i} = b_i$ for $i = 1, \dots, s$ [and hence, by (4a) and (15), $c_s = 1$]. Taking $s = 4$ and defining $a_{i,i}^{\mathrm{IM}} = \alpha_i$ for $i = 2, 3$, the Butcher tableaux (16) reduce to the following form (with, again, an FSAL implicit part):

$$\begin{array}{c|cccc}
0 & 0 & & & \\
c_2 & a_{21}^{\mathrm{IM}} & a_{22}^{\mathrm{IM}} & & \\
c_3 & b_1 & a_{32}^{\mathrm{IM}} & a_{33}^{\mathrm{IM}} & \\
1 & b_1 & b_2 & b_3 & b_4 \\
\hline
 & b_1 & b_2 & b_3 & b_4 \\
\hline
 & \hat{b}_1^{\mathrm{IM}} & \hat{b}_2^{\mathrm{IM}} & \hat{b}_3^{\mathrm{IM}} & \hat{b}_4^{\mathrm{IM}}
\end{array}
\qquad
\begin{array}{c|cccc}
0 & 0 & & & \\
c_2 & a_{21}^{\mathrm{EX}} & 0 & & \\
c_3 & b_1 & a_{32}^{\mathrm{EX}} & 0 & \\
1 & b_1 & b_2 & a_{43}^{\mathrm{EX}} & 0 \\
\hline
 & b_1 & b_2 & b_3 & b_4 \\
\hline
 & \hat{b}_1^{\mathrm{EX}} & \hat{b}_2^{\mathrm{EX}} & \hat{b}_3^{\mathrm{EX}} & \hat{b}_4^{\mathrm{EX}}
\end{array} \tag{27}$$

In order to impose third-order accuracy, five order constraints must again be imposed. To achieve $L$-stability of the DIRK component, a further simplification of (27) is motivated. To understand this simplification, we may rewrite the stability function of the scheme as a rational function of $z^{\mathrm{IM}}$ and $z^{\mathrm{EX}}$, as suggested by (5) and (18), as

$$\sigma\left(z^{\mathrm{IM}}; z^{\mathrm{EX}}\right) = \frac{1 + \sum_{i=1}^{2} p_i(z^{\mathrm{EX}})\,[z^{\mathrm{IM}}]^i + (\hat{p}_{30} + \hat{p}_{31} z^{\mathrm{EX}})\,[z^{\mathrm{IM}}]^3 + \hat{p}_{40}\,[z^{\mathrm{IM}}]^4}{1 + \sum_{i=1}^{s-1} q_i\,[z^{\mathrm{IM}}]^i},$$

where the $p_i$, $\hat{p}_{ij}$, and $q_i$ are polynomials in the Butcher tableaux parameters. Due to stiff accuracy, $\hat{p}_{40} = 0$; thus, in order to impose $L$-stability of the DIRK component [i.e., $\lim_{z^{\mathrm{IM}} \to \infty} \sigma(z^{\mathrm{IM}}; z^{\mathrm{EX}}) = 0$], it is sufficient to impose that $q_3 = a_{22}^{\mathrm{IM}} a_{33}^{\mathrm{IM}} b_4 \neq 0$ and

$$\begin{aligned}\tau_1^{L\text{-stability}} = \hat{p}_{30} = {} & -a_{22}^{\mathrm{IM}} a_{33}^{\mathrm{IM}} b_1 - a_{22}^{\mathrm{IM}} a_{33}^{\mathrm{IM}} b_2 - a_{22}^{\mathrm{IM}} a_{33}^{\mathrm{IM}} b_3 + a_{33}^{\mathrm{IM}} b_2 c_2 \\ & + a_{33}^{\mathrm{IM}} b_3 c_2 + b_1 b_3 c_2 + a_{22}^{\mathrm{IM}} b_3 c_3 - b_3 c_2 c_3 = 0,\end{aligned} \tag{A}$$

$$\begin{aligned}\tau_2^{L\text{-stability}} = \hat{p}_{31} = {} & -a_{22}^{\mathrm{IM}} a_{33}^{\mathrm{IM}} b_4 + a_{33}^{\mathrm{IM}} b_4 c_2 + b_1 b_4 c_2 - a_{33}^{\mathrm{IM}} b_1 b_4 c_2 - b_1^2 b_4 c_2 - b_1 b_2 b_4 c_2 + a_{22}^{\mathrm{IM}} b_4 c_3 \\ & - a_{22}^{\mathrm{IM}} b_1 b_4 c_3 - a_{22}^{\mathrm{IM}} b_2 b_4 c_3 - b_4 c_2 c_3 + b_1 b_4 c_2 c_3 + b_2 b_4 c_2 c_3 = 0.\end{aligned} \tag{B}$$

As noted in [8] and [9], suppressing the first column of the DIRK component, by imposing $b_1 = 0 = a_{21}^{\mathrm{IM}} = 0$ in (27), together with stiff-accuracy condition, satisfies both (A) and (B) identically; we thus incorporate these additional simplifications in the two subsections that follow. Notice that in the full-storage setting this strategy is *not* recommended, as it sacrifices $s - 1$ degrees of freedom. For a [2R] scheme, however, only two degrees of freedom are sacrificed to enforce these two equations, and thereby gain $L$-stability.

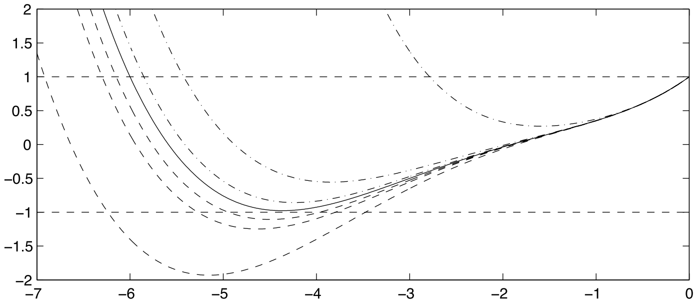


**Fig. 3.** The (real) value of $\sigma(z) = 1 + z + z^2/2 + z^3/6 + \delta z^4$ for real, negative values of $z$ and various values of $\delta$: (dashed) $\delta = 1/60$, $1/56$, $1/55$; (solid) $\delta = 1/54$; (dot-dashed) $\delta = 1/53$, $1/50$, $1/24$. See also Fig. 2.

