A fully adaptive explicit stabilized integrator for advection-diffusion-reaction problems

Ibrahim Almuslimani

A fully adaptive explicit stabilized integrator for advection-diffusion-reaction problems

Ibrahim Almuslimani*

October 19, 2022

Abstract

A novel second order family of explicit stabilized Runge–Kutta–Chebyshev methods for advection–diffusion–reaction equations is introduced. The new methods outperform existing schemes for relatively high Peclet number due to their favorable stability properties and explicitly available coefficients. The construction of the new schemes is based on stabilization using second kind Chebyshev polynomials first used in the construction of the stochastic integrator SK-ROCK. An adaptive algorithm to implement the new scheme is proposed. This algorithm is able to automatically select the suitable step size, number of stages, and damping parameter at each integration step. Numerical experiments that illustrate the efficiency of the new algorithm are presented.

Keywords: advection-diffusion-reaction equations, explicit stabilized methods, Runge-Kutta Chebyshev methods, RKC, SK-ROCK, ARKC.

AMS subject classification (2010): 65L04, 65L20, 65M12

1 Introduction

In this paper we use the idea of stabilization by combining first and second kind Chebyshev polynomials introduced in [5] to derive explicit stabilized methods for advection–diffusion problems with, possibly, costly non-stiff reaction terms,

$$\partial_t u(x,t) = \nabla \cdot (D\nabla u(x,t)) - \nabla \cdot (\mathbf{v}u(x,t)) + r(u(x,t)), \quad (x,t) \in \Omega \times [0,T],$$

with initial and boundary conditions, where $\Omega \in \mathbb{R}^d$, D is the matrix of diffusion coefficients, and $\mathbf{v}$ is the velocity vector. The function r represents non-stiff, but possibly costly, reaction terms. Note that in general, D and $\mathbf{v}$ may also depend on u leading to nonlinear diffusion and advection terms. In the linear one dimensional setting, the equation reduces to

$$\partial_t u(x,t) = d\partial_x^2 u(x,t) - a\partial_x u(x,t) + r(u(x,t)), \quad (x,t) \in \Omega \times [0,T] \tag{1}$$

where d and a are positive reals, and Ω is a real interval. The Peclet number is defined by a/d and is allowed here to be quite large. When discretizing the partial differential equation (PDE) (1) in space using centered finite difference for example, with mesh size Δx, We obtain a system of ordinary differential equations (ODEs) of the form

$$\dot{y}(t) = F_D(y(t)) + F_A(y(t)), \qquad y(0) = y_0 \in \mathbb{R}^d, \qquad t \in [0,T], \tag{2}$$

*Univ Rennes, INRIA Rennes, IRMAR - UMR 6625, F-35000 Rennes, France.
Ibrahim.Almuslimani@univ-rennes1.fr.

where F_D represents the diffusion term with eigenvalues of its Jacobian grow as $1/\Delta x^2$ on the negative real axis, and F_A represents the advection term (and possibly non-stiff reaction terms) with eigenvalues of its Jacobian are of size $1/\Delta x$ and located close to the imaginary axis and symmetric with respect to the origin. This means that the eigenvalues of the Jacobian of the obtained system are approximately located in an ellipse with the length of its minor axis proportional to the square root of the length of the major axis.

Explicit stabilized Runge–Kutta–Chebyshev methods were originally introduced in the context of purely diffusive or diffusion dominated advection–diffusion problems (very small Peclet number) as a compromise between costly implicit methods and restrictive usual explicit schemes [1, 2, 4, 8, 25]. Due to their versatility, they were extended to many other types of problems such as advection–diffusion–reaction equations [9, 22, 27, 28, 29], stochastic differential equations (SDEs) [5, 6, 7, 10], and optimal control problems [12]. Other types of stabilized methods were studied in [23, 18, 19].

