Fluid Dynamics and Co-located Conferences
June 24-27, 2013, San Diego, CA
21st AIAA Computational Fluid Dynamics Conference

10.2514/6.2013-3094

Low-storage IMEX Runge-Kutta schemes for the simulation of Navier-Stokes systems

Daniele Cavaglieri Pooriya Beyhaghi Thomas Bewley

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 one second-order, three third-order and one fourth-order IMEXRK schemes of the low-storage variety, all of which have the same or comparable low storage requirements, better stability properties, and either fewer or slightly more floating-point operations per timestep as the venerable (and, up to now, unique) second-order two-register Crank-Nicolson/Runge-Kutta-Wray (CN/RKW3) IMEXRK algorithm that has dominated the DNS/LES literature for the last two decades.

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 the pioneering work of Kim & Moin [9] and Kim, Moin, & Moser [10] 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 Le & Moin [11], 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 [18] 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, 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 diffusion terms as the "stiff terms" and the convective terms as the "nonstiff terms". Note that a better choice for discretizations with significant grid clustering implemented in one or more 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., the direction normal to the nearest wall) as the "stiff" direction, and the diffusion and convection terms with derivatives in the other directions as the "nonstiff" terms, as suggested by Akselvoll & Moin [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., Le & Moin [11]). 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 the divergence-free constraint, may subsequently be addressed in an identical manner as discussed in [1], [11], and elsewhere in the literature.

In the present work, we develop a new family of low-storage IMEXRK schemes well suited for turbulent flow simulations, and other computational grand challenge applications, using either two or three registers. With an eye on the computational cost of their implementation, we focus on third-order and fourth-order IMEXRK schemes. We denote each scheme as IMEXRKnsS[rR]x, where n is the order of accuracy, s is the number of stages, $r = 2$ is the minimum number of registers needed for implementation, and x reflects the stablility properties of the scheme's implicit component (see §1.1.1). A (hardware-dependent) trade-off between flops and storage must ultimately be conducted to select between the two-register and three-register implementation of each scheme.

The paper is organized as follows:

- §1.1 presents the general structure of IMEXRK schemes, their general implementation, conditions on their parameters for second-order and third-order accuracy, and characterizations of their stability;
- §1.2 presents the general 2-register IMEXRK form, and 3-register & 2-register implementations of this form;
- §2 presents the classical second-order, three-stage, A-stable CN/RKW3 scheme which, prior to the present work, was the only existing prototypical example of a 2-register IMEXRK scheme;
- §3 through §4 present our four 2-register IMEXRK schemes;
- §5 presents our 3-register fourth-order IMEXRK schemes; and
- §6 considers the application of all of these 2-register 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, & Lewis [7]. 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

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

for the time advancement of an ODE of the form

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

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 $A^{-1}\mathbf{b}$, 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

$$\texttt{for} \quad k = 1 : s \tag{3a}$$

$$\texttt{if} \quad k == 1, \quad \mathbf{y} = \mathbf{x}, \quad \texttt{else}, \quad \mathbf{y} = \mathbf{x} + \sum\nolimits_{i=1}^{k-1} a_{ki}^{\mathrm{IM}}\, \Delta t\, \mathbf{f}_i + \sum\nolimits_{j=1}^{k-1} a_{kj}^{\mathrm{EX}}\, \Delta t\, \mathbf{g}_j, \quad \texttt{end} \tag{3b}$$

$$\mathbf{f}_k = A\,(I - a_{kk}^{\mathrm{IM}}\, \Delta t\, A)^{-1}\mathbf{y} \qquad [\text{equivalently,} \;\; \mathbf{f}_k = (I - a_{kk}^{\mathrm{IM}}\, \Delta t\, A)^{-1} A\,\mathbf{y}\,] \tag{3c}$$

$$\mathbf{g}_k = \mathbf{g}(\mathbf{y} + a_{kk}^{\mathrm{IM}}\, \Delta t\, \mathbf{f}_k,\, t_n + c_k^{\mathrm{EX}}\, \Delta t) \tag{3d}$$

$$\texttt{end} \tag{3e}$$

$$\mathbf{x} \leftarrow \mathbf{x} + \sum\nolimits_{i=1}^{s} b_i^{\mathrm{IM}}\, \Delta t\, \mathbf{f}_i + \sum\nolimits_{j=1}^{s} b_j^{\mathrm{EX}}\, \Delta t\, \mathbf{g}_j \tag{3f}$$

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 [13]) to find the $\mathbf{z}$ such that $\mathbf{e}(\mathbf{z}) = 0$:

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

The Jacobian used in this iteration may be computed analytically or approximated numerically, and the initialization of this iteration may sometimes be significantly improved with a so-called "dense output" strategy; see [7] for details. The low-storage IMEXRK algorithms developed in this work may be applied in the linear or nonlinear setting, mutatis mutandis; §1.2.1 and §1.2.2 provide low-storage pseudocode implementations in the case in which the stiff part of the ODE is linear.

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

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

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

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

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

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

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

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

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

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

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

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

For the DIRK and ERK components in (1), when used together in an IMEX fashion, to be second-order accurate, it is additionally required that

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

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

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

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

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

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

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

and for these schemes, when used together in an IMEX fashion, to be fourth-order accurate, 44 additional constraints are required (see [7]), 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 $\sigma(\infty) = 0$.

In our naming convention (see third paragraph of §1), $x =$'A' denotes A-stability, $x =$'A+' denotes strong A stability, and $x =$'L' denotes L-stability, and parametric variations of a given scheme are indicated with a Greek suffix.

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 in concert, 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 $z^{\text{IM}} = \lambda_f\,\Delta t$, $z^{\text{EX}} = \lambda_g\,\Delta t$, and $\sigma(z^{\text{IM}}; z^{\text{EX}}) = x_{n+1}/x_n$, we may write (see [7])

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

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 $\sigma(z^{\text{IM}}; z^{\text{EX}})$ as $z^{\text{IM}} \to \infty$ for finite z^{EX}.

1.2 Low-storage IMEXRK schemes

All existing literature on low-storage RK schemes to date appears to focus 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 $\mathbf{x} = \mathbf{x}_{N\times 1}$. In the present work, we extend the two- and three-register van der Houwen class (see [16]) of ERK schemes, a comprehensive review of which is given in Kennedy, Carpenter, & Lewis [8], to the DIRK case, which can be accomplished by restricting all the elements below the first (second, for the three-register case) lower subdiagonal to be equal to the corresponding b_i in the same column. Further, we consider coordinated pairs of such two- and three-register DIRK and ERK schemes in the IMEX setting described in §1.1. In particular we will develop two-register third-order scheme and a three-register fourth-order scheme.