### 3.3.1. *Maximizing the extent of stability of the ERK component over the negative real axis*

A final (sixth) constraint is obtained by maximizing the stability envelope of the ERK component over the negative real axis. Using Cramer's rule, we may rewrite the stability function of the third-order, four-stage ERK component as

$$\sigma^{\mathrm{EX}}(z;\delta) = 1 + z\,\mathbf{b}^T\left(I - z\,A^{\mathrm{EX}}\right)^{-1}\mathbf{1} = 1 + z + z^2/2 + z^3/6 + \delta\,z^4 \quad \text{where } \delta = \sum_{i,j,k=1}^{4} b_i\,a_{ij}^{\mathrm{EX}}\,a_{jk}^{\mathrm{EX}}\,c_k.$$

For $z$ on the negative real axis, the stability region $|\sigma^{\mathrm{EX}}(z;\delta)| \le 1$ is defined by the two conditions

$$-1 \le 1 + z + z^2/2 + z^3/6 + \delta\,z^4 \le 1.$$

Plots of $\sigma^{\mathrm{EX}}(z;\delta)$ for $-7 \le z \le 0$ and various values of $\delta$ are given in Fig. 3. For

$$\delta > \delta_{\mathrm{crit}} = \left(139 - 5255/\sqrt[3]{-210\,253 + 60\,928\sqrt{51}} + \sqrt[3]{-210\,253 + 60\,928\sqrt{51}}\right)/6144 = 0.0184557,$$

the condition $-1 \le \sigma^{\mathrm{EX}}(z;\delta)$ is satisfied everywhere in this interval; we thus choose $\delta = 1/54 = 0.0185 > \delta_{\mathrm{crit}}$, which gives $|\sigma^{\mathrm{EX}}(z)| \le 1$ for $-6.00 < z < 0$, as larger values of $\delta$ reduce the extent of stability (see Figs. 2 and 3).

Parametric variation reveals that the extent of the stability region along the imaginary axis is relatively insensitive to changes in $\delta$. Among the third-order, four-stage IMEXRK scheme available in literature, the one with the widest stability region of the ERK part, which is the (full-storage) ARK3(2)4L[2R]SA scheme developed in [9], has a maximum extent along the negative real axis which is ~40% *less* than that of that of the present scheme, and a maximum extent along the imaginary axis which is only ~5% greater than that of the present scheme; the stability characteristics of the present scheme are thus seen to be quite competitive.

Thus, in order to determine the parameters of the Butcher tableaux, we impose our final (sixth) constraint as

$$\tau^{\,\delta=1/54} = \sum_{i,j,k=1}^{4} b_i\,a_{ij}^{\mathrm{EX}}\,a_{jk}^{\mathrm{EX}}\,c_k - 1/54 = 0. \tag{C}$$

The complete solution of this set of six nonlinear constraint equations has been obtained using Mathematica [21]. The scheme associated to such solution, dubbed **IMEXRKCB3c**, is given by (27) with

$$\begin{aligned}
&a_{22}^{\mathrm{IM}} = \frac{3\,375\,509\,829\,940}{4\,525\,919\,076\,317}, \quad a_{32}^{\mathrm{IM}} = -\frac{11\,712\,383\,888\,607\,531\,889\,907}{32\,694\,570\,495\,602\,105\,556\,248}, \quad a_{33}^{\mathrm{IM}} = \frac{566\,138\,307\,881}{912\,153\,721\,139},\\
&b_1 = 0, \quad b_2 = \frac{673\,488\,652\,607}{2\,334\,033\,219\,546}, \quad b_3 = \frac{493\,801\,219\,040}{853\,653\,026\,979}, \quad b_4 = \frac{184\,814\,777\,513}{1\,389\,668\,723\,319},\\
&c_2 = a_{21}^{\mathrm{EX}} = \frac{3\,375\,509\,829\,940}{4\,525\,919\,076\,317}, \quad c_3 = a_{32}^{\mathrm{EX}} = \frac{272\,778\,623\,835}{1\,039\,454\,778\,728}, \quad a_{43}^{\mathrm{IM}} = \frac{1\,660\,544\,566\,939}{2\,334\,033\,219\,546};
\end{aligned} \tag{28a}$$

the associated second-order embedded scheme has the following coefficients:

$$\begin{aligned}
&\hat{b}_1^{\mathrm{IM}} = 0, \quad \hat{b}_2^{\mathrm{IM}} = \frac{366\,319\,659\,506}{1\,093\,160\,237\,145}, \quad \hat{b}_3^{\mathrm{IM}} = \frac{270\,096\,253\,287}{480\,244\,073\,137}, \quad \hat{b}_4^{\mathrm{IM}} = \frac{104\,228\,367\,309}{1\,017\,021\,570\,740},\\
&\hat{b}_1^{\mathrm{EX}} = \frac{449\,556\,814\,708}{1\,155\,810\,555\,193}, \quad \hat{b}_2^{\mathrm{EX}} = 0, \quad \hat{b}_3^{\mathrm{EX}} = \frac{210\,901\,428\,686}{1\,400\,818\,478\,499}, \quad \hat{b}_4^{\mathrm{EX}} = \frac{480\,175\,564\,215}{1\,042\,748\,212\,601}.
\end{aligned} \tag{28b}$$

The stability boundaries of the constituent DIRK and ERK components are shown in Figs. 1(i)–1(j). This scheme is SSP under the condition (10) with $c = 0.7027915$. This result can be improved up to $c = 0.7703947$, which is achieved by replacing condition (C) with

$$\tau^{\delta=0} = \sum_{i,j,k=1}^{4} b_i \, a_{ij}^{\mathrm{EX}} \, a_{jk}^{\mathrm{EX}} \, c_k - 0 = 0. \tag{C$'$}$$