Typically, the stability domain of an explicit stabilized method contains a long narrow strip around the negative real axis. In the context of advection–diffusion problems, the authors of the article [28] propose the usage of the RKC method with very large damping parameter to make the strip wider, which means that eigenvalues with slightly larger imaginary parts coming from the advection terms can be put in. This comes at the cost of a serious shortening of the strip, which means that, for a fixed time step, less eigenvalues with negative real parts can be put in. Later, a partitioned Runge–Kutta–Chebyshev method (PRKC) of order 2 was designed in [29] based on the RKC method [20] for the integration of ODEs that have a moderately stiff term (diffusion) and non-stiff terms (advection or costly reaction terms). PRKC has a limited stability for the advection term, and it shares with the standard RKC the same stability domain length over the negative real axis. In [9], the authors propose a partitioned implicit–explicit orthogonal Runge–Kutta method (called PIROCK) for the time integration of advection–diffusion–reaction problems with possibly severely stiff reaction terms and stiff stochastic terms. The diffusion terms are solved by the explicit, nearly optimal, second order orthogonal Chebyshev method (ROCK2). Applied to advection–diffusion problems, the method has order 2 of accuracy and can handle the large Peclet number regime but it needs very large damping that reduces a lot its stability domain length over the negative real axis. In addition, PIROCK relies on the ROCK2 method, for which no explicit formulas are available to compute the coefficients for a given stage number. Recently, the authors of [22] developed an improved version of the RKC method (called IMPRKC) for advection–diffusion–reaction equations. Their idea is based on introducing an appropriate combination of RKC polynomials which leads to a significant increase of the width of the stability domain along the imaginary axis with almost no loss of its length along the real axis. This comes at the cost of a few additional function evaluations. The main drawback of the IMPRKC scheme is evaluating F_A at each stage of the method. This rich literature shows that the domain of stabilized schemes is very active, and that the construction of an adaptively efficient explicit stabilized integrator for such important class of problems is quite challenging.

In [11, Sect. 3.7.2], the author of the present paper profited from the idea of stabilization using second kind Chebyshev polynomials, first introduced in [5] in the context of SDEs, to construct a first order explicit stabilized method for advection–diffusion–reaction equations with optimal stability domain. What made this quite intuitive is the similarity between mean square stability for SDEs and the stability of the test equation for ODEs of the form (2) (see Sect. 4.1). In this work, we construct a second order integrator based on RKC that outperforms the existing methods in the literature. We propose a fully adaptive algorithm to implement the new second order method.

This paper is organized as follows: in Section 2, we give a fast revision on explicit stabilized methods. In Section 3 we present an *optimal* first order explicit stabilized method for advection–diffusion–reaction equations that we constructed in the thesis [11] inspired by previous work on

SDEs [5]. In Section 4 we derive and analyze our new second order adaptive scheme in terms of stability and convergence, and we propose a local error estimator for automatic step size selection. Section 5 is dedicated to present and analyze some numerical experiments that illustrate the efficiency of the new schemes. Finally, we conclude in Section 6.

2 Preliminaries on explicit stabilized methods

This section is devoted to present useful standard materials.

In order to study the stability of a Runge–Kutta integrator applied to an ODE, the following approach is widely used [4, 16]. Consider the test ODE

$$\dot{y}(t) = \lambda y(t), \quad y(0) = y_0, \tag{3}$$

where $\lambda \in \mathbb{C}$ with negative real part. If we apply a Runge–Kutta method with step size h on (3), we get the relation $y_{n+1} = R(h\lambda)^n y_0$, where the rational function $R(z)$ is called the stability function. Hence, the stability domain of the method is defined as

$$\mathcal{S} := \{z \in \mathbb{C}; |R(z)| \leq 1\}.$$

In the particular case of explicit methods, $R(z)$ is a polynomial which means that the stability domain is necessarily bounded. For example, the stability domain of the explicit Euler method is just a disk of radius 1 which explains the time step restriction it faces for stiff ODEs, while that of the implicit Euler method is the complementary of a disk of radius 1. This shows the advantage of implicit methods in terms of stability, however, for large dimensional problems and especially the nonlinear and ill-conditioned ones, implicit methods become very costly and difficult to implement. Here appears the need to a compromise between classical explicit methods and implicit integrators.

