2015

A Class of High-Order Runge-Kutta-Chebyshev Stability Polynomials

Stephen O'Sullivan
Technological University Dublin, stephen.osullivan@tudublin.ie

Recommended Citation

O'Sullivan,S.(2015) A class of high-order Runge-Kutta-Chebyshev stability polynomials, *Journal of Computational Physics* doi: 10. 1016/ j. jcp. 2015. 07. 050

A class of high-order Runge-Kutta-Chebyshev stability polynomials

Stephen O'Sullivan

School of Mathematical Sciences, Dublin Institute of Technology, Dublin 8, Ireland

Abstract

The analytic form of a new class of factorized Runge-Kutta-Chebyshev (FRKC) stability polynomials of arbitrary order N is presented. Roots of FRKC stability polynomials of degree $L = MN$ are used to construct explicit schemes comprising L forward Euler stages with internal stability ensured through a sequencing algorithm which limits the internal amplification factors to $\sim L^2$. The associated stability domain scales as M^2 along the real axis. Marginally stable real-valued points on the interior of the stability domain are removed via a prescribed damping procedure.

By construction, FRKC schemes meet all linear order conditions; for nonlinear problems at orders above 2, complex splitting or Butcher group composition methods are required. Linear order conditions of the FRKC stability polynomials are verified at orders 2, 4, and 6 in numerical experiments. Comparative studies with existing methods show the second-order unsplit FRKC2 scheme and higher order (4 and 6) split FRKC schemes are efficient for large moderately stiff problems.

Keywords: Stiff equations, Stability and convergence of numerical methods, Method of lines
2010 MSC: 65L04, 65L20, 65M20

1. Introduction

Runge-Kutta-Chebyshev methods are explicit numerical integration schemes with extended stability domains derived from the optimality properties of Chebyshev polynomials [1, 2]. These methods are commonly applied to moderately stiff systems of semi-discrete equations of the form

$$w' = f(t, w) \tag{1}$$

yielding an approximate solution w^n at time $t^n = nT$ defined on a spatial mesh of spacing h at points x_k with $x_{k+1} = x_k + h$. Such systems arise naturally through application of the method of lines to parabolic systems. Runge-Kutta-Chebyshev methods may be broadly categorized as factorized or recursive in nature.

Factorized Runge-Kutta-Chebyshev methods are formed from a sequence of forward Euler stages. These methods were first suggested by Saulev [3], Guillou and Lago [4] and were subsequently considered by Gentzsch and Schluter [5] and van der Houwen [6]. They have been applied at

Email address: stephen.osullivan@dit.ie (Stephen O'Sullivan)

Journal of Computational Physics *Submitted February 12, 2015* *Accepted July 22, 2015*

doi: 10.1016/j.jcp.2015.07.050

first-order and extended to second-order by Richardson extrapolation by various authors [7, 8, 9, 10]. Based on a strategy proposed by Lebedev [11], the DUMKA stability polynomials exist at orders 2, 3, and 4 [12].

Recursive Runge-Kutta-Chebyshev methods were first described by van Der Houwen and Sommeijer [13] and rely on (three-term) recursion to generate a solution. They were introduced at second-order by van Der Houwen and Sommeijer [13], and subsequently, other second, third, and fourth-order methods have been developed [14, 15, 16, 17, 18]. For orders above 2, composition methods are typically employed [see, for example, 17]. We note that alternative approaches with second-order accuracy involving Legendre polynomials have recently been proposed by Meyer et al. [19, 20]. At orders above 2, for both factorized and recursive methods, composition techniques relying on B-series theory [21, 22] are used to satisfy the full set of order conditions [12, 17].

This paper is organized as follows. In Section 2, the analytic form of the class of FRKC stability polynomials is presented. The construction of stable time-marching schemes based on the roots of these polynomials is outlined. Section 3 is given over to the derivation of the polynomial through consideration of associated recurrence relations. In Section 4, numerical tests are presented confirming the order and efficiency properties of FRKC methods. Conclusions are presented in Section 5.

2. High-order factorized Runge-Kutta-Chebyshev

2.1. General prescription

Eq. 1 may be written in autonomous form by appending t to vector of dependent variables for the system

$$w' = f(w). \tag{2}$$

Parentheses are used in the remainder of this work to differentiate exponents from indices, unless no ambiguity exists. We proceed by considering order N extended stability explicit Runge-Kutta schemes over $L = MN$ stages of the form

$$W^L = W^0 + T\sum_{l=1}^{L} a_l f(W^{l-1}), \tag{3}$$

where $W^0 = w^n$ corresponds to the approximate solution w^n at time level n, and W^L yields w^{n+1} at a time T later. The timestep related to each stage is then given by $\tau_l = a_l T$. The FRKC polynomial of rank N, and degree $L = MN$, is given by

$$B_M^N(z) = d_0^N + 2\sum_{k=1}^{N} d_k^N C_{kM}(z), \tag{4}$$

where C_{kM} denotes the the Chebyshev polynomial of the first kind of degree kM. The corresponding optimal real stability range is $[-\beta_M, 0]$, where $\beta_M = 2M^2\alpha_M$ with $\alpha_M = (\gamma_M N + 2)/3$, and $\gamma_2 \approx 0.87$, with γ_M rapidly converging to 1 with increasing M. In this limit, the polynomials generate 81%, 74% and 73% of the optimal intervals for order 2, 4, 6 respectively (see Van Der Houwen [23] and Abdulle [24] for estimates of the optimal values for α_M). The limiting step size is $T = \beta_M/|\lambda|_{\max}$, where λ are the negative-definite eigenvalues for the Jacobian of Eq. 2. We note that the form of Eq. 4 is consistent with the known result that Chebyshev expansions of stability polynomials to arbitrary order exist [25]. Furthermore, following from a proposition by Lomax [26], Riha [27]

confirmed the existence and uniqueness of optimal stability polynomials with $L-N$ local maxima in magnitude with value unity. A full derivation of the FRKC polynomial expression given by Eq. 4 is provided in Section 3.

The order coefficients d_k^N, which we refer to collectively as the order *pattern*, are determined by requiring that the (undamped) stability polynomial $R_M^N(z) = B_M^N(1 + z/M^2\alpha_M)$, consisting of shifted Chebyshev polynomials, satisfies the linear order conditions

$$R_M^{N\,(n)}(0) = 1 \quad n = 1, \cdots, N. \tag{5}$$

This requirement is met by solving the N-dimensional linear system[1]

$$\begin{bmatrix} C_M^{(1)}(1) & \dots & C_{NM}^{(1)}(1) \\ \vdots & \ddots & \vdots \\ C_M^{(N)}(1) & \dots & C_{NM}^{(N)}(1) \end{bmatrix} \begin{bmatrix} d_1^N \\ \vdots \\ d_N^N \end{bmatrix} = \begin{bmatrix} (M^2\alpha_M)^1 \\ \vdots \\ (M^2\alpha_M)^N \end{bmatrix}, \tag{6}$$

coupled with the conservation constraint

$$d_0^N = 1 - 2\sum_{k=1}^{N} d_k^N. \tag{7}$$

Following identification of the roots ζ_l of the FRKC polynomial $B_M^N(z)$, the damped order N scheme corresponding to Eq. 3 is determined by using

$$a_l = \frac{1}{M^2\alpha_M}\ \frac{1}{1-\zeta_l}. \tag{8}$$

In order to ensure a stable scheme for small perturbations from the real axis in the spectrum of Eq. 2, it is necessary to introduce a suitable damping procedure. We find an effective prescription for the damped order N scheme is given by

$$a_l = \frac{1}{(1-\nu)M^2\alpha_M}\ \frac{1-\mu_l}{1-(1-2\mu_l)\zeta_l}, \tag{9}$$

where the damping is parameterized by the small positive quantity ν, resulting in the real extent of the stability interval being reduced to $(1-\nu)\beta_M$. The value of $\nu = \nu_0/N$ is regulated by means of the reference damping parameter ν_0 such that maxima in $|R|$ along the real axis are scaled by approximately $1-\nu_0$.

For the case $\nu_0 = 0.05$, with $M = 20$, and for various values of N, Figs 1 and 2 illustrate the effect of the damping procedure. It is clear that the undamped polynomials are marginally stable at $M-1$ points on the interior of the stability domain along the real axis. (In fact, for sub-optimal α_M, internal marginally stable points occur at $M/2-1$ locations for even values of M, or $(M-1)/2$ locations for odd values of M.) Examples of the order patterns for $M = 20$ with $N = 2$, 4, 6 are given in Appendix A.

The L-tuple $[\mu_l]$ has cardinality N and regulates the implementation of damping in the scheme while preserving the nominal order of accuracy. The values of μ_l are obtained by tuning the damped stability polynomial $R_M^N(z) = \Pi_{l=1}^{L}(1 + a_l z)$ to meet the linear order conditions given in Eq. 5. We describe the procedure for the determination of the damping coefficients μ_l in Section 2.2.

[1]The identity $C_{kM}^{(l)}(1) = \prod_{i=0}^{l-1}((kM)^2 - i^2)/(2i+1)$ is useful here.