This constraint does not lead to an IMEXRK scheme with an $L$-stable implicit component; we thus instead choose a small positive $\delta$, thus ensuring $L$-stability and a nearly optimal value $c$ for strong stability. Choosing $\delta = 1/10\,000$ results in a scheme, dubbed **IMEXRKCB3d**, given by (27) with

$$\begin{aligned}
&a_{22}^{\mathrm{IM}} = \frac{418\,884\,414\,754}{469\,594\,081\,263}, \qquad a_{32}^{\mathrm{IM}} = -\frac{304\,881\,946\,513\,433\,262\,434\,901}{718\,520\,734\,375\,438\,559\,540\,570}, \qquad a_{33}^{\mathrm{IM}} = \frac{684\,872\,032\,315}{962\,089\,110\,311}, \\
&b_1 = 0, \qquad b_2 = \frac{355\,931\,813\,527}{1\,014\,712\,533\,305}, \qquad b_3 = \frac{709\,215\,176\,366}{1\,093\,407\,543\,385}, \qquad b_4 = \frac{755\,675\,305}{1\,258\,355\,728\,177}, \\
&c_2 = a_{21}^{\mathrm{EX}} = \frac{418\,884\,414\,754}{469\,594\,081\,263}, \qquad c_3 = a_{32}^{\mathrm{EX}} = \frac{214\,744\,852\,859}{746\,833\,870\,870}, \qquad a_{43}^{\mathrm{EX}} = \frac{658\,780\,719\,778}{1\,014\,712\,533\,305};
\end{aligned} \tag{29a}$$

the associated second-order embedded scheme has the following coefficients:

$$\begin{aligned}
&\hat{b}_1^{\mathrm{IM}} = 0, \qquad \hat{b}_2^{\mathrm{IM}} = \frac{226\,763\,370\,689}{646\,029\,759\,300}, \qquad \hat{b}_3^{\mathrm{IM}} = \frac{1\,496\,839\,794\,860}{2\,307\,829\,317\,197}, \qquad \hat{b}_4^{\mathrm{IM}} = \frac{353\,416\,193}{889\,746\,336\,234}, \\
&\hat{b}_1^{\mathrm{EX}} = \frac{1\,226\,988\,580\,973}{2\,455\,716\,303\,853}, \qquad \hat{b}_2^{\mathrm{EX}} = 0, \qquad \hat{b}_3^{\mathrm{EX}} = \frac{827\,818\,615}{1\,665\,592\,077\,861}, \qquad \hat{b}_4^{\mathrm{EX}} = \frac{317\,137\,569\,431}{634\,456\,480\,332}.
\end{aligned} \tag{29b}$$

The coefficient for strong stability in this case is $c = 0.7701444$. The stability boundaries of the associated DIRK and ERK components are shown in Figs. 1(k)–1(l). Since $\delta$ is chosen close to zero, the stability region of the ERK component closely resembles that of a third-order three-stage explicit Runge–Kutta scheme.

### 3.3.2. *Maximizing accuracy of the ERK component*

An alternative third-order four-stage 2-register $L$-stable strategy, with closed-form parameter values and improved accuracy, is given by replacing the final constraint, (C), with

$$\tau^{\delta=1/24} = \sum_{i,j,k=1}^{4} b_i a_{ij}^{\mathrm{EX}} a_{jk}^{\mathrm{EX}} c_k - 1/24 = 0, \tag{C$''$}$$

which sets to zero one of the fourth-order truncation terms for the explicit component. This results in a scheme, dubbed **IMEXRKCB3e**, given by

$$\boxed{\begin{array}{r|llll}
0 & 0 & & & \\
1/3 & 0 & 1/3 & & \\
1 & 0 & 1/2 & 1/2 & \\
1 & 0 & 3/4 & -1/4 & 1/2 \\
\hline
 & 0 & 3/4 & -1/4 & 1/2
\end{array}
\qquad
\begin{array}{r|llll}
0 & 0 & & & \\
1/3 & 1/3 & 0 & & \\
1 & 0 & 1 & 0 & \\
1 & 0 & 3/4 & 1/4 & 0 \\
\hline
 & 0 & 3/4 & -1/4 & 1/2
\end{array}} \tag{30}$$

A second-order embedded scheme having all third-order truncation terms nonzero could not be achieved because of assumption (C$''$). The stability boundaries of the constituent DIRK and ERK components are shown in Figs. 1(m)–1(n); IMEXRKCB3e has improved accuracy but reduced stability on the negative real axis for the ERK component, as compared with IMEXRKCB3c. In particular, because of (C$''$), the stability region for the ERK part coincides with the stability region of a standard 4-stage fourth-order explicit RK scheme.

## 4. A third-order, 3-register, 4-stage, *L*-stable scheme

All of the schemes so-far described have stage-order one for both the implicit and explicit components. It is well known in the literature (see [7]) that this limits the order of convergence of such methods when applied to stiff ODEs. In particular, if the stiffness is so high that the ODE turns into an index-1 DAE, some variables convert from differential to algebraic and their convergence rate is determined by the stage-order of the method. For this reason, an attempt has been made to improve the stage-order of the implicit scheme, as done in [9]. In this way, when the DIRK scheme is employed alone, a better convergence will be observed during the integration of a stiff ODE, as we will show in Section 6.

Hence, after imposing the same $b_i$ and $c_i$ over the explicit and implicit components and stiff accuracy for the implicit component as done previously, we impose the stage-order-two condition for the implicit component, that is:

$$\sum_{j=1}^{s} a_{ij}^{\mathrm{IM}} c_j = c_i^2/2, \quad i = 2,\, 3,\, \dots,\, s-1. \tag{31}$$

With these constraints, $\tau_2^{\mathrm{IM}(3)} = 0$ in (4d) is automatically satisfied. Hence, we must only impose four constraints for third-order accuracy, two for $L$-stability, $2(s-2)$ constraints for stage-order two for the implicit component, and $(s-1)$ constraints for stage-order one for the explicit component. We also impose $c_1 = 0$ and $c_4 = 1$ for FSAL structure. Considering a four-stage three-register scheme,

$$\boxed{\begin{array}{c|cccc} 0 & 0 & & & \\ c_2 & a_{21}^{\mathrm{IM}} & a_{22}^{\mathrm{IM}} & & \\ c_3 & a_{31}^{\mathrm{IM}} & a_{32}^{\mathrm{IM}} & a_{33}^{\mathrm{IM}} & \\ 1 & b_1 & b_2 & b_3 & b_4 \\ \hline & b_1 & b_2 & b_3 & b_4 \\ \hline & \hat{b}_1^{\mathrm{IM}} & \hat{b}_2^{\mathrm{IM}} & \hat{b}_3^{\mathrm{IM}} & \hat{b}_4^{\mathrm{IM}} \end{array} \qquad \begin{array}{c|cccc} 0 & 0 & & & \\ c_2 & a_{21}^{\mathrm{EX}} & 0 & & \\ c_3 & a_{31}^{\mathrm{EX}} & a_{32}^{\mathrm{EX}} & 0 & \\ 1 & b_1 & a_{42}^{\mathrm{EX}} & a_{43}^{\mathrm{EX}} & 0 \\ \hline & b_1 & b_2 & b_3 & b_4 \\ \hline & \hat{b}_1^{\mathrm{EX}} & \hat{b}_2^{\mathrm{EX}} & \hat{b}_3^{\mathrm{EX}} & \hat{b}_4^{\mathrm{EX}} \end{array}} \tag{32a}$$