As shown in §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 forty-four constraints for fourth-order accuracy. Before proceeding, we thus introduce some simplifying assumptions. Following [12] and [7] and the CN/RKW3 scheme of [11], 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 $b_i^{\mathrm{IM}} = b_i^{\mathrm{EX}} = b_i$ for $i = 1, \ldots, s$, as also done in [12] and [7], but which is not satisfied by CN/RKW3. Finally, for each stage, a stage order of one is also imposed such that

$$\sum\nolimits_{j=1}^{i} a_{ij}^{\mathrm{IM}} = \sum\nolimits_{j=1}^{i-1} a_{ij}^{\mathrm{EX}} = c_i \quad \text{for} \;\; i = 1, \ldots, s; \tag{6}$$

it follows that $c_1 = a_{11}^{\mathrm{IM}} = a_{11}^{\mathrm{EX}} = 0$. 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 just five, and the constraints for fourth-order accuracy are just nine.

For the third-order schemes, a second-order embedded scheme is also implemented without the assumption $\hat{b}_i^{\mathrm{IM}} = \hat{b}_i^{\mathrm{EX}}$, in order to benefit from higher freedom in the design phase. In this case, four constraints must be imposed for accuracy, i.e. (4a) and (4b). As a guideline, none of the third-order truncation terms must vanish so that each one will contribute to the error estimate. Furthermore the DIRK part must achieve at least A-stability in order for the error estimation to stay bounded[1]. The remaining free parameters are then optimized in order to increase the overall magnitude of the third-order truncation terms. The IMEX Butcher tableaux in (1) for the two-register implementation are simplified as follows:

$$\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a_{2,1}^{\mathrm{IM}} & a_{2,2}^{\mathrm{IM}} & & & & \\
c_3 & b_1 & a_{3,2}^{\mathrm{IM}} & a_{3,3}^{\mathrm{IM}} & & & \\
c_4 & b_1 & b_2 & a_{4,3}^{\mathrm{IM}} & a_{4,4}^{\mathrm{IM}} & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\
c_s & b_1 & b_2 & \cdots & b_{s-2} & a_{s,s-1}^{\mathrm{IM}} & a_{s,s}^{\mathrm{IM}} \\
\hline
 & b_1 & b_2 & \cdots & b_{s-2} & b_{s-1} & b_s \\
 & \hat{b}_1^{\mathrm{IM}} & \hat{b}_2^{\mathrm{IM}} & \cdots & \hat{b}_{s-2}^{\mathrm{IM}} & \hat{b}_{s-1}^{\mathrm{IM}} & \hat{b}_s^{\mathrm{IM}}
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a_{2,1}^{\mathrm{EX}} & 0 & & & & \\
c_3 & b_1 & a_{3,2}^{\mathrm{EX}} & 0 & & & \\
c_4 & b_1 & b_2 & a_{4,3}^{\mathrm{EX}} & 0 & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\
c_s & b_1 & b_2 & \cdots & b_{s-2} & a_{s,s-1}^{\mathrm{EX}} & 0 \\
\hline
 & b_1 & b_2 & \cdots & b_{s-2} & b_{s-1} & b_s \\
 & \hat{b}_1^{\mathrm{EX}} & \hat{b}_2^{\mathrm{EX}} & \cdots & \hat{b}_{s-2}^{\mathrm{EX}} & \hat{b}_{s-1}^{\mathrm{EX}} & \hat{b}_s^{\mathrm{EX}}
\end{array} \tag{7}$$

In case a three-register implementation is considered, instead, the IMEX Butcher tableaux in (1) simplifies as follows:

$$\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a_{2,1}^{\mathrm{IM}} & a_{2,2}^{\mathrm{IM}} & & & & \\
c_3 & a_{3,1}^{\mathrm{IM}} & a_{3,2}^{\mathrm{IM}} & a_{3,3}^{\mathrm{IM}} & & & \\
c_4 & b_1 & a_{4,2}^{\mathrm{IM}} & a_{4,3}^{\mathrm{IM}} & a_{4,4}^{\mathrm{IM}} & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\
c_s & b_1 & b_2 & \cdots & a_{s,s-2}^{\mathrm{IM}} & a_{s,s-1}^{\mathrm{IM}} & a_{s,s}^{\mathrm{IM}} \\
\hline
 & b_1 & b_2 & \cdots & b_{s-2} & b_{s-1} & b_s \\
 & \hat{b}_1^{\mathrm{IM}} & \hat{b}_2^{\mathrm{IM}} & \cdots & \hat{b}_{s-2}^{\mathrm{IM}} & \hat{b}_{s-1}^{\mathrm{IM}} & \hat{b}_s^{\mathrm{IM}}
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a_{2,1}^{\mathrm{EX}} & 0 & & & & \\
c_3 & a_{3,1}^{\mathrm{EX}} & a_{3,2}^{\mathrm{EX}} & 0 & & & \\
c_4 & b_1 & a_{4,2}^{\mathrm{EX}} & a_{4,3}^{\mathrm{EX}} & 0 & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\
c_s & b_1 & b_2 & \cdots & a_{s,s-2}^{\mathrm{IM}} & a_{s,s-1}^{\mathrm{EX}} & 0 \\
\hline
 & b_1 & b_2 & \cdots & b_{s-2} & b_{s-1} & b_s \\
 & \hat{b}_1^{\mathrm{EX}} & \hat{b}_2^{\mathrm{EX}} & \cdots & \hat{b}_{s-2}^{\mathrm{EX}} & \hat{b}_{s-1}^{\mathrm{EX}} & \hat{b}_s^{\mathrm{EX}}
\end{array} \tag{8}$$

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

$$\sigma(z^{\mathrm{IM}}; z^{\mathrm{EX}}) = \frac{1 + \sum_{i=1}^{s} p_i(z^{\mathrm{EX}})\,[z^{\mathrm{IM}}]^i}{1 + \sum_{i=1}^{s-1} q_i\,[z^{\mathrm{IM}}]^i} \quad \text{where} \quad p_i(z^{\mathrm{EX}}) = \sum\nolimits_{j=0}^{s-i} \hat{p}_{ij}\,[z^{\mathrm{EX}}]^j. \tag{9}$$

[1]Notice that these last two conditions are not always achievable, hence not all the schemes here developed come with an embedded scheme.

1.2.1 General three-register implementation of IMEXRK[2R] 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 (7) that uses three registers[2] of length N, plus one additional register for error control purpose, to advance from $\mathbf{x} = \mathbf{x}_n$ to $\mathbf{x} = \mathbf{x}_{n+1}$ proceeds as follows

$$
\begin{aligned}
&\texttt{for}\quad k = 1 : s \\
&\quad \texttt{if}\quad k == 1, \quad \mathbf{y} = \mathbf{x}, \quad \texttt{else}, \quad \mathbf{y} \leftarrow \mathbf{x} + (a^{\mathrm{IM}}_{k,k-1} - b^{\mathrm{IM}}_{k-1})\,\Delta t\,\mathbf{z} + (a^{\mathrm{EX}}_{k,k-1} - b^{\mathrm{EX}}_{k-1})\,\Delta t\,\mathbf{y}, \quad \texttt{end} \\
&\quad \mathbf{z} = (I - a^{\mathrm{IM}}_{k,k}\,\Delta t\,A)^{-1} A\,\mathbf{y} \\
&\quad \mathbf{y} \leftarrow \mathbf{g}(\mathbf{y} + a^{\mathrm{IM}}_{k,k}\,\Delta t\,\mathbf{z},\, t_n + c^{\mathrm{EX}}_k \Delta t) \\
&\quad \mathbf{x} \leftarrow \mathbf{x} + b^{\mathrm{IM}}_k\,\Delta t\,\mathbf{z} + b^{\mathrm{EX}}_k\,\Delta t\,\mathbf{y} \\
&\quad \hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \hat{b}^{\mathrm{IM}}_k\,\Delta t\,\mathbf{z} + \hat{b}^{\mathrm{EX}}_k\,\Delta t\,\mathbf{y} \\
&\texttt{end}
\end{aligned}
\tag{10}
$$

where $\mathbf{z}$ and $\mathbf{y}$ store the implicit and explicit parts of the RHS at each stage, $\hat{\mathbf{x}}$ stores the solution of the embedded scheme and $\mathbf{x}$ is used to advance the solution of the main scheme[3]. 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 in §1.1, implementation in the case with nonlinear stiff part is a straightforward extension.