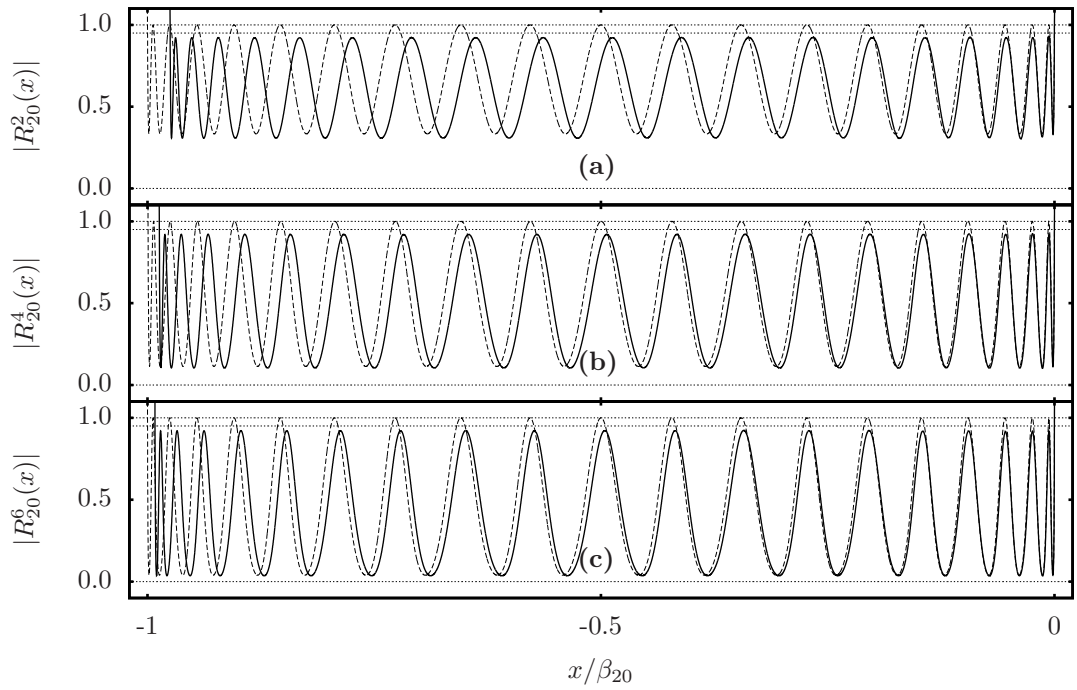

Figure 1: Absolute values along the real axis for FRKC stability polynomials corresponding to $M = 20$ at various values of N. Panels show plots of $|R^N_M(x)|$ ($x \in \mathbb{R}$): **(a)** $|R^2_{20}(x)|$; **(b)** $|R^4_{20}(x)|$; **(c)** $|R^6_{20}(x)|$. Solid lines indicate damped polynomials with $\nu_0 = 0.05$; dashed lines correspond to the associated undamped polynomials. For $N = 2$: $\gamma_{20} = 0.9988$, $\beta_{20} = 1066.0$; for $N = 4$: $\gamma_{20} = 1.0215$, $\beta_{20} = 1623.9$; for $N = 6$: $\gamma_{20} = 1.0276$, $\beta_{20} = 2177.5$. Dotted lines indicate guide values at 1.0, 0.95, 0.0.

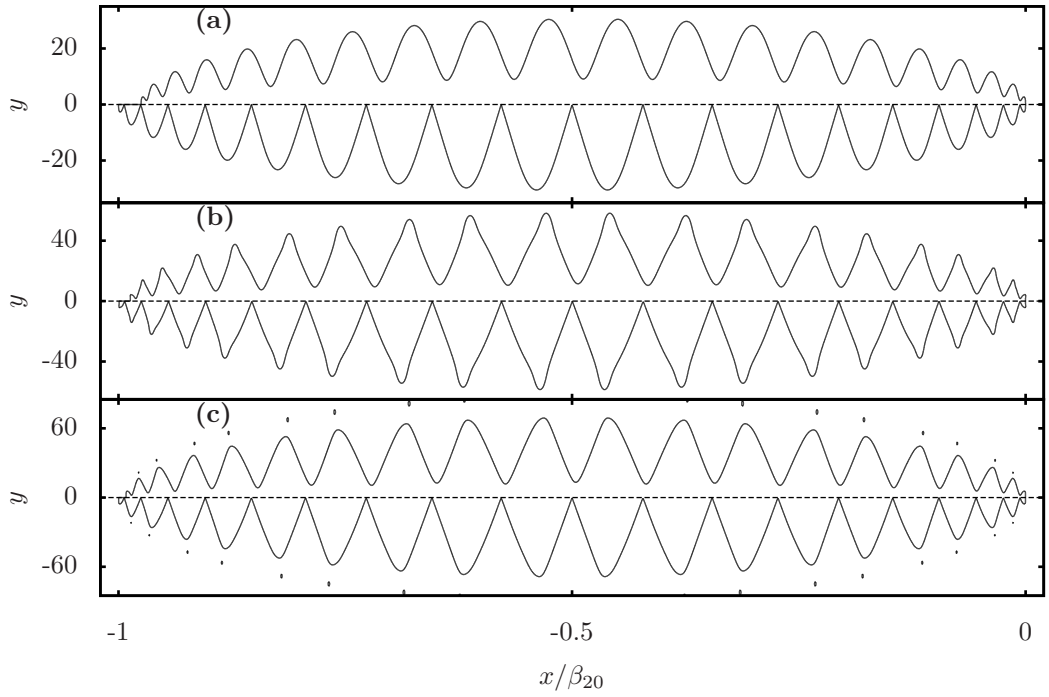

Figure 2: Stability domains for FRKC stability polynomials corresponding to $M = 20$ at various values of N. Damped polynomials with $\nu_0 = 0.05$ are represented for $y > 0$; associated undamped polynomials are illustrated for $y \leq 0$. The panels show $|R_M^N| = 1$: **(a)** $|R_{20}^4| = 1$; **(b)** $|R_{20}^4| = 1$; **(c)** $|R_{20}^6| = 1$. For $N = 2$: $\gamma_{20} = 0.9988$, $\beta_{20} = 1066.0$; $N = 4$: $\gamma_{20} = 1.0215$, $\beta_{20} = 1623.9$; $N = 6$: $\gamma_{20} = 1.0276$, $\beta_{20} = 2177.5$.

2.2. Identification of damping parameter L-tuple

The elementary symmetric polynomial, $\sigma_l^m = \sum_{1 \leq j_1 < \cdots < j_l \leq m} \prod_{i=1}^{l} \zeta_{j_i}$, is defined as the sum of all possible products formed from l unrepeated elements drawn from the first m elements of an L-tuple $[\zeta_l]$. The definition is extended by setting $\sigma_0^m = 1$ and $\sigma_{k>m}^m = 0$. The values of the parameters $\mu_l \in \mathbb{C}$ are chosen to recover the linear order conditions following application of the damping parameter ν.

We associate the L roots ζ_l, in order of increasing real component $\Re(\zeta_l)$, with the damping coefficients μ_l by cycling through the N damping coefficients a total of M times. Newton-Raphson iterations then converge rapidly to the linear order conditions Eq. 5. The effects of the damping procedure on the stability domain are shown in Figs 1 and 2.

We remark that the stage intervals τ_l given by Eq. 9 are complex in general. However, with $d_0^1 = 0$, $d_1^1 = 1/2$, the standard first-order super-timestepping scheme [7, 8] is recovered with $B_M^1 = C_M$. For $N > 1$, either one or two values of τ_l have negative real parts.

The presented prescription implements conjugate pairs separately thereby necessitating full complex arithmetic. Other than some penalty in the additional computational demand required, we find no practical disadvantage to preserving this model of treating each factor as distinct. We

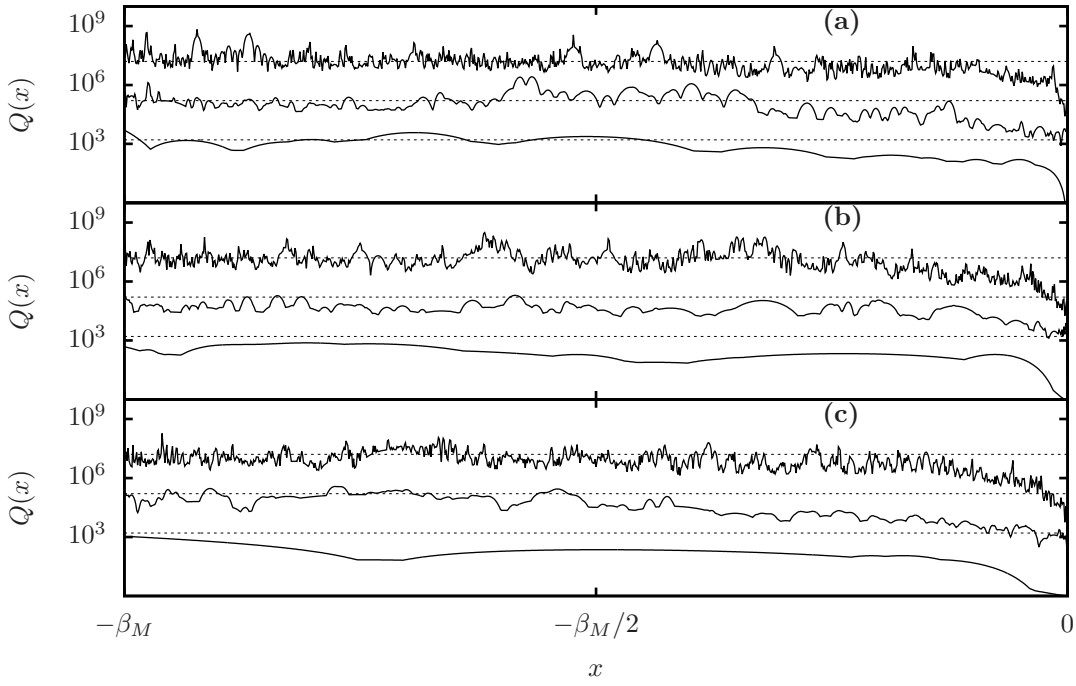

Figure 3: Maximum internal stability function $Q(x)$ for $L \approx 4000$ (upper lines), $L \approx 400$ (middle lines), $L \approx 40$ (lower lines): **(a)** $N = 2$ with $M = 2000$, 200, 20; **(b)** $N = 4$ with $M = 1000$, 100, 10; **(c)** $N = 6$ with $M = 667$, 67, 7. In all cases the default value of $\nu_0 = 0.05$ is used. Guidelines show values of L^2.

note that Lebedev [11, 28] proposed grouping roots in conjugate pairs and representing the resultant factor with a two-stage scheme.

2.3. Internal stability

Schemes comprising a high number of stages are internally unstable if the sequencing of the stages is allowed to admit uncontrolled growth of numerical errors [23, 29, 30, 31, 32]. Lebedev and Finogenov [33] first suggested sequencing of stages to manage uncontrolled growth of internal instabilities (see also [34]). Here, we present a straightforward algorithm for sequencing stages which limits the maximum amplification factor of internal instabilities to $\sim L^2$.

We define $v_{j,\,k} = |1 + a_j x_k|$, where $x_k \in [-\beta_M,\, 0]$ are discrete values spanning the spectrum of Eq. 12. The L-tuple $[\tau_l]$ is then ordered by holding the L_1 normed quantity

$$\left\| \max \left(\prod_{j=1}^{l} v_{j,\,k},\ \prod_{j=l+1}^{L} v_{j,\,k} \right) \right\|_1 \tag{10}$$

to a minimum value while l is increased from 1 to L. This procedure suppresses the growth of the internal stability functions $Q_{j,\,k}(x) = \prod_{l=j}^{k} |1 + a_l x|$, for $j,\, k = 1, \cdots, L$, over $x \in [-\beta_M,\, 0]$ and

provides excellent internal stability properties with high numbers of stages at low computational cost. In Fig. 3, we plot the maximum internal stability function $Q(x) = \max_{j,\,k}(Q_{j,\,k}(x))$ for the test cases $L \approx 4000$, 400, 40, with $N = 2$, 4, 6, and $\nu_0 = 0.05$. The optimization may be enhanced by concentrating the points x_k towards the bounds of the interval. (In this work a logistic function over a range of 15 is employed to generate the sample points.) We observe the maximum internal amplification factor scales approximately as L^2, independently of N. Hence, the internal stability properties are well within the acceptable limits of modern computing precision for any practical problem.