after these constraints are imposed, we are left with three degrees of freedom. We choose the constraint (C) to maximize the extent of the stability region of the explicit component on the negative real axis, and perform a parametric variation over the coefficients $c_2$ and $c_3$, the remaining two degrees of freedom, between 0 and 1 in order to identify an IMEXRK scheme with coefficients of the Butcher tableaux within the interval $[-5, 5]$, $L$-stability of the implicit part over the entire LHP, and minimum truncation error, defined, following [9], as

$$A^{(q+1)} = \sqrt{\sum_i \left(\tau_i^{\mathrm{IM}(q+1)}\right)^2 + \sum_i \left(\tau_i^{\mathrm{EX}(q+1)}\right)^2 + \sum_i \left(\tau_i^{\mathrm{IMEX}(q+1)}\right)^2}, \tag{32b}$$

where $q$ is the order of accuracy of the Runge–Kutta scheme, in this case equal to 3. [The same definition is used in Table 1 to compare the truncation error of the various schemes considered.]

This approach is convenient, as the constraint equations depending on both $b_i$ and $c_i$ become linear in $b_i$, which allows a significant simplification of the corresponding optimization problem. Note that all of the schemes developed in [9] follow this approach. In the present case, this strategy leads, for each pair $(c_2, c_3)$, to a set of solutions which depend on the roots of a fifth-order polynomial. Among these, only three are real, and only one of these gives an $L$-stable solution.[6] The resulting scheme, dubbed **IMEXRKCB3f**, is obtained for $c_2 = 49/50$ and $c_3 = 1/25$. The other parameter values are:

$$\begin{aligned} &a_{31}^{\mathrm{IM}} = -\frac{785\,157\,464\,198}{1\,093\,480\,182\,337}, \qquad a_{32}^{\mathrm{IM}} = -\frac{30\,736\,234\,873}{978\,681\,420\,651}, \qquad a_{33}^{\mathrm{IM}} = \frac{983\,779\,726\,483}{1\,246\,172\,347\,126}, \\ &a_{31}^{\mathrm{EX}} = \frac{13\,244\,205\,847}{647\,648\,310\,246}, \qquad a_{32}^{\mathrm{EX}} = \frac{13\,419\,997\,131}{686\,433\,909\,488}, \\ &a_{42}^{\mathrm{EX}} = \frac{231\,677\,526\,244}{1\,085\,522\,130\,027}, \qquad a_{43}^{\mathrm{EX}} = \frac{3\,007\,879\,347\,537}{683\,461\,566\,472}, \\ &b_1 = -\frac{2\,179\,897\,048\,956}{603\,118\,880\,443}, \qquad b_2 = \frac{99\,189\,146\,040}{891\,495\,457\,793}, \\ &b_3 = \frac{6\,064\,140\,186\,914}{1\,415\,701\,440\,113}, \qquad b_4 = \frac{146\,791\,865\,627}{668\,377\,518\,349}, \end{aligned} \tag{32c}$$

and $a_{21}^{\mathrm{IM}} = a_{22}^{\mathrm{IM}} = c_2/2$ and $a_{21}^{\mathrm{EX}} = c_2$ from stage-order conditions. The scheme is not SSP. The associated second-order embedded scheme is given by:

$$\begin{aligned} &\hat{b}_1^{\mathrm{IM}} = 0, \qquad \hat{b}_2^{\mathrm{IM}} = \frac{337\,712\,514\,207}{759\,004\,992\,869}, \qquad \hat{b}_3^{\mathrm{IM}} = \frac{311\,412\,265\,155}{608\,745\,789\,881}, \qquad \hat{b}_4^{\mathrm{IM}} = \frac{52\,826\,596\,233}{1\,214\,539\,205\,236}, \\ &\hat{b}_1^{\mathrm{EX}} = 0, \qquad \hat{b}_2^{\mathrm{EX}} = 0, \qquad \hat{b}_3^{\mathrm{EX}} = \frac{25}{48}, \qquad \hat{b}_4^{\mathrm{EX}} = \frac{23}{48}. \end{aligned} \tag{32d}$$

The stability boundaries of the DIRK and ERK components are shown in Figs. 1(o)–1(p). Notice that the stability region of the explicit component coincides with that of IMEXRKCB3c.

## 5. A fourth-order, 3-register, 6-stage, *L*-stable scheme

Solving the nonlinear system of equations arising from the imposition of the fourth-order accuracy constraints is a difficult task. For this reason, stage-order conditions higher than one are usually imposed, as pointed out in [8]. These

[6] The other solutions give a stability region which does not cover the entire LHP; note that this is not in contradiction with the way we have imposed stability on the scheme during the optimization of the coefficients, since we only impose the behavior of the stability function at infinity, then check the boundary of the resulting stability region only after all the parameters of the scheme have been determined.

conditions simplify the search for a solution by significantly reducing the nonconvexity of the corresponding optimization problem. For this reason, after imposing the same $b_i$ and $c_i$ over the explicit and implicit components and stiff accuracy for the implicit component, we require stage-order two for the implicit component.[7] We also again impose $c_1 = 0$ and $c_6 = 1$ for FSAL structure. This reduces the number of nonlinear equations from fourteen, i.e. one for first order, one for second order, three for third order, and nine for fourth order, to only ten, to which we have to add two constraints for $L$-stability, $2(s-2)$ constraints for stage-order two for the implicit component and $(s-1)$ constraints for stage-order one for the explicit component. Leveraging a six-stage three-register IMEXRK scheme, i.e.