This compromise is "explicit stabilized methods" (see the survey [4]). The idea is to construct explicit Runge–Kutta integrators with extended stability domain that grows quadratically with the number of stages s of the method along the negative real axis, and then allows to use large time steps typically for problems arising from *diffusion dominant* advection–diffusion–reaction PDEs for which the eigenvalues are close to the negative real axis and are very large in modulus.

Before proceeding, let us recall some useful facts on Chebyshev polynomials. The first kind Chebyshev polynomials are defined by

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_j(x) = 2xT_{j-1}(x) - T_{j-2}(x), \quad j \geq 2. \tag{4}$$

The second kind Chebyshev polynomials are defined by

$$U_0(x) = 1, \quad U_1(x) = 2x, \quad U_j(x) = 2xU_{j-1}(x) - U_{j-2}(x), \quad j \geq 2. \tag{5}$$

Moreover, the two kinds polynomials satisfy the following

$$U_{j-1}(x) = \frac{T_j'(x)}{j}. \tag{6}$$

The stabilization procedure is based on the above relations. The fact that both kinds share the same recurrence relation will be very useful in our analysis. Indeed, this allows to simultaneously derive the recurrence formulas of the methods, otherwise, the cost would be doubled.

It was shown that for any explicit, consistent (order 1) Runge–Kutta method, the maximum stability domain length over the negative real axis is $2s^2$, where s is the number of stages of the method. The polynomial that achieves this *optimal* length is the shifted Chebyshev polynomial $T_s(1 + z/s^2)$. See, for example, [17, Chap. V, Th. 1.1]. For robustness reasons, a damping of this polynomial is introduced and the resulting scheme is recalled in the following subsection.

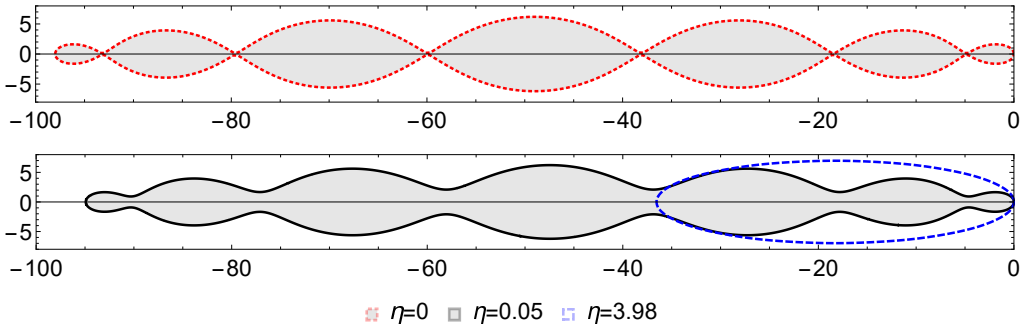


Figure 1: Stability domains of the Chebyshev method (8) for $s = 7$ and different damping values $\eta = 0, 0.05, 3.98$.

2.1 Optimal first order Chebyshev methods

Consider the ODE

$$\dot{y} = f(y), \qquad y(0) = y_0, \qquad t \in [0, T]. \tag{7}$$

Given y_0, in order to compute $y_1 \approx y(h)$ using the optimal first order Chebyshev method applied to (7) with step size h, the following recurrence is applied

$$\begin{aligned} K_0 &= y_0, \quad K_1 = K_0 + \mu_1 h f(K_0), \\ K_j &= \mu_i h f(K_{j-1}) + \nu_i K_{j-1} + (1-\nu_i) K_{j-2}, \quad j = 2, \ldots, s \\ y_1 &= K_s, \end{aligned} \tag{8}$$

where $\omega_0 := 1 + \frac{\eta}{s^2}$, $\omega_1 := \frac{T_s(\omega_0)}{T_s'(\omega_0)}$, and

$$\mu_1 := \frac{\omega_1}{\omega_0}, \quad \mu_j := \frac{2\omega_1 T_{j-1}(\omega_0)}{T_j(\omega_0)}, \quad \nu_j := \frac{2\omega_0 T_{j-1}(\omega_0)}{T_j(\omega_0)}, \quad j = 2, \ldots, s. \tag{9}$$