Consistent with these findings, we note that internal amplification factors of $\sim 10^6$ are quoted in the literature for RKC methods with 1000 stages [35], and furthermore, a quadratic dependence on stage number is suggested by Sommeijer et al. [15]. Conversely, ROCK2 methods are reported to demonstrate amplification factors of $\sim 10^9$ at 200 stages by Hundsdorfer and Verwer [35], suggesting internal instability growth rates 150 time larger than for RKC and FRKC2 schemes.

We note that the SERK scheme is also limited in stage number, albeit principally due to severely ill-conditioned matrix systems used to calculate the stability polynomials requiring 600 digits of precision for 320 stages by means of the Remez algorithm [18]. However, a subsequent revision of the SERK methodology has demonstrated a stability range which is four times larger [36].

3. Factorized Runge-Kutta-Chebyshev polynomial derivation

We consider the canonical one-dimensional diffusion equation

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial x^2}. \tag{11}$$

The semi-discrete form of Eq. 11 may be written $w' = h^{-2}Dw$, where D is a tridiagonal matrix with diagonal entries -2, subdiagonal entries 1, and describes a second-order central discretization of the spatial derivative. The eigenvalues of D, referred to hereafter as the elementary evolution operator, are negative with a maximum magnitude of 4. Application of the numerical scheme given by Eq. 3 yields

$$w^{n+1} = \prod_{l=1}^{L} \left(I + \frac{\tau_l}{h^2} D \right) w^n. \tag{12}$$

The FRKC polynomial B_M^N may be derived by consideration of the canonical scheme given by Eq. 12 over an extended timestep T, spanning time levels t^n to t^{n+1}, and consisting of M *segments*, with each segment comprising N stages. We write the solution state corresponding to w^n as W^0, and assume $W_0^0 = 1$ and $W_{k\neq 0}^0 = 0$, since more complex states may be constructed by superpositions. The solution state corresponding to w^{n+1} is then obtained from $W^M = \prod_{l=1}^{MN}(I + h^{-2}\tau_l^M D)W^0$. To aid the following discussion, Fig. 4 is provided to graphically represent solution states W^m at different segment levels for the particular case $N = 2$. A reference point value of the solution state W^M at spatial index j is shown as a black node.

To proceed, we assume schemes consisting of m segments, $m = 1, \cdots, M-1$, are known which generate the solution states, $W^m = \prod_{l=1}^{mN}(I + h^{-2}\tau_l^m D)W^0$. For $m = 1$, the solution state W^1 spans $2N + 1$ nodes from a given point profile W^0. Successive states regenerate this pattern, but spanning $2mN + 1$ nodes, with non-zero values interspersed by $(m/N - 1)$ zero valued nodes. We refer to the sequence of patterns over increasing values of m as a pattern *flow*. Illustrations of sample pattern flows are shown in Fig. 4.

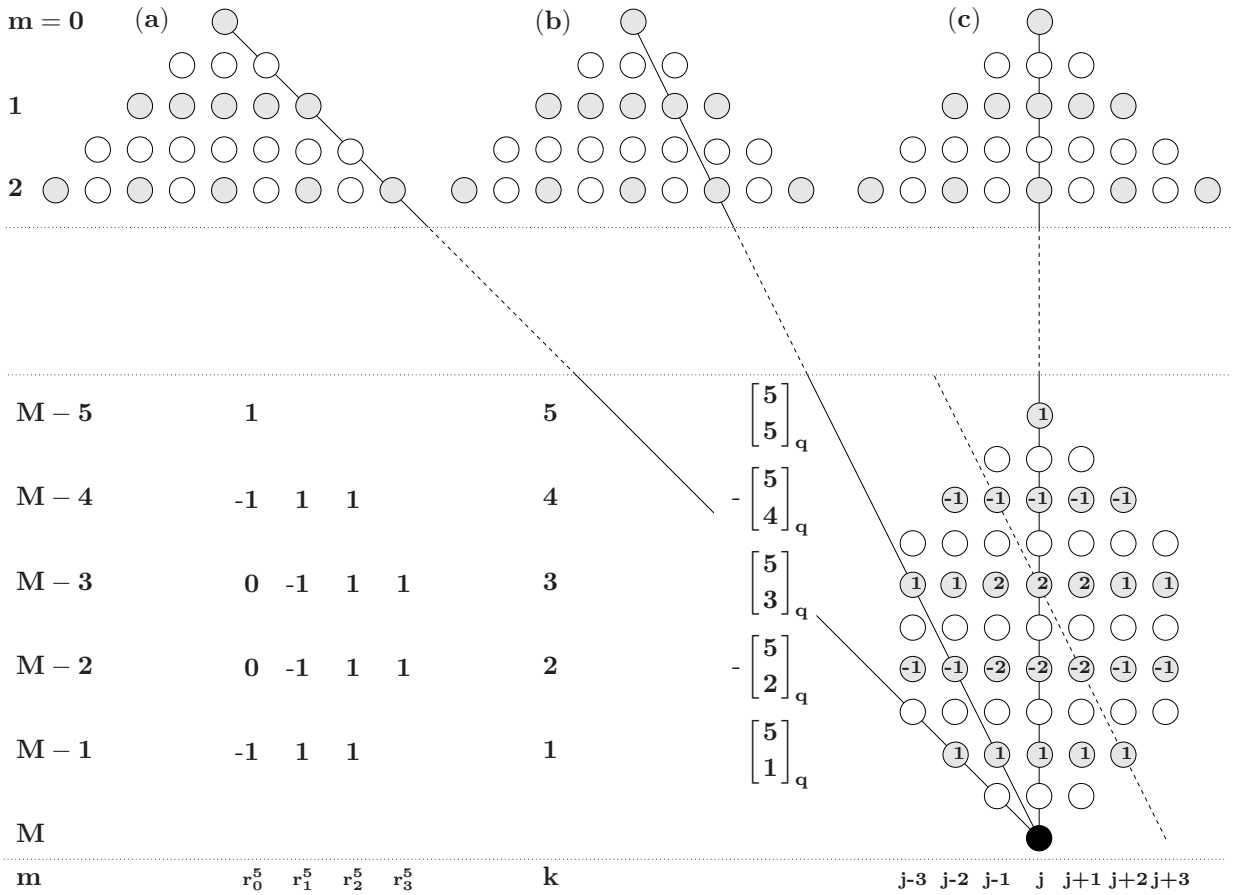

Figure 4: Graphical representation of construction of primitive recurrence relation between state value $\overline{W}_j^L$ and solution states for $m < L$ at intervals of $N = 2$; non-zero coefficients used in Eq. 18 corresponding to Gaussian polynomials $\begin{bmatrix} P \\ k \end{bmatrix}_q$ are shown ($P = 2N - 1$). Nodes at values of m which are non-integral multiples of N (empty circles) do not appear in the relation construction. Pattern flows emerging from three sample source distributions up to segment level $m = 2$ are shown (labeled a, b, and c). Rays terminating from the filled node at $m = L$ and originating at the apices of the sample distributions are shown summing to unity (solid lines). Rays which do not similarly project the solution pattern from $m = 2$ through to $m = L$ sum to zero (dashed line). Also shown are the coefficients $r_g^{P,\,k}$ prescribed by Eq. 20 at given values of k.

Using Eq. 8, the components of the states W^m may be recast as $\overline{W}_j^m = W_j^m \prod_{l=1}^{mN}(1-\zeta_l^m)$. Over a single timestep, Eq. 12 then takes the simplified form

$$\overline{W}^m = \prod_{l=1}^{mN} \overline{D}_l^m \overline{W}^0, \tag{13}$$

where $\overline{D}_l^m$ is a tridiagonal matrix with diagonal entries $-\zeta_l^m$ and subdiagonal entries $1/2$. In terms of the elementary symmetric polynomials we have

$$\overline{W}_j^m = \sum_{l=0}^{m} c_{j,\,l}^m \sigma_l^m, \tag{14}$$

where $c_{j,\,l}^m$ are coefficients dependent on the scheme Eq. 13. By induction, these coefficients have the properties

$$\begin{aligned} c_{0,\,m}^m \quad &= (-1)^m, & c_{j\neq 0,\,m}^m \quad &= 0, \\ c_{j,\,l\neq m}^m \quad &= \frac{1}{2}\left(c_{j-1,\,l}^{m-1} + c_{j+1,\,l}^{m-1}\right), & c_{j,\,l\neq 0}^m \quad &= -\,c_{j,\,l-1}^{m-1}. \end{aligned} \tag{15}$$

The $(m+1)$-tuples $[\sigma_l^m]$ fully determine $\overline{W}^m$ through the roots ζ_l^m of the associated polynomial B_m^N defined by

$$\frac{B_m^N}{(2)^{mN-1}} = \sum_{l=0}^{m} (-1)^l \sigma_l^m (z)^{m-l}, \tag{16}$$

where $(2)^{mN-1}$ is a normalization factor. Hence, the mN-tuple $[\tau_l^m]$ is completely specified by $[\sigma_l^m]$.

3.1. Derivation

The $(m+1)$-tuple $[0^{b_L},\, \sigma^{m-b},\, 0^{b_R}]$ is constructed from the elementary symmetric polynomials corresponding to the solution $\overline{W}^{m-b}$, where σ^{m-b} indicates the ordered elements $\sigma_0^{m-b}, \ldots, \sigma_{m-b}^{m-b}$, zero superscripts denote multiplicity, and $b = b_L + b_R$. Through Eq. 16, the $(m+1)$-tuple maps to the degree $m - b_L$ polynomial $(-1)^{b_L}(z)^{b_R}(2)^{-(m-b)N+1} B_{m-b}^N$. Inserting $[0^{b_L},\, \sigma^{m-b},\, 0^{b_R}]$ into Eq. 14 and appealing to the properties of the coefficients $c_{j,\,l}^m$, as given by Eq. 15, yields a direct correspondence to $(-1)^{b_L}\left(\frac{1}{2}\right)^{b_R} \sum_{g=0}^{b_R} \binom{b_R}{g} \overline{W}_{j-b_R+2g}^{m-b}$. Hence, we derive the association

$$\sum_{g=0}^{b_R} \binom{b_R}{g} \overline{W}_{j-b_R+2g}^{m-b} \sim (2z)^{b_R} \frac{B_{m-b}^N}{(2)^{(m-b)N-1}}. \tag{17}$$

By Eq. 13, the solution state $\overline{W}^{m+1}$ generates a pattern scaled by a factor of $1/2$ with respect to the pattern corresponding to the solution state $\overline{W}^m$. Hence, a recurrence relation generating the correct pattern comprising any weighted average of $(2)^{L-m}\overline{W}^m$ over available values of m will yield a valid solution state $\overline{W}^L$.