$$\boxed{\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a^{\text{IM}}_{21} & a^{\text{IM}}_{22} & & & & \\
c_3 & a^{\text{IM}}_{31} & a^{\text{IM}}_{32} & a^{\text{IM}}_{33} & & & \\
c_4 & b_1 & a^{\text{IM}}_{42} & a^{\text{IM}}_{43} & a^{\text{IM}}_{44} & & \\
c_5 & b_1 & b_2 & a^{\text{IM}}_{53} & a^{\text{IM}}_{54} & a^{\text{IM}}_{55} & \\
1 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \\
\hline
 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \\
\hline
 & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \hat{b}_4 & \hat{b}_5 & \hat{b}_6
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a^{\text{EX}}_{21} & 0 & & & & \\
c_3 & a^{\text{EX}}_{31} & a^{\text{EX}}_{32} & 0 & & & \\
c_4 & b_1 & a^{\text{EX}}_{42} & a^{\text{EX}}_{43} & 0 & & \\
c_5 & b_1 & b_2 & a^{\text{EX}}_{53} & a^{\text{EX}}_{54} & 0 & \\
1 & b_1 & b_2 & b_3 & a^{\text{EX}}_{64} & a^{\text{EX}}_{65} & 0 \\
\hline
 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \\
\hline
 & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \hat{b}_4 & \hat{b}_5 & \hat{b}_6
\end{array}} \tag{33a}$$

we have 30 degrees of freedom to satisfy 25 constraints. [For the embedded scheme, the coordination assumption $\hat{b}^{\text{EX}} = \hat{b}^{\text{IM}} = \hat{b}$ is again imposed, which proves to provide sufficient freedom in the search for a solution.] As in Section 4, we again perform a (tedious) parametric variation over the coefficients $c_2$, $c_3$, $c_4$, and $c_5$ in the range $[0, 1]$. The last degree of freedom is taken as one of the diagonal terms of the Butcher tableau of the implicit part (we select $a^{\text{IM}}_{55}$), which is varied in the range $[0, 1/2]$ in order to minimize the truncation error (32b). With this approach, it is possible to numerically solve the nonlinear systems arising during the IMEXRK scheme design phase. In particular, 114 solutions are found for each quintuplet $(c_2, c_3, c_4, c_5, a^{\text{IM}}_{55})$. Among these, over half have imaginary coefficients, and are therefore discarded immediately. Among of the remaining solutions, only a few satisfy $L$-stability of the implicit part, and have coefficients in the range $[-5, 5]$. Among the schemes that survived this initial downselection, we have selected the one offering the smallest truncation error while still exhibiting a large extent of the stability region of the explicit part on the negative real axis. It has been found that the set

$$c_2 = 1/4, \qquad c_3 = 3/4, \qquad c_4 = 3/8, \qquad c_5 = 1/2, \qquad a^{\text{IM}}_{55} = 1/2 \tag{33b}$$

gives the best results. The scheme thus obtained, dubbed **IMEXRKCB4** is given by:

$$\begin{aligned}
&a^{\text{IM}}_{31} = \frac{216\,145\,252\,607}{961\,230\,882\,893}, \quad a^{\text{IM}}_{32} = \frac{257\,479\,850\,128}{1\,143\,310\,606\,989}, \quad a^{\text{IM}}_{33} = \frac{30\,481\,561\,667}{101\,628\,412\,017},\\
&a^{\text{IM}}_{42} = -\frac{381\,180\,097\,479}{1\,276\,440\,792\,700}, \quad a^{\text{IM}}_{43} = -\frac{54\,660\,926\,949}{461\,115\,766\,612}, \quad a^{\text{IM}}_{44} = \frac{344\,309\,628\,413}{552\,073\,727\,558},\\
&a^{\text{IM}}_{53} = -\frac{100\,836\,174\,740}{861\,952\,129\,159}, \quad a^{\text{IM}}_{54} = -\frac{250\,423\,827\,953}{1\,283\,875\,864\,443},\\
&a^{\text{EX}}_{31} = \frac{153\,985\,248\,130}{1\,004\,999\,853\,329}, \quad a^{\text{EX}}_{32} = \frac{902\,825\,336\,800}{1\,512\,825\,644\,809},\\
&a^{\text{EX}}_{42} = \frac{99\,316\,866\,929}{820\,744\,730\,663}, \quad a^{\text{EX}}_{43} = \frac{82\,888\,780\,751}{969\,573\,940\,619},\\
&a^{\text{EX}}_{53} = \frac{57\,501\,241\,309}{765\,040\,883\,867}, \quad a^{\text{EX}}_{54} = \frac{76\,345\,938\,311}{676\,824\,576\,433},\\
&a^{\text{EX}}_{64} = -\frac{4\,099\,309\,936\,455}{6\,310\,162\,971\,841}, \quad a^{\text{EX}}_{65} = \frac{1\,395\,992\,540\,491}{933\,264\,948\,679},\\
&b_1 = \frac{232\,049\,084\,587}{1\,377\,130\,630\,063}, \quad b_2 = \frac{322\,009\,889\,509}{2\,243\,393\,849\,156}, \quad b_3 = -\frac{195\,109\,672\,787}{1\,233\,165\,545\,817},\\
&b_4 = -\frac{340\,582\,416\,761}{705\,418\,832\,319}, \quad b_5 = \frac{463\,396\,075\,661}{409\,972\,144\,477}, \quad b_6 = \frac{323\,177\,943\,294}{1\,626\,646\,580\,633},
\end{aligned} \tag{33c}$$

[7] Note that, even if it were desired to impose the same stage-order for both implicit and explicit components, in order to improve algebraic variable accuracy, this is not possible, as the low-storage structure used here removes the necessary degrees of freedom to impose such a condition.

and $a_{21}^{\mathrm{EX}} = c_2$ and $a_{21}^{\mathrm{IM}} = a_{22}^{\mathrm{IM}} = c_2/2$ from the stage-order conditions. The scheme is not SSP. The associated third-order embedded scheme is:

$$\hat{b}_1 = \frac{5\,590\,918\,588}{49\,191\,225\,249}, \qquad \hat{b}_2 = \frac{92\,380\,217\,342}{122\,399\,335\,103}, \qquad \hat{b}_3 = -\frac{29\,257\,529\,014}{55\,608\,238\,079},$$
$$\hat{b}_4 = -\frac{126\,677\,396\,901}{66\,917\,692\,409}, \qquad \hat{b}_5 = \frac{384\,446\,411\,890}{169\,364\,936\,833}, \qquad \hat{b}_6 = \frac{58\,325\,237\,543}{207\,682\,037\,557}. \tag{33d}$$

The stability boundaries of the DIRK and ERK components are shown in Figs. 1(q)–1(r).