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

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 §1.2.1 may be rewritten as:

$$
\begin{aligned}
&\texttt{for}\quad k = 1 : s \\
&\quad \texttt{if}\quad k == 1, \quad \mathbf{y} = \mathbf{x}, \quad \texttt{else} \\
&\qquad \mathbf{y} \leftarrow \mathbf{x} + (a^{\mathrm{IM}}_{k,k-1} - b^{\mathrm{IM}}_{k-1})\,\Delta t\,A\,\mathbf{y} + (a^{\mathrm{EX}}_{k,k-1} - b^{\mathrm{EX}}_{k-1})\,\Delta t\,\mathbf{g}(\mathbf{y},\, t_n + c^{\mathrm{EX}}_{k-1}\Delta t) \\
&\quad \texttt{end} \\
&\quad \mathbf{y} \leftarrow (I - a^{\mathrm{IM}}_{k,k}\,\Delta t\,A)^{-1}\mathbf{y} \\
&\quad \mathbf{x} \leftarrow \mathbf{x} + b^{\mathrm{IM}}_k\,\Delta t\,A\,\mathbf{y} + b^{\mathrm{EX}}_k\,\Delta t\,\mathbf{g}(\mathbf{y},\, t_n + c^{\mathrm{EX}}_k\Delta t) \\
&\quad \hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \hat{b}^{\mathrm{IM}}_k\,\Delta t\,A\,\mathbf{y} + \hat{b}^{\mathrm{EX}}_k\,\Delta t\,\mathbf{g}(\mathbf{y},\, t_n + c^{\mathrm{EX}}_k\Delta t) \\
&\texttt{end}
\end{aligned}
\tag{11}
$$

In this case, one linear solve of the form $(I - cA)^{-1}\mathbf{b}$ and two operations of the form[4] $\mathbf{x} + c\,A\,\mathbf{y} + d\,\mathbf{g}(\mathbf{y}, t)$ are computed per stage (an additional operation and register must be considered when the embedded scheme is used for error control), in addition to various level-1 BLAS operations, but the storage requirements are 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 §6.

[2] That is, in addition to the extra memory required to solve the linear system, which is problem dependent.

[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 §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.

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

For the development of the fourth-order scheme only, due to the complexity of the problem and the significant number of constraints to be imposed, a three-register implementation (8) has been considered, which, for the sake of reducing the computational cost, allows a four-register implementation, as shown below:

$$
\begin{aligned}
&\texttt{for}\quad k = 1 : s \\
&\quad \texttt{if}\quad k == 1, \quad \mathbf{y} = \mathbf{x}, \quad \mathbf{z}^{\mathrm{IM}} = \mathbf{x}, \quad \mathbf{z}^{\mathrm{EX}} = \mathbf{x}, \quad \texttt{else} \\
&\qquad \mathbf{z}^{\mathrm{EX}} \leftarrow \mathbf{y} + a^{\mathrm{EX}}_{k,k-1}\,\Delta t\,\mathbf{z}^{\mathrm{EX}} \\
&\qquad \mathbf{y} \leftarrow \mathbf{x} + (a^{\mathrm{IM}}_{k,k-1} - b^{\mathrm{IM}}_{k-1})\,\Delta t\,\mathbf{z}^{\mathrm{IM}} + (a^{\mathrm{EX}}_{k,k-1} - b^{\mathrm{EX}}_{k-1})\,(\mathbf{z}^{\mathrm{EX}} - \mathbf{y})/a^{\mathrm{EX}}_{k,k-1} \\
&\qquad \mathbf{z}^{\mathrm{EX}} \leftarrow \mathbf{z}^{\mathrm{EX}} + a^{\mathrm{IM}}_{k,k-1}\,\Delta t \mathbf{z}^{\mathrm{IM}} \\
&\quad \texttt{end} \\
&\quad \mathbf{z}^{\mathrm{IM}} = (I - a^{\mathrm{IM}}_{k,k}\,\Delta t\,A)^{-1} A\,\mathbf{z}^{\mathrm{EX}} \\
&\quad \mathbf{z}^{\mathrm{EX}} \leftarrow \mathbf{g}(\mathbf{z}^{\mathrm{EX}} + a^{\mathrm{IM}}_{k,k}\,\Delta t\,\mathbf{z}^{\mathrm{IM}},\, t_n + c^{\mathrm{EX}}_k \Delta t) \\
&\quad \mathbf{x} \leftarrow \mathbf{x} + b^{\mathrm{IM}}_k\,\Delta t\,\mathbf{z}^{\mathrm{IM}} + b^{\mathrm{EX}}_k\,\Delta t\,\mathbf{z}^{\mathrm{EX}} \\
&\quad \hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \hat{b}^{\mathrm{IM}}_k\,\Delta t\,\mathbf{z}^{\mathrm{IM}} + \hat{b}^{\mathrm{EX}}_k\,\Delta t\,\mathbf{z}^{\mathrm{EX}} \\
&\texttt{end}
\end{aligned}
\tag{12}
$$

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, $\hat{\mathbf{x}}$ stores the solution of the embedded scheme and $\mathbf{x}$ is used to advance the solution of the main scheme. As in the three-register implementation of the two-register 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 IMEXRK[3R] schemes