The parameter η is called the damping parameter and it is necessary to avoid singularities in the stability domain which ensures the robustness of the method (See Figure 1). Typically for this method, η is fixed to 0.05. It can be easily verified, using the recurrence (4) and proceeding by induction, that applied to the test problem (3), the above method produces after one step $y_1 = K_s = R_s(h\lambda) y_0$ with

$$R_s(z) = \frac{T_s(\omega_0 + \omega_1 z)}{T_s(\omega_0)},$$

for which the stability domain contains a narrow strip around the interval $[-C_\eta s^2, 0]$ with

$$C_\eta = \frac{1 + \omega_0}{s^2 \omega_1} \simeq 2 - 4/3\eta$$

is very close to 2 (the optimal value for order 1). For $\eta = 0$ the stability function is again $T_s(1 + z/s^2)$. The method has low memory requirements (only two stages have to be stored) and reasonable propagation of round-off errors even for large values of s needed in practice [25, 26]. The fact that the length of the stability domain on the negative real axis enjoys a quadratic growth with respect to the number of stages s is crucial to the success of explicit stabilized Runge–Kutta methods.

2.2 Second order RKC methods

To design a second order method, we need the stability polynomial to satisfy*

$$R(z) = 1 + z + \frac{z^2}{2} + \mathcal{O}(z^3).$$

A correction to the first order shifted Chebyshev polynomials was introduced to construct a stabilized scheme of order 2 [13, 24, 26]. The obtained second order polynomial is the following

$$R_s(z) = a_s + b_s T_s(\omega_0 + \omega_2 z),$$

where,

$$a_s = 1 - b_s T_s(\omega_0), \quad b_s = \frac{T_s''(\omega_0)}{(T_s'(\omega_0)^2)}, \quad \omega_0 = 1 + \frac{\eta}{s^2}, \quad \omega_2 = \frac{T_s'(\omega_0)}{T_s''(\omega_0)}, \quad \eta = 0.15. \tag{10}$$

For each s, $|R_s(z)|$ remains bounded by $a_s + b_s = 1 - \eta/3 + \mathcal{O}(\eta^2)$ for z in the stability interval (except for a small interval near the origin). The stability interval along the negative real axis is $[-\frac{1+\omega_0}{\omega_2}, 0]$ which is approximately $[-0.65s^2, 0]$, and covers about 80% of the optimal stability interval for second order stability polynomials, and the formula now for calculating s for a given time step h is

$$s := \left[\sqrt{\frac{h\lambda_{\max} + 1.5}{0.65}} + 0.5 \right],$$

where the brackets mean rounding to the nearest integer, and $\lambda_{\max}$ is the spectral radius of the Jacobian of f that can be calculated at each step using power method for example. Using the recurrence relation of the Chebyshev polynomials, the RKC method as introduced in [26] is defined by

$$\begin{aligned} &K_0 = y_0, \quad K_1 = K_0 + h b_1 \omega_2 f(K_0), \\ &K_j = \mu_j h (f(K_{j-1}) - a_{j-1} f(K_0)) + \nu_j K_{j-1} + \kappa_j K_{j-2} + (1 - \nu_j - \kappa_j) K_0, \\ &y_1 = K_s, \end{aligned} \tag{11}$$

where

$$\mu_j = \frac{2 b_j \omega_2}{b_{j-1}}, \quad \nu_i = \frac{2 b_j \omega_0}{b_{j-1}}, \quad \kappa_j = -\frac{b_j}{b_{j-2}}, \quad b_j = \frac{T_j''(\omega_0)}{T_j'(\omega_0)^2}, \quad a_j = 1 - b_j T_j(\omega_0), \tag{12}$$

for $j = 2, \ldots, s$. The parameters b_0 and b_1 are free ($R_0(z)$ is constant and only order 1 is possible for R_1(z)) and the values $b_0 = b_1 = b_2$ are suggested in [21].