## 6. Order reduction

We now consider the *order reduction* present when the schemes developed above are applied to the van der Pol equation. It is well documented in the literature (see, e.g., [8]) that whenever an RK method is used to integrate a singular perturbation problem (that is, an ODE characterized by a stiffness parameter $\varepsilon$ whose behavior transitions towards that of an index-1 DAE as the stiffness increases), the observed convergence rate appears to be lower than the nominal order of accuracy of the RK scheme used. In the seminal work of Hairer et al. [7], it is shown that the global error of DIRK schemes applied to singular perturbation problems may be written in the convenient form $E = C_1\,(\Delta t)^{n_1} + C_2\,\varepsilon\,(\Delta t)^{n_2}$. For the differential variables, DIRK methods have $n_1 = n$ and $n_2 = n_{SO} + 1$, where $n$ is the nominal order of accuracy and $n_{SO}$ is the stage order of the scheme. For the algebraic variables, if the DIRK method satisfies the aforementioned “stiff-accuracy” conditions, it turns out that[8] $n_1 = n$ and $n_2 = n_{SO}$; if not, however, $n_1 = n_{SO} + 1$ and $C_2 = 0$, which is generally much worse.

For IMEXRK methods, very little is known about order reduction outside of the empirical work of Kennedy and Carpenter in [9] and [5], where various IMEX schemes are tested on a range of singular perturbation problems. In this work, the greatest order reduction is observed in the case of the van der Pol equation; for this reason, we focus on this model problem in the present paper in order to characterize the order reduction phenomenon. The van der Pol equation describes the dynamics of a nonlinear oscillator of the form

$$\frac{dy}{dt} = z, \qquad \varepsilon\,\frac{dz}{dt} = \left(1 - y^2\right) z - y, \tag{34}$$

where $\varepsilon$ is known as the stiffness parameter. It is seen that, for $\varepsilon \to 0$, this ODE system transitions into an index-1 DAE, where $y(t)$ is a differential variable, and $z(t)$ transitions into an algebraic variable. The initial conditions used are $y(0) = 2$ and $z(0) = -0.6666654321121172$. All of the schemes introduced in this paper have been tested on this system over the time interval $0 \le t \le T$, taking $T = 0.5$, with various values for the (constant) stepsize $\Delta t$ and stiffness parameter $\varepsilon$. The error at $t = T$ has then been used to estimate the convergence rate (that is, $n_1$ and $n_2$) as the stiffness parameter $\varepsilon$ is decreased. The procedure used is analogous to that described in [9]: by fixing $\varepsilon$ and varying $\Delta t$ in the $\Delta t \to \varepsilon$ limit, the change of slope in the convergence rate has been detected and used to estimate $n_1$ and $n_2$. Results of such simulations are reported in Fig. 4, and empirical estimates of the convergence rates for each method are reported in Table 2. When only the DIRK component of the schemes are used, the results generally show good agreement with the theoretical bounds provided in [7]. If the entire IMEX schemes are used, results do not differ substantially from those reported in [9]. The order-reduction phenomenon tends to be problem dependent; results in practice (see [9]) often indicate behavior significantly better than the corresponding theoretical bounds. Note also that imposing stage-order two on the DIRK component of a scheme does not influence the convergence of the entire IMEX scheme, though it significantly improves the accuracy when the DIRK component only is used.