Leveraging matrix inversion lemma, as done in the two-register implementation of the register scheme, it is possible to obtain a three-register implementation for the three-register scheme, as follows:

$$
\begin{aligned}
&\texttt{for}\quad k = 1 : s \\
&\quad \texttt{if}\quad k == 1, \quad \mathbf{y} = \mathbf{x}, \quad \mathbf{z} = \mathbf{x}, \quad \texttt{else} \\
&\qquad \mathbf{z} \leftarrow \mathbf{y} + a^{\mathrm{IM}}_{k,k-1}\,\Delta t\,A\,\mathbf{z} \\
&\qquad \mathbf{y} \leftarrow A^{-1}\,(\mathbf{z} - \mathbf{y})/(a^{\mathrm{IM}}_{k,k-1}\,\Delta t) \\
&\qquad \mathbf{z} \leftarrow \mathbf{z} + a^{\mathrm{EX}}_{k,k-1}\,\Delta t\,\mathbf{g}(\mathbf{y},\, t_n + c^{\mathrm{EX}}_{k-1}\Delta t) \\
&\qquad \mathbf{y} \leftarrow \mathbf{x} + (a^{\mathrm{IM}}_{k,k-1} - b^{\mathrm{IM}}_{k-1})\,\Delta t\,A\,\mathbf{y} + (a^{\mathrm{EX}}_{k,k-1} - b^{\mathrm{EX}}_{k-1})\,\Delta t\,\mathbf{g}(\mathbf{y},\, t_n + c^{\mathrm{EX}}_{k-1}\Delta t) \\
&\quad \texttt{end} \\
&\quad \mathbf{z} \leftarrow (I - a^{\mathrm{IM}}_{k,k}\,\Delta t\,A)^{-1}\,\mathbf{z} \\
&\quad \mathbf{x} \leftarrow \mathbf{x} + b^{\mathrm{IM}}_k\,\Delta t\,A\,\mathbf{z} + b^{\mathrm{EX}}_k\,\Delta t\,\mathbf{g}(\mathbf{z},\, t_n + c^{\mathrm{EX}}_k\Delta t) \\
&\quad \hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \hat{b}^{\mathrm{IM}}_k\,\Delta t\,A\,\mathbf{z} + \hat{b}^{\mathrm{EX}}_k\,\Delta t\,\mathbf{g}(\mathbf{z},\, t_n + c^{\mathrm{EX}}_k\Delta t) \\
&\texttt{end}
\end{aligned}
\tag{13}
$$

In this case, considering the possibility of storing the inverse of matrix A, one linear system, four matrix/vector products and three nonlinear function evaluations must be computed per stage, otherwise three linear systems, two matrix/vector products and three nonlinear function evaluations have to be performed, to which we must add one more matrix/vector product and one nonlinear function evaluation in case the embedded scheme is adopted for error control.

2 The classical 2nd-order, 3-stage, A-stable CN/RKW3 scheme

The classical second-order, A-stable CN/RKW3 method may easily be written in the low-storage IMEXRK Butcher tableaux form (7) (albeit with the $b_i^{\rm IM} = b_i^{\rm EX} = b_i$ constraint relaxed) with the four-stage IMEX Butcher tableaux

$$\begin{array}{r|cccc} 0 & 0 & & & \\ 8/15 & 4/15 & 4/15 & & \\ 2/3 & 4/15 & 1/3 & 1/15 & \\ 1 & 4/15 & 1/3 & 7/30 & 1/6 \\ \hline & 4/15 & 1/3 & 7/30 & 1/6 \end{array} \qquad \begin{array}{r|cccc} 0 & 0 & & & \\ 8/15 & 8/15 & 0 & & \\ 2/3 & 1/4 & 5/12 & 0 & \\ 1 & 1/4 & 0 & 3/4 & 0 \\ \hline & 1/4 & 0 & 3/4 & 0 \end{array} \tag{14}$$

A DIRK scheme with $c_1 = 0$ and $c_s = 1$ [such as that shown at left in (14)] 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 $b_s^{\rm EX} = 0$ 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 (14) are shown in Figures 1a-1b; 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 $z = z^{\rm EX}$) a stability function of

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

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

3 A simple 2nd-order, 2-stage implicit, 3-stage explicit, L-stable scheme

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, three-stage, 2-register alternative to CN/RKW3 which requires fewer flops per timestep to implement than CN/RKW3 and comes with a first-order embedded scheme and whose implicit part is L-stable, dubbed **IMEXRK23S[2R]L**, is given by[5]

$$\begin{array}{r|ccc} 0 & 0 & & \\ 2/5 & 0 & 2/5 & \\ 1 & 0 & 5/6 & 1/6 \\ \hline & 0 & 5/6 & 1/6 \\ \hline & 0 & 4/5 & 1/5 \end{array} \qquad \begin{array}{r|ccc} 0 & 0 & & \\ 2/5 & 2/5 & 0 & \\ 1 & 0 & 1 & 0 \\ \hline & 0 & 5/6 & 1/6 \end{array} \tag{15}$$

The explicit component of this scheme also satisfies the strong-stability preserving (SSP) property, under the condition

$$\Delta t \le c\, \Delta t_{FE}, \tag{16}$$

where Δt_{FE} is the maximum Δt allowed by a forward Euler discretization of the ODE (see [14] and [15] for a detailed description of strong-stability property). The coefficient c for strong stability in (16) is $c = 1$, which is the maximum attainable as proved in [5]. As noted in [6], a simplifying condition which, if $A^{\rm IM}$ is nonsingular, ensures that an A-stable DIRK scheme is in fact L-stable [i.e., that $\sigma(\infty) = 0$] is $a_{s,i} = b_i$

[5] For details on how this scheme was discovered, see §4, which applies the same techniques used to discover (15) to the 3rd-order, 3-stage implicit, 4-stage explicit, L-stable case.

for $i = 1, \ldots, s$; this condition is known as "stiff accuracy" [since $a^{\text{IM}}_{11} = 0$ in our schemes, A^{IM} is singular, and thus stiff accuracy alone does *not* ensure that an A-stable scheme is in fact L-stable; the stiff accuracy condition is still a useful simplifying assumption, however, as discussed further in §4—see (A)-(B) and surrounding discussion]. Applying stiff accuracy to (4a) and (6), it follows that $c_s = 1$. 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 condition, such as (15), are FSAL, and thus require only $s-1$ implicit solves per timestep. This is especially apparent in (15), in which the entire first column of the Butcher tableau of the implicit component equals zero.

The stability boundaries of the constituent DIRK and ERK components of (15) are shown in Figures 1c-1d.

The remainder of this paper focuses on third-order schemes of the two-register IMEXRK form given in (7).

4 Three 3rd-order, 3-stage implicit, 4-stage explicit, L-stable schemes

In order to achieve L-stability of the DIRK component, as noted in §3 and [6], a useful (but, for singular A^{IM}, not sufficient) simplifying assumption is the "stiff accuracy" condition $a_{s,i} = b_i$ for $i = 1, \ldots, s$ [and hence, by (4a) and (6), $c_s = 1$]. Taking $s = 4$ and defining $a^{\text{IM}}_{ii} = \alpha_i$ for $i = 2, 3$, the Butcher tableaux (7) reduce to the following form (with, again, an FSAL implicit part):

$$\begin{array}{c|cccc} 0 & 0 & & & \\ c_2 & c_2 - \alpha_2 & \alpha_2 & & \\ c_3 & b_1 & c_3 - b_1 - \alpha_3 & \alpha_3 & \\ 1 & b_1 & b_2 & b_3 & b_4 \\ \hline & b_1 & b_2 & b_3 & b_4 \\ & \hat{b}^{\text{IM}}_1 & \hat{b}^{\text{IM}}_2 & \hat{b}^{\text{IM}}_3 & \hat{b}^{\text{IM}}_4 \end{array} \qquad \begin{array}{c|cccc} 0 & 0 & & & \\ c_2 & c_2 & 0 & & \\ c_3 & b_1 & c_3 - b_1 & 0 & \\ 1 & b_1 & b_2 & 1 - b_1 - b_2 & 0 \\ \hline & b_1 & b_2 & b_3 & b_4 \\ & \hat{b}^{\text{EX}}_1 & \hat{b}^{\text{EX}}_2 & \hat{b}^{\text{EX}}_3 & \hat{b}^{\text{EX}}_4 \end{array} \tag{17a}$$