Remark 2.1. *For simplicity of the presentation, we consider the case of autonomous problems (with f independent of time) but we highlight that our approach also applies straightforwardly to non-autonomous problems $\dot{y}(t) = f(t, y(t))$. Indeed, a standard approach is to consider the augmented system with $z(t) = t$, i.e. $\frac{dz}{dt} = 1$, $z(0) = 0$ and define $\tilde{y}(t) = (y(t), z(t))^T$, see e.g. [14, Chap. III] for details.*

*Indeed, up to order two, the order conditions for nonlinear problems are the same as the order conditions for linear problems [14, Chap. III].

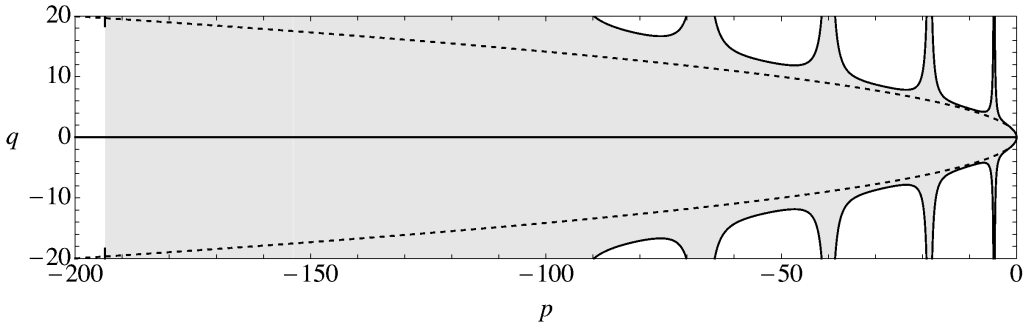


Figure 2: Stability domain of the new optimal first order method (15) in the $p-q$ plane for $s = 10$ and $\eta = 0.05$. The dashed lines correspond to $\pm\sqrt{-2p}$.

3 Optimal first order scheme

Note that we have discussed the content of this section in the thesis [11], but we recall it here since it gives insight about the construction of the second order adaptive integrator in the next section.

Consider the linear test problem

$$\dot{y} = \lambda y + i\mu y, \qquad y(0) = y_0, \tag{13}$$

where $\lambda \in \mathbb{R}^-$, $\mu \in \mathbb{R}$, and $i = \sqrt{-1}$. Applying a Runge–Kutta method to the above equation, one gets an induction of the form

$$y_{n+1} = R(p, q)y_n,$$

with $p = h\lambda$ and $q = h\mu$. We define the stability domain of a Runge–Kutta method applied to (13) by

$$\mathcal{S} = \{(p, q) \in \mathbb{R}^2 \ ; \ |R(p, q)| \leq 1\}.$$

Equation (13) can be seen as the test equation for linear SDEs with the $i\mu$ replacing the noise. Hence, inspired by SK-ROCK [5], we consider the following stability polynomial

$$R(p, q) = A(p) + B(p)iq := \frac{T_s(\omega_0 + \omega_1 p)}{T_s(\omega_0)} + \frac{U_{s-1}(\omega_0 + \omega_1 p)}{U_{s-1}(\omega_0)}(1 + \frac{\omega_1}{2}p)iq, \tag{14}$$

where T_s and U_s are the first and the second kind Chebyshev polynomials of degree s (the number of stages), and the coefficients ω_0 and ω_1 are the same as for the Chebyshev method (8). The stability condition $|R(p, q)| \leq 1$ is equivalent to $A(p)^2 + B(p)^2q^2 \leq 1$ which is exactly the mean square stability condition described in [5, Sect. 2]. By [5, Theorem 3.2] and [5, Remark 3.6], for $\eta > 0$ and $s \in \mathbb{N}$, $|R(p, q)| \leq 1$ for all $p \in [-2\omega_1^{-1}, 0]$ and all q such that $|q| \leq \sqrt{-2p}$ (See Figure 2).