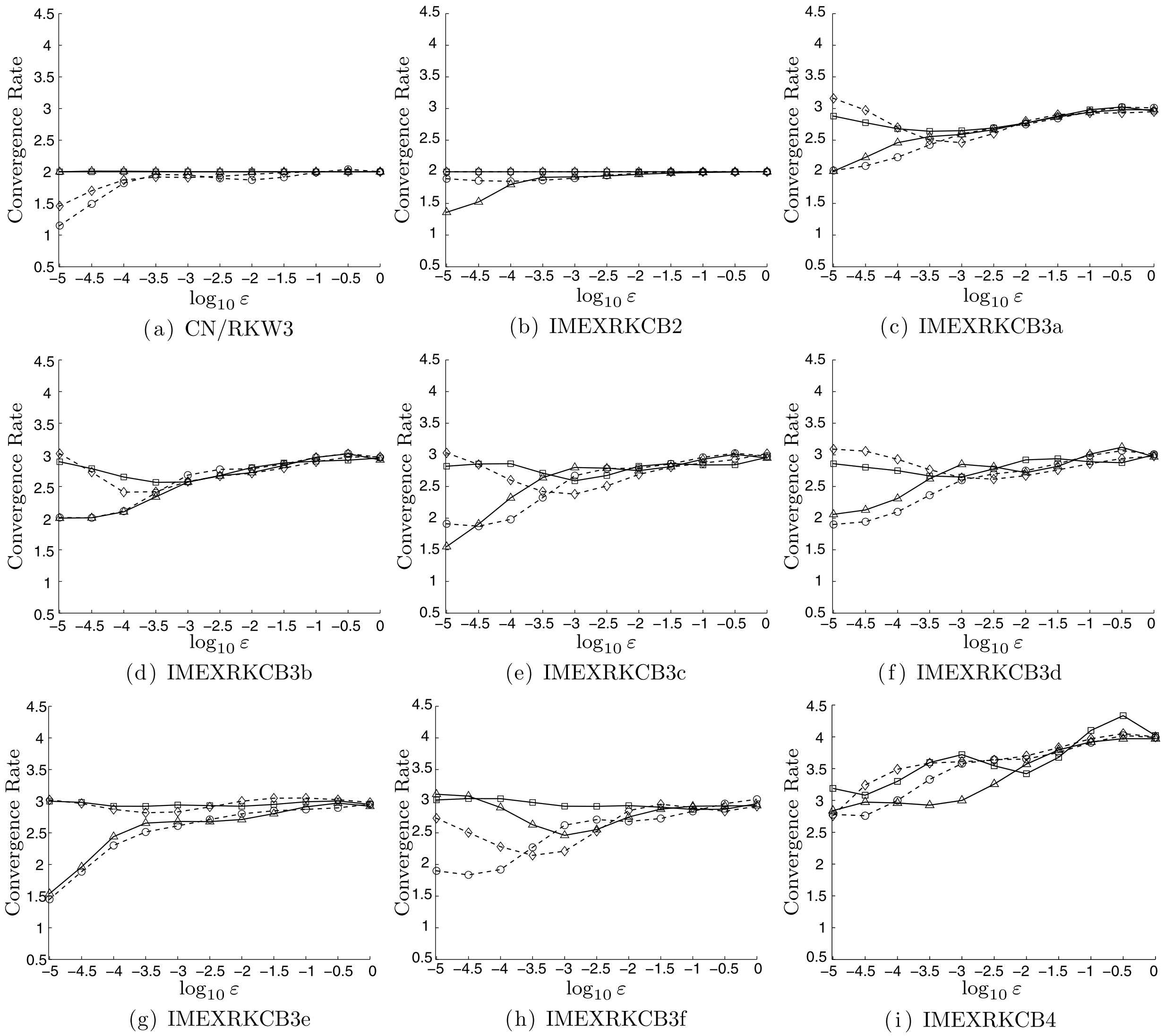
## 7. Computational cost

To illustrate the relative computational cost of our new low-storage IMEXRK schemes on a representative PDE model problem discretized on $N \gg 1$ gridpoints, we now compare the efficient implementation of each of the methods developed herein to CN/RKW3 and several full-storage IMEX Runge–Kutta schemes available in literature. We consider as a model PDE problem the one-dimensional Kuramoto–Sivashinsky equation

$$\frac{\partial u}{\partial t} = -u\,\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4} \tag{35}$$

over the domain $x \in [-L/2, L/2]$ with $u = \partial u/\partial x = 0$ at $x = \pm L/2$, where $L$ is the width of the domain. It should be remarked that, unlike the van der Pol case, this example represents a rather undemanding application of our IMEXRK schemes. The sole purpose of this analysis is the comparison of the computational cost that our new schemes require with respect to other IMEXRK schemes available in literature; the implementation of a selection of these schemes in a DNS code for the simulation of an incompressible turbulent channel flow is currently underway, and will be reported elsewhere. The RHS of (35) consists of a nonlinear convective term, treated explicitly, and two linear terms, treated implicitly.

[8] Indeed, it is precisely for this reason that these “stiff-accuracy” conditions are so named.



(a) CN/RKW3

(b) IMEXRKCB2

(c) IMEXRKCB3a

(d) IMEXRKCB3b

(e) IMEXRKCB3c

(f) IMEXRKCB3d

(g) IMEXRKCB3e

(h) IMEXRKCB3f

(i) IMEXRKCB4

**Fig. 4.** Convergence rates for the low-storage IMEXRK schemes considered in this paper when applied to the van der Pol equation, as a function of $\varepsilon$. Solid lines are for simulations using the DIRK component only (squares for the differential variables, triangles for the algebraic variables), whereas dashed lines are for the simulations using the entire IMEXRK scheme (diamonds for the differential variables, circles for the algebraic variables).

**Table 2**
Estimated convergence rates of the differential and algebraic variables on the van der Pol equation for CN/RKW3 and the IMEXRK schemes presented in this paper, and their associated DIRK components only.

| Method | IMEXRK scheme differential part | IMEXRK scheme algebraic part | DIRK scheme only differential part | DIRK scheme only algebraic part |
|---|---|---|---|---|
| CN/RKW3 | $(\Delta t)^2 + \varepsilon(\Delta t)^1$ | $(\Delta t)^2 + \varepsilon(\Delta t)^1$ | $(\Delta t)^2 + \varepsilon(\Delta t)^2$ | $(\Delta t)^2 + \varepsilon(\Delta t)^2$ |
| IMEXRKCB2 | $(\Delta t)^2 + \varepsilon(\Delta t)^2$ | $(\Delta t)^2 + \varepsilon(\Delta t)^1$ | $(\Delta t)^2 + \varepsilon(\Delta t)^2$ | $(\Delta t)^2 + \varepsilon(\Delta t)^1$ |
| IMEXRKCB3a | $(\Delta t)^3 + \varepsilon(\Delta t)^2$ | $(\Delta t)^2 + \varepsilon(\Delta t)^1$ | $(\Delta t)^3 + \varepsilon(\Delta t)^2$ | $(\Delta t)^3 + \varepsilon(\Delta t)^1$ |
| IMEXRKCB3b | $(\Delta t)^3 + \varepsilon(\Delta t)^2$ | $(\Delta t)^2 + \varepsilon(\Delta t)^1$ | $(\Delta t)^3 + \varepsilon(\Delta t)^2$ | $(\Delta t)^3 + \varepsilon(\Delta t)^1$ |
| IMEXRKCB3c | $(\Delta t)^3 + \varepsilon(\Delta t)^2$ | $(\Delta t)^2 + \varepsilon(\Delta t)^1$ | $(\Delta t)^3 + \varepsilon(\Delta t)^2$ | $(\Delta t)^3 + \varepsilon(\Delta t)^1$ |
| IMEXRKCB3d | $(\Delta t)^3 + \varepsilon(\Delta t)^2$ | $(\Delta t)^2 + \varepsilon(\Delta t)^1$ | $(\Delta t)^3 + \varepsilon(\Delta t)^2$ | $(\Delta t)^3 + \varepsilon(\Delta t)^1$ |
| IMEXRKCB3e | $(\Delta t)^3 + \varepsilon(\Delta t)^2$ | $(\Delta t)^2 + \varepsilon(\Delta t)^1$ | $(\Delta t)^3 + \varepsilon(\Delta t)^2$ | $(\Delta t)^3 + \varepsilon(\Delta t)^1$ |
| IMEXRKCB3f | $(\Delta t)^3 + \varepsilon(\Delta t)^2$ | $(\Delta t)^2 + \varepsilon(\Delta t)^1$ | $(\Delta t)^3 + \varepsilon(\Delta t)^3$ | $(\Delta t)^3 + \varepsilon(\Delta t)^2$ |
| IMEXRKCB4 | $(\Delta t)^4 + \varepsilon(\Delta t)^2$ | $(\Delta t)^3 + \varepsilon(\Delta t)^1$ | $(\Delta t)^4 + \varepsilon(\Delta t)^3$ | $(\Delta t)^4 + \varepsilon(\Delta t)^2$ |

Following a five-point central finite-difference (FD) approach on a uniform grid, (35) can be approximated as

$$\frac{d\mathbf{u}}{dt} = A\,\mathbf{u} + \mathbf{g}(\mathbf{u}),$$

where $A$ is a pentadiagonal Toeplitz matrix obtained by discretizing the last two terms on the RHS of (35), and $g_i(\mathbf{u}) = -u_i(u_{i-2} - 8u_{i-1} + 8u_{i+1} - u_{i+2})/(12\Delta x)$. As an example, using the 3-register implementation (19) of the CN/RKW3 method (23), $6N$ flops times 3 stages are required for the evaluation of the nonlinear term, $19N$ flops times 3 stages are required for the implicit (pentadiagonal) solves, and $40N$ additional flops are required for basic product/sum operations; thus, $115N$ flops per timestep are required.