We define a *ray* as any connection on a uniformly spaced graph which passes through nodes on every segment level m, $m = 1, \cdots, M-1$. The sum of the recurrence weightings over any ray terminating at $m = L$ must be unity if the ray originates at the origin of a pattern flow at $m = 0$, and zero otherwise. The coefficients of the Gaussian polynomials $\begin{bmatrix} P \\ k \end{bmatrix}_q$, $k = 1, \cdots, P$, denoted $\begin{bmatrix} P \\ k \end{bmatrix}_q^l$,

possess the required properties. In Fig. 4, rays are shown summing to unity and zero, with a list of weightings satisfying these properties for all possible rays in the particular case of $N = 2$. Defining $P = 2N + 1$, the primitive form of the recurrence relation for $\overline{W}^L$

$$\overline{W}_j^L = \sum_{k=1}^{N} (-1)^{k+1} \sum_{l=0}^{G_k} \begin{bmatrix} P \\ k \end{bmatrix}_q^l \quad \Bigg[\quad \left(\frac{1}{2}\right)^{kN} \overline{W}_{j-\frac{1}{2}G_k+l}^{L-kN} - \left(\frac{1}{2}\right)^{(P-k)N} \overline{W}_{j-\frac{1}{2}G_k+l}^{L-(P-k)N} \Bigg] + \left(\frac{1}{2}\right)^{PN} \overline{W}_j^{L-PN}, \tag{18}$$

where $G_k = kP - k^2$ is the degree of $\begin{bmatrix} P \\ k \end{bmatrix}_q$ for $k \leq N$. We note that the Gaussian polynomial $\begin{bmatrix} P \\ k \end{bmatrix}_q$ possesses a unique representation as a summation of the binomial powers $(1 + q^2)^g$, for $g = 0, \cdots, G_k/2$, given by

$$\begin{bmatrix} P \\ k \end{bmatrix}_q = \sum_{g=0}^{\frac{1}{2}G_k} r_g^{P,\,k} q^{\frac{1}{2}G_k - g} (1 + q^2)^g, \tag{19}$$

where the coefficients $r_g^{P,\,k}$ follow the generating function

$$\sum_{k=0}^{\infty} \sum_{g=0}^{\infty} (-1)^k (2)^g r_g^{P,\,k} (t)^k (z)^g = (1 - t) \prod_{k=1}^{N} (1 + (t)^2 - 2tC_k). \tag{20}$$

Then, using Eq. 19, we recast Eq. 18 in the form

$$\overline{W}_j^L = \sum_{k=1}^{N} (-1)^{k+1} \sum_{g=0}^{\frac{1}{2}G_k} r_g^{P,\,k} \sum_{l=0}^{g} \binom{g}{l} \quad \Bigg[\quad \left(\frac{1}{2}\right)^{kN} \overline{W}_{j-g+2l}^{L-kN} - \left(\frac{1}{2}\right)^{(P-k)N} \overline{W}_{j-g+2l}^{L-(P-k)N} \Bigg] + \left(\frac{1}{2}\right)^{PN} \overline{W}_j^{L-PN}. \tag{21}$$

Applying the association given in Eq. 17 for the terms in Eq. 21, the recurrence relation for B_M^N follows as

$$B_M^N = \sum_{k=1}^{N} (-1)^{k+1} \left[B_{M-k}^N - B_{M-P+k}^N \right] \sum_{g=0}^{\frac{1}{2}G_k} r_g^{P,\,k} (2z)^g + B_{M-P}^N. \tag{22}$$

We continue by noting that the generating function for B_k^N derived from the recurrence relation given by Eq. 22 is

$$\sum_{k=0}^{\infty} (t)^k B_k^N = \frac{\sum_{k=0}^{2N} (t)^k b_k^N}{1 - \sum_{k=1}^{N} (-1)^{k+1} \left[(t)^k - (t)^{(P-k)} \right] \sum_{g=0}^{\frac{1}{2}G_k} r_g^{P,\,k} (2z)^g - (t)^P}, \tag{23}$$

where b_k^N are coefficients determined by the seed states of B_m^N. Appealing to Eq. 20, the generating function derived from the recurrence relation given by Eq. 22 is

$$\sum_{k=0}^{\infty}(t)^k B_k^N = \frac{b_0^N}{1-t} + 2\sum_{k=1}^{N}\frac{b_k^N(1-zt)}{1+(t)^2-2tC_k}(1-t)\prod_{k=1}^{N}(1+(t)^2-2tC_k), \tag{24}$$

where b_k^N are coefficients determined by the seed states of B_m^N and the normalization $B_k^N(1) = b_0^N(1)+\sum_{k=1}^{N}2b_k^N(1)$ has been imposed in order to fix the forms of the numerators in the separated fractions.

Finally, noting that the generating function for C_{km} is $\sum_{m=0}^{\infty}(t)^m C_{km} = (1-zt)/(1+(t)^2-2C_k)$, we conclude that $B_k^N = b_0^N + 2\sum_{k=1}^{N} b_k^N C_{km}$. Consideration of the particular case $N = 1$, $M = 1$ indicates a correspondence between b_k^1 and d_k^1 is required in order to match the required solution pattern and normalization properties. A general correspondence between b_k^N and d_k^N is established by considering successive values of N with $M = 1$ for the limiting cases of $d_k^N = 0$, $0 < k < N$, and bootstrapping the solution from $N = 1$ by setting M to N. This completes the derivation of the analytic expression for the FRKC stability polynomial given by Eq. 4.

4. Tests

In this section numerical studies of a two-dimensional two-species Brusselator diffusion-reaction problem are presented which confirm that high-order FRKC stability polynomials meet all relevant linear order conditions and that the derived factorized numerical schemes are both stable and efficient. Split schemes, denoted FRKCs, are obtained by means of complex splitting techniques; the linear diffusion operator are treated via FRKC methods while the nonlinear reaction terms are integrated using standard Runge-Kutta techniques. The performance of the second-order accurate unsplit FRKC2 scheme is compared to second-order RKC and CVODE2 codes . Finally, comparisons of higher order split FRKC schemes (at orders 4 and 6) with fourth-order ROCK4 and fifth-order CVODE are presented.

4.1. High-order splitting

FRKC stability polynomials satisfy linear order conditions to an arbitrary order of accuracy. This property may be exploited in solving numerical problems for semi-linear stiff systems of equations through operator splitting methods [37, 38, 39, 40]. We note that in the literature, the linear and nonlinear terms of reaction-diffusion models have been decoupled under a variety of numerical integration techniques including: splitting methods [41, and previous references], Implicit-Explicit Runge-Kutta-Chebyshev (IMEX RKC) methods [42, 43], PIROCK [44], and Local Linearization Runge-Kutta (LLRK) methods [45, 46]. Integration of linear terms is suited to extended stability explicit schemes, while stiff nonlinear reaction term may be handled by standard numerical techniques [47, 48].

We note, however, that high-order splitting has been shown to give rise to an order reduction effect in some reaction-diffusion cases [49]. For Dirichlet and Neumann boundary conditions, splitting techniques may give rise to order reduction at boundaries [35, 38]. It has also been observed that the full order is recovered on the interior of the computational domain when it is taken sufficiently far from the influence of the boundaries [50, 51]. Boundary conditions for the separate operator updates are necessary to avoid order reduction effectively, however, as yet, no consistent treatment exists [52].

Assuming Eq. 2 is linearized and split in the form $w' = (A + B)w$, the solution over a timestep T requires an approximation to the operator $e^{T(A+B)}$. High-order approximations may be obtained through appropriate choice of partial steps T_j where

$$w^{n+1} = e^{T_{k_J} B} e^{T_{k_{J-1}} A} \cdots e^{T_{k_3} B} e^{T_{k_2} A} e^{T_{k_1} B} w^n. \tag{25}$$

Formally, with support from numerical studies [37, 53], the splitting scheme given by Eq. 25 may be may be extended to the semi-linear parabolic form of Eq. 2 given by

$$w' = Aw + f_B(w) \tag{26}$$

by replacing the exponential operator $e^{T_{k_j} B}$ with a step of the nonlinear equation $w' = f_B(w)$ over the interval T_{k_j}. For reference, the complex splitting schemes used in this work are provided in Appendix B.

4.2. Brusselator

The Brusselator [54, 21] is a stiff nonlinear diffusion-reaction problem describing chemical kinetics of a tri-molecular chemical reaction. The test case considered here is a two-dimensional hybrid of the one- and two-dimensional Brusselator problems presented by Hairer et al. [21], and Hairer and Wanner [31], with governing equations given by

$$\begin{aligned} \partial v/\partial t &= \epsilon \left(\partial^2 v/\partial {x_1}^2 + \partial^2 v/\partial {x_2}^2 \right) + A - (B+1)v + wv^2, \\ \partial w/\partial t &= \epsilon \left(\partial^2 w/\partial {x_1}^2 + \partial^2 w/\partial {x_2}^2 \right) + Bv - v^2 w, \end{aligned} \tag{27}$$

and initial conditions $v(0,\, x) = A + \sin(2\pi x)$, $v(0,\, x) = B/A + \cos(2\pi y)$. The initial state is therefore a simple perturbation of the equilibrium solution. The problem is configured with parameters $\epsilon = 0.02$, $A = 1$, and $B = 3$, and the solution is obtained at $t = 2$, or $t = 8$, on the domain $0 \le x_1 \le 1$, $0 \le x_2 \le 1$, under periodic boundary conditions.

4.3. Linear order conditions

The semi-discrete form of Eq. 27 may be written $w' = Aw + f_B(w)$, where A describes the discretization of the Laplacian with respect to x_1 and x_2, and $f_B(w)$ contains the reaction terms. Linear diffusion terms are integrated using FRKC methods and nonlinear reaction terms via standard techniques. The linear order properties of the FRKC stability polynomials are confirmed by considering the convergence rates of the approximated solution to the exact solution at $t = 2$ as a function of step size.