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 (17a) is motivated. To understand this simplification, we may rewrite the stability function of the scheme as a rational function of z^{IM} and z^{EX}, as suggested by (5) and (9), as

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

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., $\lim_{z^{\text{IM}} \to \infty} \sigma(z^{\text{IM}}; z^{\text{EX}}) = 0$], it is sufficient to impose that $q_3 = \alpha_2 \, \alpha_3 \, b_4 \neq 0$ and

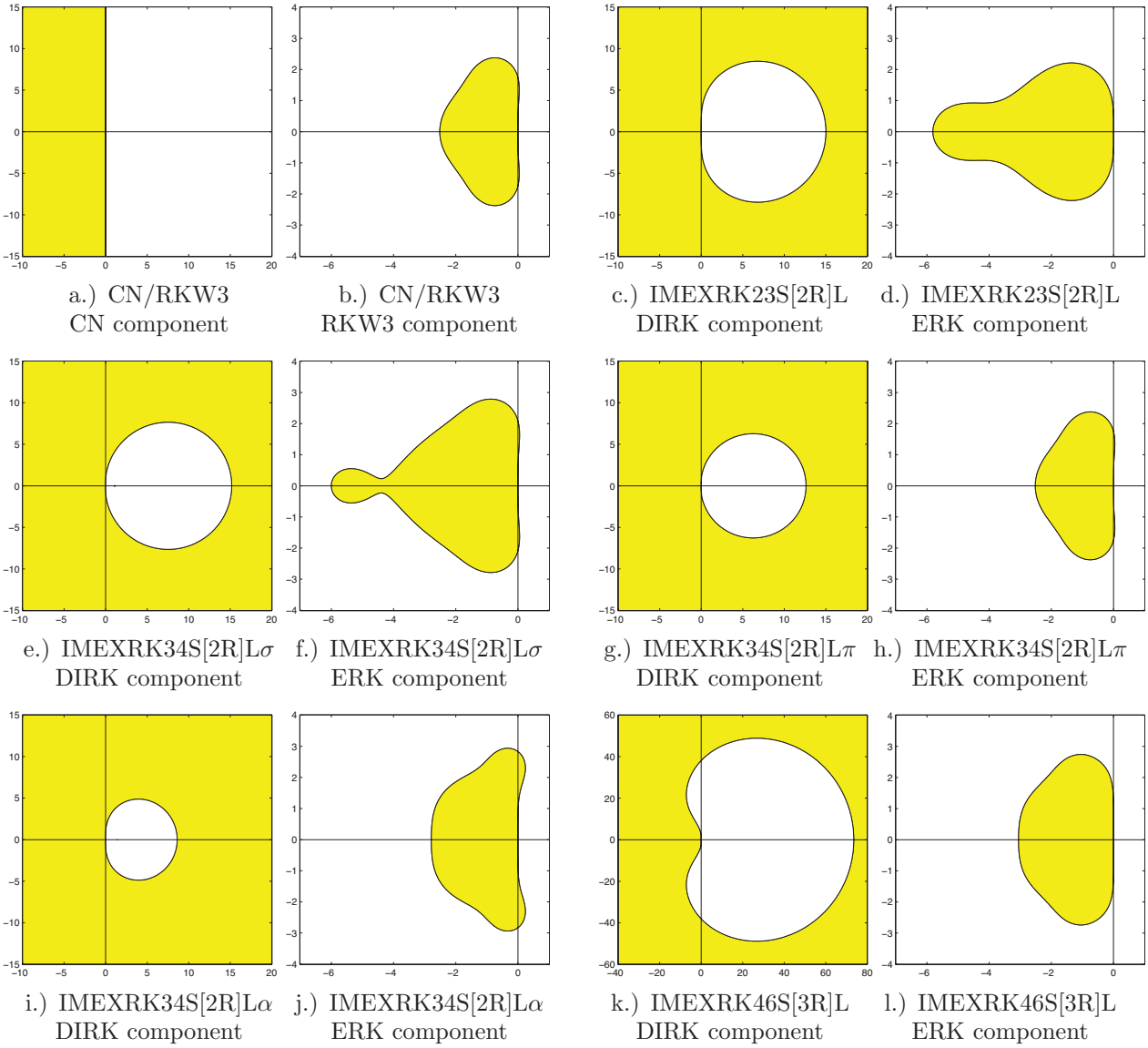
$$\tau_1^{L\text{-stability}} = \hat{p}_{30} = -\alpha_2 \, \alpha_3 \, b_1 - \alpha_2 \, \alpha_3 \, b_2 - \alpha_2 \, \alpha_3 \, b_3 + \alpha_3 \, b_2 \, c_2 + \alpha_3 \, b_3 \, c_2 + b_1 \, b_3 \, c_2 + \alpha_2 \, b_3 \, c_3 - b_3 \, c_2 \, c_3 = 0, \tag{A}$$

$$\begin{aligned} \tau_2^{L\text{-stability}} = \hat{p}_{31} = &-\alpha_2 \, \alpha_3 \, b_4 + \alpha_3 \, b_4 \, c_2 + b_1 \, b_4 \, c_2 - \alpha_3 \, b_1 \, b_4 \, c_2 - b_1^2 \, b_4 \, c_2 - b_1 \, b_2 \, b_4 \, c_2 + \alpha_2 \, b_4 \, c_3 \\ &- \alpha_2 \, b_1 \, b_4 \, c_3 - \alpha_2 \, b_2 \, b_4 \, c_3 - b_4 \, c_2 \, c_3 + b_1 \, b_4 \, c_2 \, c_3 + b_2 b_4 \, c_2 \, c_3 = 0. \end{aligned} \tag{B}$$

As noted [6] and [7], suppressing the first column of the DIRK component, by imposing $b_1 = 0$ and $\alpha_2 = c_2$ in (17a), satisfies both (A) and (B) identically; we thus incorporate these additional simplifications in the two subsections that follow.