Remark 3.1. *For SDEs, the condition $|q| \leq \sqrt{-2p}$ guarantees the stability of the solution of the continuous problem, which means that numerical stability under this condition is sufficient. However, for our test equation* (13)*, the exact solution is stable when $\lambda \leq 0$ and $\mu \in \mathbb{R}$, hence even when $|q| \leq \sqrt{-2p}$ is violated. Although stability under this condition is already a significant improvement comparing to standard explicit methods, we can use larger damping parameter η to increase the width of the stability region along the imaginary direction in the case of large advection.*

The new order one method for space discretized advection–diffusion–reaction equations (2), is defined in the same way as the SK-ROCK method [5], by replacing the noise term by the

advection-reactions terms,

$$\begin{aligned} K_0 &= y_0 \\ K_1 &= y_0 + \mu_1 h F_D(y_0 + \nu_1 h F_A(y_0)) + \kappa_1 h F_A(y_0) \\ K_j &= \mu_j h F_D(K_{j-1}) + \nu_j K_{j-1} + \kappa_j K_{j-2}, \quad j = 2, \ldots, s, \\ y_1 &= K_s, \end{aligned} \tag{15}$$

where $\nu_1 = s\omega_1/2$, $\kappa_1 = s\omega_1/\omega_0$ and the rest of the coefficients are identical to those defined in (9). Assuming enough regularity on F_D and F_A, the convergence proof is straightforward and based on [5, Lemma 4.2]. The above scheme is also optimal in the sense that its stability region achieves the maximum possible length over the negative real axis for an explicit consistent scheme.

Note that the method requires only 1 evaluation of the advection-reaction terms F_A per time step, while other existing methods for advection diffusion problems such as PIROCK [9] and PRKC [29] require 3 and 4 evaluations of F_A respectively per time step. It also outperform the mentioned methods by far in terms of stability. However, it has only order 1 of accuracy, which motivates the construction of a second order integrator that competes with existing schemes in terms of stability and convergence, taking advantage of the above analysis.

Note that we can adaptively increase the stability for advection by increasing η in the price of loosing some length on the negative real axis. This will be explained in details in the next section for the second order scheme, for which this adaptive increase of damping is a key feature.

4 New second order scheme

In this section, we introduce the new adaptive second order integrator based on the material presented in the first 3 sections and inspired by the SK-ROCK method introduced in [5].

ARKC integrator The ARKC integrator (for adaptive RKC) applied to (2) is defined as follows for $s \geq 2$:

$$\begin{aligned} G &= hF_A(y_0 + \frac{h}{2}F_A(y_0 + \frac{\omega_2}{2}hF_D(y_0)) + \frac{h}{2}F_D(y_0)) + hF_D(y_0 + \frac{\omega_2 - 1}{2}hF_A(y_0)) \\ &\quad - hF_D(y_0), \\ K_{-1} &= y_0, \\ K_0 &= K_{-1} + \frac{\omega_2}{2}G, \\ K_1 &= K_0 + b_1\omega_2 hF_D(y_0) + \alpha G, \quad \text{and for } j = 2, \ldots, s, \\ K_j &= \mu_j h(F_D(K_{j-1}) - F_D(K_0) + (1 - a_{j-1})F_D(y_0)) + \nu_j K_{j-1} + \kappa_j K_{j-2} \\ &\quad + (1 - \nu_j - \kappa_j)K_0, \\ y_1 &= K_s, \end{aligned} \tag{16}$$

where $\alpha = \left(1 - \frac{\omega_2}{2}\right) b_1 s\omega_2$, and the other coefficients are identical to those defined in (10) and (12). Note that for purely diffusive equations ($F_A \equiv 0$), the method reduces to standard RKC (11).