For all presented results, we use $M = 20$ and the approximation $\gamma_M = 1$. The number of grid points is 400 in each of the two spatial variables. For these parameters, the FRKC stability polynomials achieve approximately 81% ($\beta_R = 1066.667$), 74% ($\beta_R = 1600$) and 73% ($\beta_R = 2133.333$) of the optimal intervals for $N = 2$, 4, 6 respectively. For the corresponding standard explicit Runge-Kutta schemes these values represent a speedup in efficiency by factors of about 27 for $N = 2$, and 30 for both $N = 4$ and $N = 6$. All polynomials are damped with damping parameter $\nu_0 = 0.05$, reducing the stability domains' real extent by factors of $1 - \nu_0/N$. Finally, in order to meet the specified solution time, timesteps are scaled by 0.9846, 0.9001, 0.7563 for $N = 2$, 4, 6 respectively. Quadruple precision is used in all calculations. Results are presented in Table 1 where the L_1 and L_∞ errors are shown over a range of resolutions at each considered value

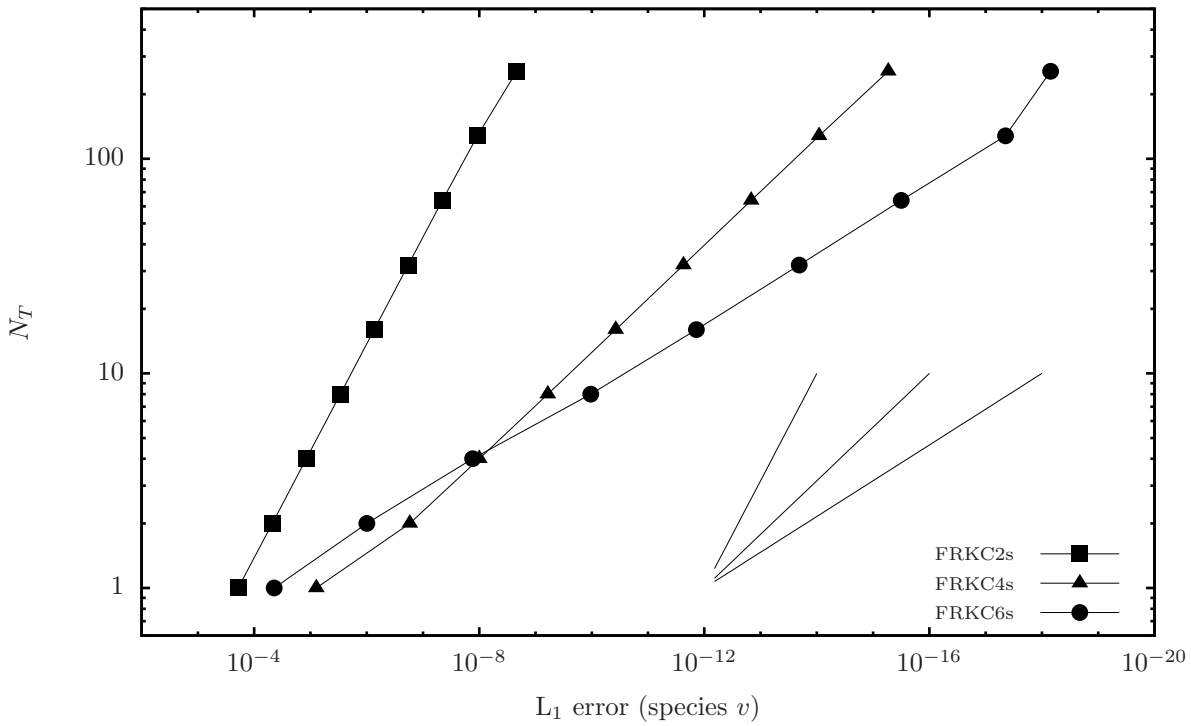

Figure 5: L_1 errors plotted against number of timesteps, N_T, for species v of the two-dimensional Brusselator problem. Results correspond to split problem with linear diffusion treated via FRKC methods at orders 2, 4, and 6, and nonlinear reaction terms integrated via standard techniques. Guide lines are shown for $(L_1\text{error})^{-1/2}$, $(L_1\text{error})^{-1/4}$, $(L_1\text{error})^{-1/6}$. Table 1 gives the values for all points shown.

Table 1: Error convergence results for the two-dimensional Brusselator test problem solved via split FRKCs schemes with $N = 2, 4, 6$. Each row corresponds to a specific test with columns listing: N, order of accuracy; N_T, the number of timesteps; L_1 norm of the error between the approximate and exact solutions; L_1 order of convergence with reference to previous row; L_∞ error; L_∞ order. Errors refer to the solution for species v. L_1 errors are also shown in Fig.5.

N	N_T	L_1 error	L_1 order	L_∞ error	L_∞ order
2	50	1.92×10^{-4}	-	4.25×10^{-4}	-
	100	4.76×10^{-5}	2.02	1.03×10^{-4}	2.04
	200	1.18×10^{-5}	2.01	2.55×10^{-5}	2.02
	400	2.95×10^{-6}	2.00	6.32×10^{-6}	2.01
	800	7.36×10^{-7}	2.00	1.57×10^{-6}	2.01
	1600	1.83×10^{-7}	2.01	3.92×10^{-7}	2.01
	3200	4.53×10^{-8}	2.02	9.68×10^{-8}	2.02
	6400	1.08×10^{-8}	2.07	2.30×10^{-8}	2.07
	12 800	2.16×10^{-9}	2.32	4.61×10^{-9}	2.32
4	50	7.85×10^{-6}	-	1.21×10^{-5}	-
	100	1.71×10^{-7}	5.52	2.76×10^{-7}	5.46
	200	9.96×10^{-9}	4.10	1.65×10^{-8}	4.06
	400	6.05×10^{-10}	4.04	1.02×10^{-9}	4.02
	800	3.76×10^{-11}	4.01	6.36×10^{-11}	4.00
	1600	2.35×10^{-12}	4.00	3.98×10^{-12}	4.00
	3200	1.47×10^{-13}	4.00	2.49×10^{-13}	4.00
	6400	9.13×10^{-15}	4.01	1.55×10^{-14}	4.00
	12 800	5.37×10^{-16}	4.09	9.15×10^{-16}	4.09
6	50	4.41×10^{-5}	-	7.37×10^{-5}	-
	100	9.92×10^{-7}	5.48	2.35×10^{-6}	4.97
	200	1.31×10^{-8}	6.24	2.88×10^{-8}	6.35
	400	1.04×10^{-10}	6.98	1.78×10^{-10}	7.34
	800	1.39×10^{-12}	6.23	2.34×10^{-12}	6.25
	1600	2.06×10^{-14}	6.07	3.52×10^{-14}	6.05
	3200	3.16×10^{-16}	6.02	5.49×10^{-16}	6.01
	6400	4.44×10^{-18}	6.16	9.87×10^{-18}	5.80
	12 800	7.01×10^{-19}	2.66	2.01×10^{-18}	2.30

Table 2: Errors from FRKC2, RKC, CVODE2 from tests of the two-dimensional Brusselator problem. The number of timesteps, N_T, and the number of stages per timestep, L, (or the error tolerance, Tol, in the case of CVODE) are given in the first two columns respectively. The wall time taken for each run, T_{WALL}, is presented in the third column. L_1 and L_∞ errors for both species are presented in the remaining columns. The L_1 error for species v is plotted in Fig. 6.

			Species v		Species w	
N_T	L/Tol	T_{WALL} (s)	L_1 error	L_∞ error	L_1 error	L_∞ error
			FRKC2			
50	80	25	5.16×10^{-3}	5.52×10^{-3}	1.30×10^{-3}	1.40×10^{-3}
137	48	42	1.41×10^{-3}	1.45×10^{-3}	6.06×10^{-4}	6.25×10^{-4}
308	32	63	3.24×10^{-4}	3.34×10^{-4}	1.64×10^{-4}	1.68×10^{-4}
548	24	84	1.08×10^{-4}	1.11×10^{-4}	5.66×10^{-5}	5.81×10^{-5}
1317	16	137	1.85×10^{-5}	1.90×10^{-5}	1.00×10^{-5}	1.03×10^{-5}
2341	12	186	6.03×10^{-6}	6.20×10^{-6}	3.30×10^{-6}	3.38×10^{-6}
5266	8	288	1.26×10^{-6}	1.30×10^{-6}	6.94×10^{-7}	7.11×10^{-7}
9361	6	399	4.23×10^{-7}	4.35×10^{-7}	2.32×10^{-7}	2.38×10^{-7}
21 062	4	637	9.48×10^{-8}	9.76×10^{-8}	5.16×10^{-8}	5.30×10^{-8}
105 026	2	1881	5.71×10^{-9}	5.88×10^{-9}	3.11×10^{-9}	3.19×10^{-9}
			RKC			
39	90	32	4.97×10^{-3}	5.81×10^{-3}	8.05×10^{-3}	8.54×10^{-3}
78	64	45	6.59×10^{-4}	7.64×10^{-4}	6.50×10^{-4}	7.02×10^{-4}
137	48	60	2.49×10^{-4}	2.83×10^{-4}	1.08×10^{-4}	1.26×10^{-4}
309	32	90	5.07×10^{-5}	5.73×10^{-5}	1.36×10^{-5}	1.75×10^{-5}
549	24	119	1.59×10^{-5}	1.81×10^{-5}	3.84×10^{-6}	5.12×10^{-6}
1237	16	177	3.13×10^{-6}	3.55×10^{-6}	7.27×10^{-7}	9.85×10^{-7}
2206	12	237	9.92×10^{-7}	1.13×10^{-6}	2.28×10^{-7}	3.10×10^{-7}
5007	8	359	2.00×10^{-7}	2.27×10^{-7}	4.50×10^{-8}	6.17×10^{-8}
9012	6	486	6.57×10^{-8}	7.46×10^{-8}	1.40×10^{-8}	1.94×10^{-8}
21 028	4	750	1.52×10^{-8}	1.71×10^{-8}	1.88×10^{-9}	3.03×10^{-9}
39 428	3	1056	6.56×10^{-9}	7.20×10^{-9}	4.13×10^{-10}	7.98×10^{-10}
			CVODE2			
1226	5×10^{-6}	107	2.37×10^{-3}	2.42×10^{-3}	9.65×10^{-4}	1.01×10^{-3}
1499	10^{-6}	128	7.79×10^{-4}	8.26×10^{-4}	5.85×10^{-5}	8.45×10^{-5}
2534	10^{-7}	194	4.69×10^{-5}	5.83×10^{-5}	2.99×10^{-5}	3.91×10^{-5}
4991	10^{-8}	312	3.06×10^{-5}	3.22×10^{-5}	3.13×10^{-5}	3.27×10^{-5}
10 029	10^{-9}	523	5.47×10^{-6}	5.60×10^{-6}	4.35×10^{-6}	4.45×10^{-6}
22 763	10^{-10}	987	7.05×10^{-7}	7.14×10^{-7}	5.06×10^{-7}	5.08×10^{-7}
48 444	10^{-11}	1706	1.55×10^{-7}	1.58×10^{-7}	1.07×10^{-7}	1.09×10^{-7}
109 474	10^{-12}	3405	2.96×10^{-8}	3.01×10^{-8}	2.06×10^{-8}	2.10×10^{-8}
232 430	10^{-13}	6756	6.05×10^{-9}	6.16×10^{-9}	4.16×10^{-9}	4.24×10^{-9}

of N. Fig. 5 illustrates the dependence of the L_1 errors on the number of timesteps, N_T, for species v. With the exception of the final point for the sixth-order integration, where machine precision is exceeded, all solutions are converging in good agreement with the nominal orders of accuracy (ie. $\mathrm{L}_1^{-1/N}$). Fitting the L_1 errors yields observed orders 2.04 ± 0.01, 4.08 ± 0.04, 6.1 ± 0.2 for $N = 2$, 4, 6 respectively, while the L_∞ errors give 2.05 ± 0.01, 4.09 ± 0.05, and 6.0 ± 0.2. We conclude that FRKC methods demonstrate internal stability and comply with linear order conditions to the specified order of accuracy.