4.1 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



a.) CN/RKW3 CN component
b.) CN/RKW3 RKW3 component
c.) IMEXRK23S[2R]L DIRK component
d.) IMEXRK23S[2R]L ERK component
e.) IMEXRK34S[2R]Lσ DIRK component
f.) IMEXRK34S[2R]Lσ ERK component
g.) IMEXRK34S[2R]Lπ DIRK component
h.) IMEXRK34S[2R]Lπ ERK component
i.) IMEXRK34S[2R]Lα DIRK component
j.) IMEXRK34S[2R]Lα ERK component
k.) IMEXRK46S[3R]L DIRK component
l.) IMEXRK46S[3R]L ERK component

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

ERK component as

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

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

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

For

$$\delta > \delta_{\mathrm{crit}} = \left(139 - 5255/\sqrt[3]{-210253 + 60928\sqrt{51}} + \sqrt[3]{-210253 + 60928\sqrt{51}}\right)/6144 = 0.0184557,$$

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

Parametric variation reveals that the extent of the stability region along the imaginary axis is relatively insensitive to changes in δ. 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 [7], has a maximum extent along the negative real axis which is $\sim 40\%$ *less* than that of that of the present scheme, and a maximum extent along the imaginary axis which is only $\sim 5\%$ 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

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

Finding solution(s) of such a set of six nonlinear constraint equations is difficult even for commercially available numerical solvers. For this reason, the solution of this nonlinear system has been recast as a global optimization problem leveraging the Delaunay-based Derivative-free Optimization via Global Surrogate (ΔDOGS) algorithm developed by our group. After assembling the unknowns $\{\alpha_3, b_2, b_3, b_4, c_2, c_3\}$ as a parameter vector $\mathbf{x}$, a cost function $J(\mathbf{x})$ is defined by summing the square of the LHS of each of the six constraints. Thus, though nonconvex, $J(\mathbf{x}) \geq 0$, and $J(\mathbf{x}) = 0$ corresponds to a solution. The domain searched is $\{\alpha_i, b_i\} \in [-1, 1]$ and $\{c_i\} \in [0, 1]$. Our optimization routine, which will be discussed elsewhere, finds several local minima with $J(\mathbf{x}) > 0$, but finally leads to a solution for which $J(\mathbf{x}) = 0$, dubbed **IMEXRK34S[2R]L**σ, given by

$$\begin{gathered}
\alpha_2 = 0.7458175396027730, \quad \alpha_3 = 0.6206610736335834, \\
b_1 = 0, \quad b_2 = 0.2885514426131443, \quad b_3 = 0.5784565900123583, \quad b_4 = 0.1329919673744975, \\
c_2 = 0.7458175396027730, \quad c_3 = 0.2624247147805739.
\end{gathered} \tag{17b}$$

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

$$\begin{gathered}
\hat{b}_1^{\mathrm{IM}} = 0, \quad \hat{b}_2^{\mathrm{IM}} = 0.33510152222762435, \quad \hat{b}_3^{\mathrm{IM}} = 0.5624145479249864, \quad \hat{b}_4^{\mathrm{IM}} = 0.10248392984738919, \\
\hat{b}_1^{\mathrm{EX}} = 0.3889537200272892, \quad \hat{b}_2^{\mathrm{EX}} = 0, \quad \hat{b}_3^{\mathrm{EX}} = 0.15055585809070993, \quad \hat{b}_4^{\mathrm{EX}} = 0.4604904218820009
\end{gathered}$$

The stability boundaries of the constituent DIRK and ERK components are shown in Figures 1e-1f. Moreover this scheme is SSP under condition (16) with $c = 0.7027915$. This result can be improved up to $c = 0.7703947$, which is achieved by replacing condition (C) with

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

However, this constraint does not lead to an IMEXRK scheme with L-stable implicit component, but it is possible to choose a positive δ small enough to guarantee L-stability and a nearly optimal value c for strong stability. Choosing $\delta = 1/10000$, for example, gives a scheme, named **IMEXRK34S[2R]L**π, with the following choice of parameters:

$$\begin{gathered}
\alpha_2 = 0.8920138295341937, \quad \alpha_3 = 0.7118592498085877, \\
b_1 = 0, \quad b_2 = 0.3507710822962850, \quad b_3 = 0.6486283917251868, \quad b_4 = 0.0006005259785281534, \\
c_2 = 0.8920138295341937, \quad c_3 = 0.2875403235378705,
\end{gathered} \tag{17c}$$

and the associated second-order embedded scheme:

$$\begin{gathered}
\hat{b}_1^{\mathrm{IM}} = 0, \quad \hat{b}_2^{\mathrm{IM}} = 0.35101071959085495, \quad \hat{b}_3^{\mathrm{IM}} = 0.6485920703520673, \quad \hat{b}_4^{\mathrm{IM}} = 0.0003972100570779, \\
\hat{b}_1^{\mathrm{EX}} = 0.4996459562094747, \quad \hat{b}_2^{\mathrm{EX}} = 0, \quad \hat{b}_3^{\mathrm{EX}} = 0.0004969316892197, \quad \hat{b}_4^{\mathrm{EX}} = 0.4998571121013055
\end{gathered}$$

The coefficient for strong stability is $c = 0.7701444$. The stability boundaries of the associated DIRK and ERK components are shown in Figures 1g-1h. Since δ 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.

4.2 Maximizing accuracy of the ERK component

An alternative third-order four-stage 2-register L-stable strategy, with closed-form parameter values and a stability region for the ERK part which coincides with the stability region of the standard 4-step RK4 scheme, is given by replacing the final constraint, (C), with

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

which results in a scheme, dubbed **IMEXRK34S[2R]L**α, given by

$$\begin{array}{r|cccc} 0 & 0 & & & \\ 1/3 & 0 & 1/3 & & \\ 1 & 0 & 1/2 & 1/2 & \\ 1 & 0 & 3/4 & -1/4 & 1/2 \\ \hline & 0 & 3/4 & -1/4 & 1/2 \end{array} \qquad \begin{array}{r|ccccc} 0 & 0 & & & & \\ 1/3 & 1/3 & 0 & & & \\ 1 & 0 & 1 & 0 & & \\ 1 & 0 & 3/4 & 1/4 & 0 & \\ \hline & 0 & 3/4 & -1/4 & 1/2 \end{array} \tag{18}$$

A second-order embedded scheme having all third-order truncation terms could not be achieved because of assumption (C"). Moreover, since $b_3 < 0$ the scheme is not SSP. The stability boundaries of the constituent DIRK and ERK components are shown in Figures 1i-1j; IMEXRK34S[2R]Lα has improved accuracy but reduced stability on the negative real axis as compared with IMEXRK34S[2R]Lσ.