Complexity The method requires $s + 2$ evaluations of F_D and 3 evaluations of F_A per time step. The standard RKC and the PRKC methods both require s evaluations of F_D per time step, but RKC requires s evaluations of F_A while PRKC requires only 4. The PIROCK merhod needs $s + 2 + l$ evaluations of F_D ($l = 1$ or 2) and 3 evaluations of F_A per time step. Finally, the IMPRKC scheme requires $s + \hat{s}$ evaluations of each F_D and F_A, where $\hat{s}$ becomes quite large for

very stiff problems. The advantage of our new scheme comes from having larger stability region that changes adaptively, which allows to use bigger time steps.

Construction In order to construct a second order scheme, we need our stability function to satisfy the following equality

$$\begin{aligned} R_2(p,q) &= 1 + p + iq + \frac{1}{2}(p+iq)^2 + \mathcal{O}(((p+iq)^3) \\ &= 1 + p + iq + \frac{p^2}{2} + ipq - \frac{q^2}{2} + \mathcal{O}((p+iq)^3). \end{aligned} \tag{17}$$

A natural approach would be to modify the last stage of the first order scheme to reach second order, but such naive modification will cause severe instability. Our idea is to start the stabilization using second kind Chebyshev polynomials from the beginning of the integration process. Hence, inspired by the previous section, we consider the following polynomial

$$\begin{aligned} R_2(p,q) &= A_2(p) + B_2(p)(iq - \frac{q^2}{2}) \\ &:= a_s + b_s T_s(\omega_0 + \omega_2 p) \\ &+ \left(\frac{\omega_2}{2} + \left(1 - \frac{\omega_2}{2}\right) \frac{U_{s-1}(\omega_0 + \omega_2 p)}{U_{s-1}(\omega_0)} \right) (1 + \frac{\omega_2}{2} p)(iq - \frac{q^2}{2}), \end{aligned} \tag{18}$$

where all the coefficients are defined in (10), from which we can derive the new adaptive second order ARKC method (16) for advection–diffusion–reaction problems, using the relations (4), (5), and (6). The first part of our stability function (18) corresponds to the stability polynomial of the standard second order RKC scheme, while the second part involves second kind Chebyshev polynomials (as in (14)) in order to stabilize the advection part, and has the correct order. The construction of the method from the polynomial (18) is done by induction using the relations (4)-(6). See also Lemma 4.2 below.

Unlike the case of standard ODEs where the term $z^2/2$ is enough to have order two for nonlinear problems, some additional *coupling* conditions need to be satisfied here due to the partitioned nature of the scheme (see [14, Sect. III.2]). These conditions result from the fact that the term ipq in (17) is in fact the sum of two terms: $\frac{1}{2}ipq$ and $\frac{1}{2}iqp$, which are not necessarily equal for multidimensional and/or nonlinear problems. The quantity G is constructed carefully to get order 2 of convergence for general nonlinear problems. For linear equations, $G = (1 + \frac{\omega_2}{2})(iq - \frac{q^2}{2})$. Analogously to some existing second order methods [9, 28], our damping parameter is not fixed to a small value, but it is an increasing function of the stage number s. However, for relatively large Peclet numbers, no huge damping is needed, and the method still perform very well as will be shown in Section 5.

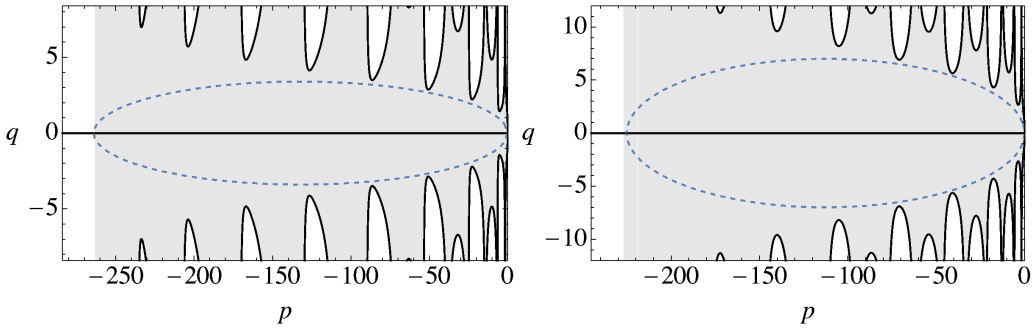
4.1 Stability analysis

Remark 4.1. *The stability analysis of Runge–Kutta type methods is usually made on linear test problems, but we emphasize that such test equations only give insight on the stability of the stabilized method under study (here ARKC). This is because practical problems are usually nonlinear, and even for linear problems, the involved operators cannot in general be diagonalized simultaneously.*