4.4. Second-order comparative studies

Since all order conditions are linear at second-order, FRKC2 schemes will naturally maintain second-order accuracy for nonlinear problems without the necessity of splitting or composition methods. Here we present comparative studies between FRKC2 and a number of alternative numerical integration methods. In particular, we provide comparisons with the RKC method [15] which, similarly to FRKC2, depends on the properties of Chebyshev polynomials. We also compare results

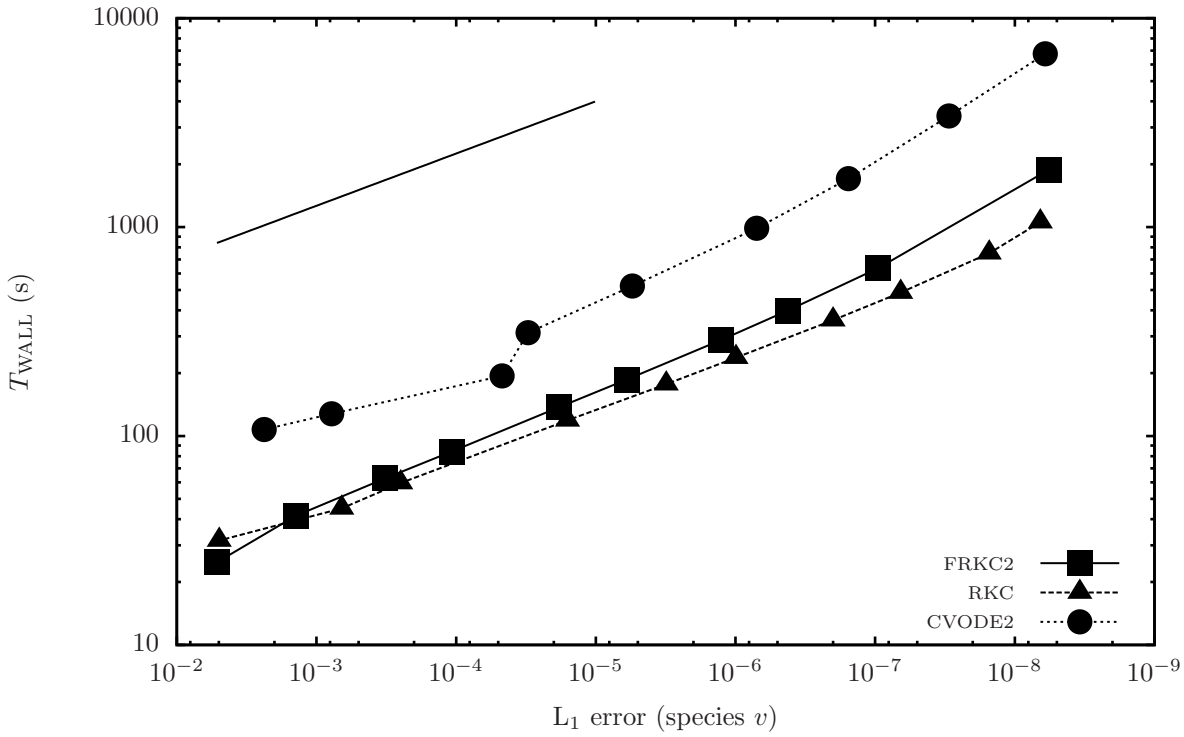

Figure 6: Performance results for FRKC2, RKC, CVODE2 for stiff nonlinear Brusselator problem. L_1 error for species v is plotted for all cases. A guide line is shown for $(L_1\text{error})^{-1/4}$. Values are tabulated in Table 2.

with a GMRES Krylov-preconditioned BDF integrator from the CVODE numerical integration package [55]. The CVODE solver maintains a specified tolerance by means of adaptive stepping up to a maximum fifth-order accuracy. However, the order is restricted to 2 for the CVODE2 solver used in these comparisons.

We proceed by considering the two-dimensional Brusselator problem described in Section 4.2 with 1000 grid points along each spatial axis, and the solution taken at time $t = 8$. The stepsize is fixed for individual tests of the explicit schemes and the number of internal stages is optimized for the selected stepsize. As such, each of the numerical solutions generated for these tests is derived from a single distinct stability polynomial. In general, however, error control procedures may be implemented [15] which will result in stability polynomials of varied degree contributing to particular solutions. The optimal efficiency for extended stability explicit solvers follows $T_{\mathrm{WALL}} \propto \mathrm{error}^{-1/2N}$ (where T_{WALL} is the wall-time required for computation of a particular problem).

Results are provided in Table 2 for FRKC2, RKC, and CVODE2. The L_1 errors for species v are plotted in Fig. 6 against the time required for the simulations to be carried out on a standard desktop machine at double precision. While the FRKC2 solver requires complex arithmetic, this is compensated by smaller errors than for the RKC solver at equivalent numbers of timesteps. Overall, FRKC2 runs at about 70% of the efficiency of RKC. As previously noted, following a similar strategy to Lebedev [11, 28] will improve performance.

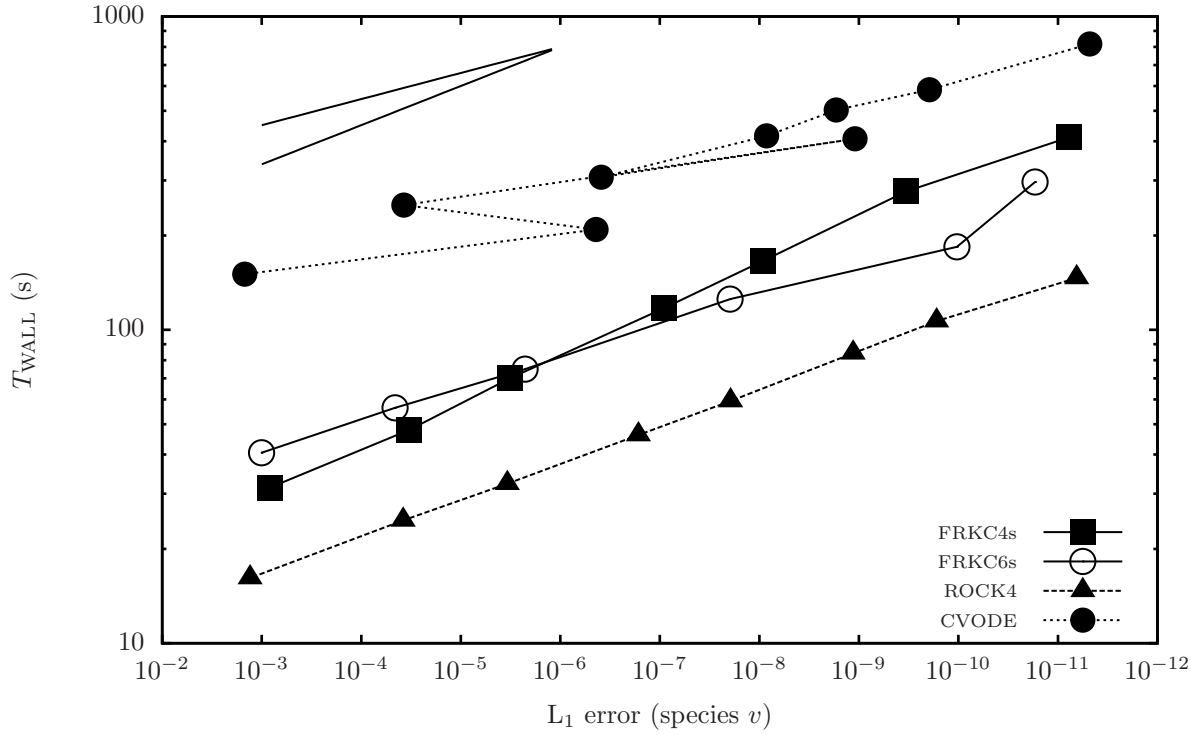

Figure 7: Performance results for FRKC4s, FRKC6s, ROCK4, CVODE for stiff nonlinear Brusselator problem. L_1 error for species v is plotted for all cases. Guide lines are shown for $(\mathrm{L}_1\mathrm{error})^{-1/8}$ and $(\mathrm{L}_1\mathrm{error})^{-1/12}$. Values are tabulated in Table 3.

Table 3: Errors from FRKC4s, FRKC6s, ROCK4, CVODE from tests of the two-dimensional Brusselator problem. The number of timesteps, N_T, and the number of stages per timestep, L, (or the error tolerance, Tol, in the case of CVODE) are given in the first two columns respectively. The wall time taken for each run, T_{WALL}, is presented in the third column. L_1 and L_∞ errors for both species are presented in the remaining columns. The L_1 error for species v is plotted in Fig. 7.