5 A 4th-order, 6-stage, L-stable scheme

Solving the nonlinear equations associated with the requirement of fourth-order accuracy is a difficult task. 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 invoke the so-called simplifying assumptions, as described in [6]. In particular, we restrict such assumptions to the parameters of the implicit component only, by setting:

$$\sum_{j=1}^{s} a_{ij}^{\mathrm{IM}}\, c_j = c_i^2 ?2, \quad i = 2,\, 3,\, \ldots,\, s-1 \tag{19}$$

which is equivalent to require stage-order two for the implicit component. This reduces the number of nonlinear equations from fourteen, i.e. one for first order, one for second, three for third 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. Using a six-stage 3-register IMEXRK scheme, we have 30 degrees of freedom to satisfy 25 constraints. Since determining the c_i coefficients can be troublesome since we want them to be inside the range [0, 1], we exploit the remaining degrees of freedom to impose such coefficients. In particular, we set:

$$c_1 = 0, \quad c_2 = 1/10, \quad c_3 = 2/5, \quad c_4 = 3/5, \quad c_5 = 9/10, \quad c_6 = 1. \tag{20}$$

As done in §4, we leverage ΔDOGS algorithm to find a solution to the system of nonlinear equations. The scheme hence developed, dubbed **IMEXRK46S[3R]L** is reported below:

$$\begin{array}{r|cccccc} 0 & 0 & & & & & \\ c_2 & a_{21}^{\mathrm{IM}} & a_{22}^{\mathrm{IM}} & & & & \\ c_3 & a_{31}^{\mathrm{IM}} & a_{32}^{\mathrm{IM}} & a_{33} & & & \\ c_4 & b_1 & a_{42}^{\mathrm{IM}} & a_{43}^{\mathrm{IM}} & a_{44}^{\mathrm{IM}} & & \\ c_5 & b_1 & b_2 & a_{53}^{\mathrm{IM}} & a_{54}^{\mathrm{IM}} & a_{55}^{\mathrm{IM}} & \\ 1 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \\ \hline & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{array} \qquad \begin{array}{r|cccccc} 0 & 0 & & & & & \\ c_2 & a_{21}^{\mathrm{EX}} & 0 & & & & \\ c_3 & a_{31}^{\mathrm{EX}} & a_{32}^{\mathrm{EX}} & 0 & & & \\ c_4 & b_1 & a_{42}^{\mathrm{EX}} & a_{43}^{\mathrm{EX}} & 0 & & \\ c_5 & b_1 & b_2 & a_{53}^{\mathrm{EX}} & a_{54}^{\mathrm{EX}} & 0 & \\ 1 & b_1 & b_2 & b_3 & a_{64}^{\mathrm{EX}} & a_{65}^{\mathrm{EX}} & 0 \\ \hline & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{array} \tag{21}$$

where

$$a^{\rm IM}_{31} = 0.16036818466407831073, \quad a^{\rm IM}_{32} = 0.05284242044789558570, \quad a^{\rm IM}_{33} = 0.18678939488026103575,$$
$$a^{\rm IM}_{42} = 0.26765292855424752582, \quad a^{\rm IM}_{43} = -0.4806631563015242346, \quad a^{\rm IM}_{44} = 0.57583328277530823545,$$
$$a^{\rm IM}_{53} = 2.4049192562328432369, \quad a^{\rm IM}_{54} = -3.0133537881037294103, \quad a^{\rm IM}_{55} = 1.4048985145990107267,$$
$$a^{\rm EX}_{31} = -0.28122430371955223659, \quad a^{\rm EX}_{32} = 0.68122430371955223659,$$
$$a^{\rm EX}_{42} = -0.18908270367987563237, \quad a^{\rm EX}_{43} = 0.55190575870790715902,$$
$$a^{\rm EX}_{53} = -0.18135366450888254458, \quad a^{\rm EX}_{54} = 0.97781764723700709797,$$
$$a^{\rm EX}_{64} = 0.20444384824133449118, \quad a^{\rm EX}_{65} = 0.30254485081172593969,$$
$$b_1 = 0.23717694497196847336, \quad b_2 = -0.13364092770009302675, \quad b_3 = 0.38947528367506412252,$$
$$b_4 = 0.41044138083424541514, \quad b_5 = -0.14761832580621388850, \quad b_6 = 0.24416564402502890423.$$

The stability boundaries of the DIRK and ERK components are shown in Figures 1k-1l. As we can observe, the scheme is only $L(\alpha)$ stable with $\alpha = 70°$. Furthermore, the scheme is not SSP.

What follows is a thorough analysis of the computational cost of the schemes here developed applied to a model problem, in comparison with the state-of-the-art CN/RKW3.

6 Application to a model problem

To illustrate the relative computational costs of our new low-storage IMEXRK schemes on a representative PDE model problem discretized on $N \gg 1$ gridpoints, we now compare the three- and two-register implementations, (10) and (11), of each of the methods developed herein to CN/RKW3 and five full-storage IMEX Runge-Kutta schemes available in literature, implemented as in (3). We consider as a model PDE problem the one-dimensional Kuramoto Sivashinsky equation

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4} \tag{22}$$

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. The RHS of (22) consists of a nonlinear convective term, treated explicitly, and two linear derivative terms, treated implicitly.

Following a five-point central finite-difference approach on a uniform grid, (22) 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 (22), 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 (10) of the CN/RKW3 method (14), $6N$ 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 approach, with nonlinear products computed in physical space and spatial derivatives computed in Fourier space, (22) can be written in wavenumber space as

$$\frac{d\hat{u}_n}{dt} = -ik_{x_n}\widehat{(u \cdot u)}_n + (k^2_{x_n} - k^4_{x_n})\hat{u}_n \tag{23}$$

where $i = \sqrt{-1}$, $k_{x_n} = 2\pi n/L$ is the wavenumber, and $\widehat{(u \cdot u)}_n$ denotes the n'th wavenumber component of the function computed by transforming u to physical space on $N = 2^p$ equispaced gridpoints, computing $u \cdot u$ 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 other schemes may be counted similarly; results are summarized in Tables 1 and 2. It is seen that, if computational cost is naively 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-register and three-register implementations, however, bestows them with a distinct advantage for high-dimensional ODE discretizations of PDE systems. Further specifics of the comparisons between our low-storage IMEXRK schemes and CN/RKW3 are discussed in §7.

7 Conclusions

We have developed five new IMEX Runge-Kutta schemes with low-storage requirements:

(A) IMEXRK23S[2R]L (15) is a simple SSP second-order, two-stage implicit, three-stage explicit, L-stable scheme with closed-form parameter values.

(B) IMEXRK34S[2R]Lσ (17b) is a SSP third-order, three-stage implicit, four-stage explicit, L-stable scheme with parameter values found numerically to maximize the domain of stability of the ERK component on the negative real axis.

(C) IMEXRK34S[2R]Lπ (17c) is a SSP third-order, three-stage implicit, four-stage explicit, L-stable scheme with parameter values found numerically in order to optimize the CFL limit for strong stability.

(D) IMEXRK34S[2R]Lα (18) is a non-SSP third-order, three-stage implicit, four-stage explicit L-stable scheme with closed-form parameter values selected to maximize the accuracy of the ERK component of the scheme.

(E) IMEXRK46S[3R]L (21) is a non-SSP fourth-order, six-stage L-stable scheme with parameter values found numerically.