Following a pseudospectral (PS) approach, with nonlinear products computed in physical space and spatial derivatives computed in Fourier space, (35) can be written in wavenumber space as

$$\frac{d\hat{u}_n}{dt} = -\frac{\imath\, k_{x_n}}{2}\widehat{(u^2)}_n + \left(k_{x_n}^2 - k_{x_n}^4\right)\hat{u}_n \tag{36}$$

where $\imath = \sqrt{-1}$, $k_{x_n} = 2\pi n/L$ is the wavenumber, and $\widehat{(u^2)}_n$ denotes the $n$th wavenumber component of the function computed by transforming $u$ to physical space on $N = 2^p$ equispaced gridpoints, computing $u^2$ at each gridpoint, and transforming the result back to Fourier space. Since computing FFTs requires $\sim 5N \log N$ real flops while all other operations are linear in $N$, the number of FFTs performed represents the leading-order computational cost for large $N$. As an example, the 3-register implementation of CN/RKW3 requires 2 FFTs per stage for each of three stages.

The computational cost of the other schemes may be counted similarly; results are summarized in the last two columns of Table 1. It is seen that, if computational cost is naïvely characterized simply by the number of floating point operations required per timestep, the present low-storage IMEXRK schemes are in fact competitive with both CN/RKW3 and all of the full-storage IMEXRK schemes available in the literature of the corresponding order. The fact that CN/RKW3 and all of our low-storage IMEXRK schemes admit two-, three-, or four-register implementations, however, bestows them with a distinct operational advantage for high-dimensional ODE discretizations of PDE systems.

## 8. Conclusions

We have developed eight new IMEX Runge–Kutta schemes with reduced storage requirements, the properties of which are succinctly summarized and compared with competing schemes in Table 1. It is seen that:

- IMEXRKCB2 is second-order accurate, like CN/RKW3; IMEXRKCB3a–3f are third-order accurate, and IMEXRKCB4 is fourth-order accurate.
- IMEXRKCB2 and 3a–3e, like CN/RKW3, admit both two-register and three-register implementations, with the three-register implementations requiring slightly fewer flops.
- IMEXRKCB3f and 4 admit both three-register and four-register implementations, with the four-register implementations requiring significantly fewer flops; the four-register implementations of these two schemes are thus generally recommended, unless the additional storage that the four-register implementations require represents a particularly acute computational disadvantage.
- IMEXRKCB2 and IMEXRKCB3a generally require fewer floating-point operations per timestep than CN/RKW3, whereas the other schemes we have developed generally require progressively more; this comparison, however, is somewhat problem dependent.
- IMEXRKCB2, 3c–3f, and 4 are $L$-stable, whereas IMEXRKCB3a and 3b are strongly $A$-stable (CN/RKW3 is only $A$-stable), making them well suited for stiff ODEs.
- IMEXRKCB2, 3c, 3d, 3f, and 4 are each provided with a reduced-order embedded scheme following the guidelines listed in Section 1.2, making them well suited for application in adaptive time-stepping applications.
- IMEXRKCB3b incorporates an ESDIRK implicit component, and is thus better suited to leverage an LU decomposition during the implicit solves than either CN/RKW3 or our other schemes.
- IMEXRKCB2, 3c, and 3d are strong stability preserving (SSP) under the appropriate timestep restriction, and are thus better suited for application to hyperbolic systems than either CN/RKW3 or our other schemes.
- IMEXRKCB3f and 4 have stage order two, whereas CN/RKW3 and our other schemes have stage order one; these two schemes thus show better convergence properties when applied to especially stiff ODE systems.

Implementation of these schemes into our lab's benchmark DNS code, diablo, is currently underway.

## Acknowledgements

The authors would like to thank one of the reviewers for the especially valuable comments and suggestions, which contributed to improve the results of the present work. The authors also gratefully acknowledge the financial support of AFOSR FA9550-12-1-0046 and NSF CNS-1035828.

## References

[1] K. Akselvoll, P. Moin, Large eddy simulation of turbulent confined coannular jets and turbulent flow over a backward facing step, Rep. TF-63, Thermosciences Division, Dept. of Mech. Eng., Stanford University, 1995.

[2] 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.

[3] J.C. Butcher, Numerical Methods for Ordinary Differential Equations, Wiley, 2008.

[4] 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.

[5] M.H. Carpenter, C.A. Kennedy, H. Bijl, S.A. Viken, V.N. Vatsa, Fourth-order Runge–Kutta schemes for fluid mechanics applications, J. Sci. Comput. 25 (2005).

[6] S. Gottlieb, C.W. Shu, E. Tadmor, Strong-stability-preserving high order time discretization methods, SIAM Rev. 43 (2001) 89–112.

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

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

[9] C.A. Kennedy, M.H. Carpenter, R.M. Lewis, Additive Runge–Kutta schemes for convection–diffusion–reaction equations, Appl. Numer. Math. 44 (2003) 139–181.

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

[11] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308–323.

[12] J. Kim, P. Moin, B. Moser, Turbulence statistics in fully developed channel flow at low Reynolds number, J. Fluid Mech. 177 (1987) 133–166.

[13] H. Le, P. Moin, An improvement of fractional step methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 92 (1991) 369–379.

[14] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.

[15] L. Pareschi, G. Russo, Implicit–explicit Runge–Kutta schemes and applications to hyperbolic systems with relaxation, J. Sci. Comput. 25 (2003) 129–155.

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

[17] C.W. Shu, Total-variation-diminishing time discretizations, SIAM J. Sci. Stat. Comput. 9 (1988) 1073–1084.

[18] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.

[19] P. van der Houwen, Explicit Runge–Kutta formulas with increased stability boundaries, Numer. Math. 20 (1972) 149–164.

[20] J.H. Williamson, Low-storage Runge–Kutta schemes, J. Comput. Phys. 35 (1980) 48–56.

[21] S. Wolfram, The Mathematica Book, fifth edition, Cambridge University Press, Cambridge, 2003.

[22] A.A. Wray, Minimal-storage time advancement schemes for spectral methods, NASA Technical Report, 1986.