			Species v		Species w	
N_T	L	T_{WALL} (s)	L_1 error	L_∞ error	L_1 error	L_∞ error
			FRKC4s			
23	96	31	8.27×10^{-4}	8.49×10^{-4}	5.49×10^{-4}	5.62×10^{-4}
51	64	48	3.30×10^{-5}	3.30×10^{-5}	2.35×10^{-5}	2.37×10^{-5}
91	48	70	3.21×10^{-6}	3.25×10^{-6}	2.31×10^{-6}	2.32×10^{-6}
222	32	117	9.17×10^{-8}	9.27×10^{-8}	6.66×10^{-8}	6.68×10^{-8}
395	24	166	9.17×10^{-9}	9.27×10^{-9}	6.68×10^{-9}	6.71×10^{-9}
887	16	277	3.34×10^{-10}	3.38×10^{-10}	2.42×10^{-10}	2.43×10^{-10}
1577	12	411	7.83×10^{-12}	7.91×10^{-12}	4.03×10^{-12}	4.09×10^{-12}
			FRKC6s			
5	144	41	1.00×10^{-3}	1.02×10^{-3}	8.60×10^{-4}	8.74×10^{-4}
10	96	56	4.58×10^{-5}	4.62×10^{-5}	3.57×10^{-5}	3.61×10^{-5}
17	72	75	2.26×10^{-6}	2.31×10^{-6}	1.79×10^{-6}	1.84×10^{-6}
42	48	125	1.97×10^{-8}	2.01×10^{-8}	1.55×10^{-8}	1.57×10^{-8}
75	36	184	1.04×10^{-10}	1.04×10^{-10}	8.27×10^{-11}	8.33×10^{-11}
168	24	296	1.70×10^{-11}	1.72×10^{-11}	1.34×10^{-11}	1.35×10^{-11}
			ROCK4			
56	102	16	1.31×10^{-3}	7.26×10^{-3}	1.34×10^{-3}	7.19×10^{-3}
130	67	25	3.79×10^{-5}	2.13×10^{-4}	2.88×10^{-5}	1.49×10^{-4}
224	51	32	3.40×10^{-6}	1.83×10^{-5}	2.76×10^{-6}	1.42×10^{-5}
451	36	46	1.64×10^{-7}	8.18×10^{-7}	1.38×10^{-7}	7.56×10^{-7}
749	28	59	1.95×10^{-8}	8.90×10^{-8}	1.60×10^{-8}	8.14×10^{-8}
1483	20	84	1.14×10^{-9}	4.64×10^{-9}	8.99×10^{-10}	4.74×10^{-9}
2345	16	107	1.65×10^{-10}	5.58×10^{-10}	1.25×10^{-10}	5.66×10^{-10}
4282	12	146	6.48×10^{-12}	3.32×10^{-11}	6.13×10^{-12}	2.80×10^{-11}
			CVODE			
2001	10^{-6}	151	1.48×10^{-3}	1.51×10^{-3}	1.13×10^{-3}	1.15×10^{-3}
3203	10^{-7}	209	4.38×10^{-7}	1.26×10^{-6}	4.83×10^{-7}	1.71×10^{-6}
3309	10^{-8}	250	3.74×10^{-5}	3.90×10^{-5}	2.91×10^{-5}	3.03×10^{-5}
6628	10^{-9}	407	1.09×10^{-9}	5.14×10^{-9}	9.33×10^{-10}	4.06×10^{-9}
4818	10^{-10}	307	3.88×10^{-7}	3.89×10^{-7}	3.27×10^{-7}	3.28×10^{-7}
6246	10^{-11}	416	8.46×10^{-9}	8.97×10^{-9}	4.88×10^{-9}	4.95×10^{-9}
8376	10^{-12}	503	1.69×10^{-9}	1.76×10^{-9}	1.32×10^{-9}	1.37×10^{-9}
10 515	10^{-13}	584	1.96×10^{-10}	1.97×10^{-10}	1.38×10^{-10}	1.41×10^{-10}
16 036	10^{-14}	817	4.79×10^{-12}	5.29×10^{-12}	3.38×10^{-12}	3.84×10^{-12}

4.5. High-order comparative studies

An advantage of FRKC methods over other extended stability methods is extensibility to arbitrarily high-order linear stability polynomials. In order to apply high (above second)-order FRKC stability polynomials to nonlinear problems complex splitting techniques may be employed, as demonstrated in Sec. 4.3. In the following tests, we consider fourth and sixth-order solutions of the two-dimensional Brusselator problem presented in Sec. 4.4. We note that finishing stages based on the theory of the composition of B-series may also be may be used to meet nonlinear order conditions [21, 12, 17]. While composition methods may, in principle, offer improved efficiency over splitting techniques, in the case of ROCK4, the application of finishing stages has been observed to result in order reduction problems, as well as erratic convergence properties, limiting the number of internal stages to a relatively small number of internal stages[24, 35]. The limit adopted within the ROCK4 code is at $L = 152$, whereas split FRKCs schemes do not suffer from the destabilizing influence of finishing stages. Furthermore, the number of nonlinear order conditions, and hence the complexity of the composition strategy, grows rapidly with increasing order [22]: there are four nonlinear order conditions at fourth-order, 31 order conditions at sixth-order, and 192 at eighth-order.

We present comparisons of the split schemes FRKC4s and FRKC6s with the fourth-order ROCK4 scheme. Reference solutions obtained using the CVODE solver are also presented with integration carried out to a maximum fifth-order accuracy. The test conditions are otherwise as described in Sec. 4.4. All data are presented in Table 3, and L_1 errors for species v are plotted in Fig. 7 against the time required for the simulations at double precision. FRKC4s is shown to run at approximately half the efficiency of ROCK4. This is primarily due to the additional computational overhead of carrying out calculations with complex values quantities which may be countered by rolling conjugate pair calculations together [11, 28]. In terms of efficiency, for the presented problem, the FRKC6s and FRKC4s methods lie approximately midway between ROCK4 and CVODE. Except at the very lowest acceleration parameters considered, the FRKCs trials show the predicted behaviour (ie. $T_{\mathrm{WALL}} \propto (\mathrm{L}_1\mathrm{error})^{-1/2N}$).

5. Conclusions

The fully prescribed analytic form of a new class of extended stability polynomials which satisfy all required linear order conditions to arbitrarily high-order has been presented. Factorized Runge-Kutta-Chebyshev (FRKC) stability polynomials are derived from first principles by inductive considerations of the implied recurrence relations. At order N, the FRKC polynomial of rank N, and degree $L = MN$, is shown to have the form of a summation of Chebyshev polynomials, with degrees at intervals of M, up to degree L. The $N+1$ weightings of the contributing Chebyshev polynomials are chosen to comply with the N linear order conditions, coupled with a conservation constraint. A damping procedure for broadening the stability domain of the FRKC stability polynomials to a finite width along the real axis is described which preserves the order of accuracy. The resultant stability polynomials have been demonstrated to have 81%, 74% and 73% of the optimal intervals for orders 2, 4, 6 respectively. FRKC numerical integration schemes are represented as a sequence of L sequenced forward Euler steps (stages) involving complex-valued timesteps constructed from the roots of FRKC stability polynomials of degree L. Internal stability is maintained by means of a sequencing algorithm, which limits the maximum internal amplification factor to $\sim L^2$: reserving 8 digits for accuracy, a hypothetical scheme of 10,000 stages is therefore viable in a numerical integration carried out at 16 digit precision.

Split FRKCs schemes have been applied at orders 2, 4, and 6, to the linear diffusion operator in numerical experiments on a stiff two-dimensional Brusselator reaction-diffusion system leading to the verification of expected convergence rates and hence compliance with the necessary linear order conditions.

We have presented comparative studies of the performance of FRKC2 , RKC , an established explicit extended stability code, and CVODE2 , an implicit preconditioned BDF solver from the CVODE suite limited to second-order accuracy. FRKC2 has been shown to be substantially more efficient than the CVODE2 solver, while performing at about 70% of the efficiency of RKC.

At higher orders, nonlinear order conditions require special attention. We have considered treatment of these nonlinear conditions through complex splitting techniques in efficiency tests of higher order (4 and 6) split FRKCs schemes in comparison with results from the the fourth-order ROCK4 code, which uses composition methods, and the implicit fifth-order CVODE solver. The tested FRKCs methods are found to have intermediate efficiency to ROCK4 and CVODE . We propose implementing conjugate pairing and Butcher group composition methods in future high-order implementations of FRKC methods.

6. Acknowledgments

Calculations involving the coefficients and roots of polynomials are handled with the `gmp` and `mpfr` libraries. Polynomial roots are obtained by means of the `MPSolve` package. MPSolve was written by Dario Andrea Bini and Giuseppe Fiorentino, Dipartimento di Matematica, Universita' di Pisa, Italy.

The author thanks an anonymous referee for constructive comments which have contributed to significant improvements in this paper.

References

[1] A. A. Markov, On a question by DI Mendeleev, Zapiski Imperatorskoi Akademii Nauk 62 (1890) 1–24.

[2] V. Markov, On functions deviating least from zero in a given interval, Izdat. Imp. Akad. Nauk, St. Petersburg (1892) 218–258.

[3] V. Saulev, Integration of parabolic equations by the grid method, Fizmatgiz, Moscow (1960).

[4] A. Guillou, B. Lago, Domaine de stabilité associé aux formules dintégration numérique déquations différentielles, a pas séparés et a pas liés. recherche de formules a grand rayon de stabilité, Ier Congr. Ass. Fran. Calcul., AFCAL (1960) 43–56.

[5] W. Gentzsch, A. Schluter, On one-step methods with cyclic stepsize changes for solving parabolic differential equations, Z. Angew. Math. Mech 58 (1978) T415–T416.

[6] P. van der Houwen, The development of Runge-Kutta methods for partial differential equations, Applied Numerical Mathematics 20 (1996) 261 – 272.

[7] V. Alexiades, G. Amiez, P. Gremaud, Super-time-stepping acceleration of explicit schemes for parabolic problems, Communications in numerical methods in engineering 12 (1996) 31–42.

[8] S. O'Sullivan, T. P. Downes, An explicit scheme for multifluid magnetohydrodynamics, Monthly Notices of the Royal Astronomical Society 366 (2006) 1329–1336.

[9] S. O'Sullivan, T. P. Downes, A three-dimensional numerical method for modelling weakly ionized plasmas, Monthly Notices of the Royal Astronomical Society 376 (2007) 1648–1658.

[10] S. O'Sullivan, C. O'Sullivan, On the acceleration of explicit finite difference methods for option pricing, Quantitative Finance 11 (2011) 1177–1191.

[11] V. I. Lebedev, Explicit difference schemes for solving stiff problems with a complex or separable spectrum, Computational mathematics and mathematical physics 40 (2000) 1729–1740.

[12] A. A. Medovikov, High order explicit methods for parabolic equations, BIT Numerical Mathematics 38 (1998) 372–390.

[13] P. J. van Der Houwen, B. P. Sommeijer, On the internal stability of explicit, m-stage Runge-Kutta methods for large m-values, ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift fr Angewandte Mathematik und Mechanik 60 (1980) 479–485.

[14] J. G. Verwer, Explicit Runge-Kutta methods for parabolic partial differential equations, Applied Numerical Mathematics 22 (1996) 359–379.

[15] B. Sommeijer, L. Shampine, J. Verwer, RKC: an explicit solver for parabolic PDEs, Journal of Computational and Applied Mathematics 88 (1998) 315–326.

[16] A. Abdulle, A. A. Medovikov, Second order Chebyshev methods based on orthogonal polynomials, Numerische Mathematik 90 (2001) 1–18.

[17] A. Abdulle, Fourth order Chebyshev methods with recurrence relation, SIAM Journal on Scientific Computing 23 (2002) 2041–2054.