Various properties of these schemes, and some competing full-storage IMEXRK scheme, are given in Tables 1 and 2. The particular measure of truncation error of a scheme of order q used in the tables, adapted from [7], is

$$A^{(q+1)} = \sqrt{\sum\nolimits_i \left(\tau_i^{\mathrm{IM}(q+1)}\right)^2 + \sum\nolimits_i \left(\tau_i^{\mathrm{EX}(q+1)}\right)^2 + \sum\nolimits_i \left(\tau_i^{\mathrm{IMEX}(q+1)}\right)^2}.$$

In comparison with the venerable CN/RKW3 scheme,

- All our second- and third-order schemes, like CN/RKW3, admit both two-register and three-register implementations, with the three-register implementations requiring fewer flops.
- Scheme (A) is the same order of accuracy as CN/RKW3 (second), while the remaining schemes are a higher order of accuracy (third).
- Schemes (A), (B), (C) and (D) schemes are L-stable (note that CN/RKW3 is only A-stable).
- Schemes (A) generally requires fewer floating-point operations per timestep than CN/RKW3, whereas schemes (B), (C), & (D) generally require slightly more.
- Schemes (E) requires more floating point operations and one more register with respect to CN/RKW3, while it benefits from higher accuracy and better stability properties.

As for future developments, an analysis of the order reduction of the present IMEXRK schemes applied to our test ODE is highly desired in order to check their robustness with respect to stiff problems. Implementation of all of the present schemes into our benchmark DNS code is also underway.

References

[1] K. Akselvoll, and & 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.

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

[3] J.C. Butcher *Numerical methods for ordinary differential equations*, Wiley, 2008.

[4] M.P. Calvo, J. de Frutos & J. Novo, *Linearly implicit RungeKutta methods for advection-reaction-diffusion equations*, Appl. Num. Math. 37 (4), p. 535-549, 2001.

[5] S. Gottlieb, C.W. Shu, E. Tadmor, *Strong-stability-preserving high order time discretization methods*, SIAM Review 43, p. 89-112, 2001.

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

[7] C.A. Kennedy, M.H. Carpenter, & R.M. Lewis, *Additive Runge-Kutta Schemes for Convection-Diffusion-Reaction Equations*, Appl. Num. Math., 44, p. 139-181, 2003.

[8] C.A. Kennedy, M.H. Carpenter & R.M. Lewis, *Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations*, Appl. Num. Math., 35, p. 177-219, 2000.

[9] J. Kim & P. Moin, *Application of a Fractional-Step Method to Incompressible Navier-Stokes Equations*, J. Comput. Phys., 59, p. 308-323, 1985.

[10] J. Kim, P. Moin, & B. Moser *Turbulence statistics in fully developed channel flow at low Reynolds number*, J. Fluid Mech., 177, p. 133-166, 1987.

[11] H. Le & P. Moin *An improvement of fractional step methods for the incompressible Navier-Stokes equations*, J. Comput. Phys., 92, p. 369-379, 1991.

[12] L. Pareschi & G. Russo, *Implicit-explicit Runge-Kutta schemes and applications to hyperbolic sytems with relaxation*, Journal of Scientific Computing 25, p. 129-155, 2003.

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

[14] C.W. Shu, *Total-variation-diminishing time discretizations*, SIAM J. Sci. Statist. Comput., 9, p. 1073-1084, 1988.

[15] C.W. Shu & S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys., 77, p.439-471, 1988.

[16] P. van der Houwen, *Explicit Runge-Kutta formulas with increased stability boundaries*, Numerische Mathematik 20, p. 149-164, 1972.

[17] J.H. Williamson, *Low-storage Runge-Kutta schemes*, J. Comput. Phys. 35, p. 4856, 1980.

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

Scheme:	CN/RKW3	IMEXRK23S[2R]L	IMEXRK34S[2R]Lσ IMEXRK34S[2R]Lπ IMEXRK34S[2R]Lα	IMEXRK46S[3R]L
Accuracy	second-order	second-order	third-order	fourth-order
Stability of DIRK part	A-stable	L-stable	L-stable	$L(\alpha)$-stable, $\alpha = 70^\circ$
Stability of ERK part on negative real axis	$-2.51 \le z^{\mathrm{EX}} \le 0$	$-5.81 \le z^{\mathrm{EX}} \le 0$	σ: $-6.00 \le z^{\mathrm{EX}} \le 0$ π: $-2.52 \le z^{\mathrm{EX}} \le 0$ α: $-2.79 \le z^{\mathrm{EX}} \le 0$	$-2.96 \le z^{\mathrm{EX}} \le 0$
$\sigma(z^{\mathrm{IM}} \to \infty; z^{\mathrm{EX}})$	-1	0	0	0
SSP	-	Yes	σ: Yes π: Yes α: No	No
Embedded Scheme	No	Yes	σ: Yes π: Yes α: No	No
Truncation error	$A^{(3)} = 0.0387$	$A^{(3)} = 0.114$	σ: $A^{(4)} = 0.113$ π: $A^{(4)} = 0.207$ α: $A^{(4)} = 0.0824$	$A^{(5)} = 0.123$
Cost - Finite Difference	$115N$ flops (3-reg) $127N$ flops (2-reg)	$90N$ flops (3-reg) $101N$ flops (2-reg)	$133N$ flops (3-reg) $157N$ flops (2-reg)	$264N$ flops (4-reg) $504N$ flops (3-reg)
Cost - Pseudospectral	6 FFTs (3-reg)	6 FFTs (3-reg)	8 FFTs (3-reg)	12 FFTs (4-reg)

Table 1: Summary of the two- and three-register IMEXRK schemes considered in this paper, and their leading-order computational cost per timestep for efficient implementation (using $R+1$ or R registers in the implementation) on the 1D KS equation.

Scheme:	Ascher(2, 3, 3)	Ascher(3, 4, 3)	Ascher(4, 4, 3)	LIRK3	ARK3(2)4L[2]SA
Accuracy	third-order	third-order	third-order	third-order	third-order
Stability of DIRK part	strongly A-stable	L-stable	L-stable	L-stable	L-stable
Stability of ERK part on negative real axis	$-2.51 \le z^{\mathrm{EX}} \le 0$	$-2.78 \le z^{\mathrm{EX}} \le 0$	$-2.14 \le z^{\mathrm{EX}} \le 0$	$-2.21 \le z^{\mathrm{EX}} \le 0$	$-3.66 \le z^{\mathrm{EX}} \le 0$
$\sigma(z^{\mathrm{IM}} \to \infty; z^{\mathrm{EX}})$	$-0.732 - 0.732\, z^{\mathrm{EX}}$	$0.106\, z^{\mathrm{EX}}$	0	0	0
SSP	-	-	No	No	No
Embedded scheme	No	No	No	No	Yes
Truncation error	$A^{(4)} = 0.206$	$A^{(4)} = 0.103$	$A^{(4)} = 0.163$	$A^{(4)} = 0.100$	$A^{(4)} = 0.0722$
Number of registers used in implementation	7	9	10	9	10
Cost - Finite Difference	$92N$ flops	$141N$ flops	$190N$ flops	$139N$ flops	$159N$ flops
Cost - Pseudospectral	6 FFTs	8 FFTs	8 FFTs	8 FFTs	8 FFTs

Table 2: Some competing full-storage third-order IMEX schemes, and their leading-order computational cost per timestep for efficient implementation on the 1D KS equation. The Ascher schemes are from [2], the LIRK3 scheme is from [4], and the ARK3(2)4L[2]SA scheme is from [7].