[18] J. Martin-Vaquero, B. Janssen, Second-order stabilized explicit Runge-Kutta methods for stiff problems, Computer Physics Communications 180 (2009) 1802–1810.

[19] C. D. Meyer, D. S. Balsara, T. D. Aslam, A second-order accurate Super TimeStepping formulation for anisotropic thermal conduction, Monthly Notices of the Royal Astronomical Society 422 (2012) 2102–2115.

[20] C. D. Meyer, D. S. Balsara, T. D. Aslam, A stabilized Runge-Kutta-Legendre method for explicit super-time-stepping of parabolic and mixed equations, Journal of Computational Physics 257 (2014) 594–626.

[21] E. Hairer, S. P. Nørsett, G. Wanner, Solving ordinary differential equations I: nonstiff problems, volume 1, Springer Science & Business, 2008.

[22] J. C. Butcher, Numerical methods for ordinary differential equations. 2nd revised ed., 2nd revised ed. ed., Hoboken, NJ: John Wiley & Sons, 2008. doi:10.1002/9780470753767.

[23] P. J. Van Der Houwen, Construction of integration formulas for initial value problems, North Holland, 1977.

[24] A. Abdulle, Chebyshev methods based on orthogonal polynomials, Ph.D. thesis, 2001.

[25] M. Bakker, Analytisch Aspekten Van Een Minimaxprobleem, CWI Technical Report Stichting Mathematisch Centrum. Toegepaste Wiskunde-TN 62/71, 1971. URL: http://oai.cwi.nl/oai/asset/7896/7896A.pdf.

[26] H. Lomax, On the construction of highly stable, explicit numerical methods for integrating coupled ODEs with parasitic eigenvalues, NASA Technical Note NASAIN D/4547 (1968).

[27] W. Riha, Optimal stability polynomials, Computing 9 (1972) 37–43.

[28] V. Lebedev, How to solve stiff systems of differential equations by explicit methods, Numerical methods and applications (1994) 45–80.

[29] V. Lebedev, S. Finogenov, Utilization of ordered Chebyshev parameters in iterative methods, USSR Computational Mathematics and Mathematical Physics 16 (1976) 70–83.

[30] J. Verwer, W. Hundsdorfer, B. Sommeijer, Convergence properties of the Runge-Kutta-Chebyshev method, Numerische Mathematik 57 (1990) 157–178.

[31] E. Hairer, G. Wanner, Solving ordinary differential equations II: stiff and differential-algebraic problems, 1996.

[32] D. I. Ketcheson, L. Lóczi, M. Parsani, Propagation of internal errors in explicit Runge-Kutta methods and internal stability of ssp and extrapolation methods, arXiv preprint arXiv:1309.1317 (2013).

[33] V. I. Lebedev, S. Finogenov, Solution of the problem of parameter ordering in Chebyshev iteration methods, Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki 13 (1973) 18–33.

[34] G. Marchuk, V. I. Lebedev, Numerical methods in the theory of neutron transport, Harwood Academic Pub., New York, NY, 1986.

[35] W. Hundsdorfer, J. G. Verwer, Numerical solution of time-dependent advection-diffusion-reaction equations, volume 33, Springer, 2003.

[36] J. Martín-Vaquero, A. Khaliq, B. Kleefeld, Stabilized explicit Runge-Kutta methods for multi-asset American options, Computers & Mathematics with Applications 67 (2014) 1293–1308.

[37] F. Castella, P. Chartier, S. Descombes, G. Vilmart, Splitting methods with complex times for parabolic equations, BIT Numerical Mathematics 49 (2009) 487–508.

[38] E. Hansen, A. Ostermann, High order splitting methods for analytic semigroups exist, BIT Numerical Mathematics 49 (2009) 527–542.

[39] E. Hansen, A. Ostermann, Dimension splitting for quasilinear parabolic equations, IMA Journal of Numerical Analysis 30 (2010) 857–869.

[40] P. Dörsek, E. Hansen, High order splitting schemes with complex timesteps and their application in mathematical finance, Journal of Computational and Applied Mathematics 262 (2014) 234–243.

[41] R. I. McLachlan, G. R. W. Quispel, Splitting methods, Acta Numerica 11 (2002) 341–434.

[42] U. M. Ascher, S. J. Ruuth, R. J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, Applied Numerical Mathematics 25 (1997) 151–167.

[43] L. Shampine, B. Sommeijer, J. Verwer, IRKC: an IMEX solver for stiff diffusion-reaction PDEs, Journal of computational and applied mathematics 196 (2006) 485–497.

[44] A. Abdulle, G. Vilmart, PIROCK: a swiss-knife partitioned implicit-explicit orthogonal Runge-Kutta Chebyshev integrator for stiff diffusion-advection-reaction problems with or without noise, Journal of Computational Physics 242 (2013) 869–888.

[45] H. D. l. Cruz, R. Biscay, F. Carbonell, J. Jimenez, T. Ozaki, Advanced numerical algorithms-Local Linearization Runge-Kutta (llrk) methods for solving ordinary differential equations, Lecture Notes in Computer Science 3991 (2006) 132–139.

[46] H. De la Cruz, R. Biscay, J. C. Jiménez, F. Carbonell, Local Linearization Runge-Kutta methods: A class of A-stable explicit integrators for dynamical systems, Mathematical and Computer Modelling 57 (2013) 720–740.

[47] P. J. Van Der Houwen, B. P. Sommeijer, Parallel iteration of high-order runge-kutta methods with stepsize control, Journal of Computational and Applied Mathematics 29 (1990) 111–127.

[48] B. Sommeijer, An explicit Runge-Kutta method of order twenty-five, CWI Quarterly 11 (1989) 75–82.

[49] M. Warnez, B. Muite, Reduced temporal convergence rates in high-order splitting schemes, arXiv preprint arXiv:1310.3901 (2013).

[50] C. Lubich, A. Ostermann, Interior estimates for time discretizations of parabolic equations, Applied numerical mathematics 18 (1995) 241–251.

[51] C. Lubich, C. Makridakis, Interior a posteriori error estimates for time discrete approximations of parabolic problems, Numerische Mathematik 124 (2013) 541–557.

[52] W. Hundsdorfer, J. Verwer, A note on splitting errors for advection-reaction equations, Applied Numerical Mathematics 18 (1995) 191–199.

[53] S. Blanes, F. Casas, P. Chartier, A. Murua, Optimized high-order splitting methods for some classes of parabolic equations, Mathematics of Computation 82 (2013) 1559–1576.

[54] R. Lefever, G. Nicolis, Chemical instabilities and sustained oscillations, Journal of theoretical Biology 30 (1971) 267–284.

[55] S. D. Cohen, A. C. Hindmarsh, CVODE, a stiff/nonstiff ODE solver in C, Computers in physics 10 (1996) 138–143.

[56] G. Strang, On the construction and comparison of difference schemes, SIAM Journal on Numerical Analysis 5 (1968) 506–517.

Appendix A. Scheme patterns $M = 20$

$$
\begin{aligned}
d_0^2 &= \frac{267}{400} \\
d_1^2 &= -\frac{1}{1800} \\
d_2^2 &= \frac{1201}{7200}
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
d_0^4 &= \frac{3126039467}{6144000000} \\
d_1^4 &= \frac{244573733}{7680000000} \\
d_2^4 &= \frac{3212226667}{15360000000} \\
d_3^4 &= -\frac{63194381}{7680000000} \\
d_4^4 &= \frac{789861181}{61440000000}
\end{aligned}
\tag{A.2}
$$

$$
\begin{aligned}
d_0^6 &= \frac{7446093942631413209}{17915904000000000000} \\
d_1^6 &= \frac{158532158867283313}{2985984000000000000} \\
d_2^6 &= \frac{1022936325403301087}{4777574400000000000} \\
d_3^6 &= -\frac{35821864811075087}{1074954240000000000} \\
d_4^6 &= \frac{1048968349471238687}{35831808000000000000} \\
d_5^6 &= -\frac{32100268736824717}{17915904000000000000} \\
d_6^6 &= \frac{180240686854539517}{214990848000000000000}
\end{aligned}
\tag{A.3}
$$

Appendix B. Splitting schemes

Table B.4: Complex operator splitting parameters for $N = 2$, 4, 6 [56, 37, 53]. The final row for each quoted value of N lists: J, the number of distinct sweep configurations required; $k_1 \cdots k_J$, the sequence of J sweeps, labeled by j, required for a single extended interval to order-N. The remaining rows are in pairs listing: j, the index of the distinct sweep; $\Re(T_j)$, the real component of the sweep timescale; $\Im(T_j)$, the second row lists the imaginary part of the sweep timescale.

N	j	$\Re(T_j)$ $\Im(T_j)$
	J	$k_1 \cdots k_J$
2	1	1.0 *0.0*
	2	0.5 *0.0*
	3	2 1 2
4	1	1/4 *0*
	2	1/10 *−1/30*
	3	4/15 *2/15*
	4	4/15 *−1/5*
	9	2 1 3 1 4 1 3 1 2
6	1	0.0625 *0.0*
	2	0.024 694 876 087 018 064 640 910 864 996 842 247 838 60 *−0.007 874 795 562 906 877 058 171 577 949 526 942 163 20*
	3	0.063 813 474 021 302 699 779 366 304 188 200 146 963 20 *0.035 365 761 034 143 327 804 629 404 649 714 741 812 70*
	4	0.068 425 094 030 316 441 970 397 007 821 744 684 058 50 *−0.062 262 244 450 748 676 995 332 540 644 447 596 046 10*
	5	0.088 047 701 092 267 837 626 997 195 869 408 667 577 20 *0.045 473 871 502 298 704 383 762 549 187 977 426 444 69*
	6	0.023 689 611 129 847 060 696 141 912 470 009 364 325 33 *0.009 624 326 064 089 624 057 698 035 290 637 306 663 95*
	7	0.042 729 722 386 773 382 202 964 300 577 074 218 553 88 *−0.033 994 403 923 957 610 554 083 948 457 844 358 264 99*
	8	0.122 334 686 316 845 772 960 428 517 001 962 563 078 80 *−0.010 435 859 079 752 510 669 380 827 100 590 549 551 78*
	9	0.041 898 432 829 693 886 043 536 850 607 262 239 764 26 *0.069 362 492 631 696 384 275 158 174 307 144 262 130 30*
	10	0.048 732 804 211 869 708 158 514 092 934 991 735 680 80 *−0.090 518 296 429 724 730 488 558 538 566 128 582 051 30*
	33	2 1 3 1 4 1 5 1 6 1 7 1 8 1 9 1 10 1 9 1 8 1 7 1 6 1 5 1 4 1 3 1 2