Lemma 4.2. *The scheme* (16) *applied to the linear test problem* (13) *with step size h, produces the following recurrence*

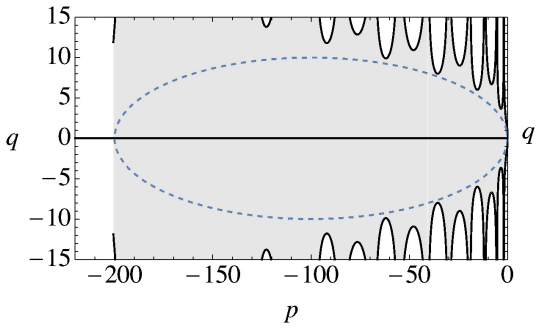
$$y_{n+1} = R_s(h\lambda, h\mu) y_n,$$

where the stability polynomial $R_s(p,q)$ is defined in (18).

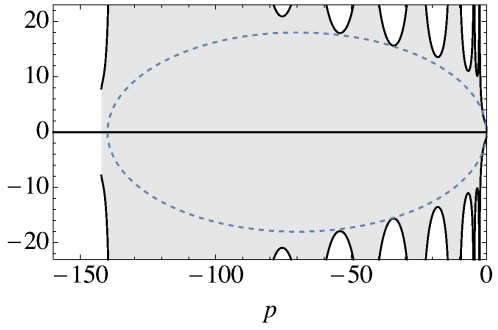


(a) $\eta = 0.15$, $\delta_s = 0.65s^2$, $\alpha_s = 0.17s$.

(b) $\eta = 1.5$, $\delta_s = 0.56s^2$, $\alpha_s = 0.35s$.



(c) $\eta = 3$, $\delta_s = 0.5s^2$, $\alpha_s = 0.5s$.



(d) $\eta = 10$, $\delta_s = 0.35s^2$, $\alpha_s = 0.9s$.

Figure 3: Illustration of the stability regions of the ARKC method (16) in the $p - q$ plane for $s = 20$ and different values of the damping parameter η. For each stability region, we plot the largest possible ellipse that fits inside.

Proof. By induction on j, it can be shown that, for every $j \geq 1$, the internal stages satisfy

$$K_j = R_j(p, q) y_n,$$

where,

$$R_j(p,q) \coloneqq a_j + b_j T_j(\omega_0 + \omega_2 p) + \left(\frac{\omega_2}{2} + \left(1 - \frac{\omega_2}{2}\right) U_{j-1}(\omega_0 + \omega_2 p)\right) b_j s \omega_2 (1 + \frac{\omega_2}{2} p)(iq - \frac{q^2}{2}).$$

For $j = s$, we have that $b_s s\omega_2 = 1/U_{s-1}(\omega_0)$ because $s^{-1}T_s'(\omega_0) = U_{s-1}(\omega_0)$ and the proof is done. □ □

In contrast to the first order scheme (15) and the stochastic integrator SK-ROCK [5], the damping parameter for the ARKC method is not fixed. Indeed, it is an increasing function of the Peclet number and the number of stages s. We provide numerical stability analysis that illustrates the nice features of the new scheme (see Figures 3 and 4).

In what follows, we will respectively denote by δ_s and α_s the half width and the half height of the largest ellipse that can be put in the stability region of the corresponding method.

The method (16) is designed to handle quite large Peclet numbers, this means that we need the width of the stability region in the imaginary direction to be as large as possible. While the length over the negative real axis grows quadratically with s, we cannot achieve more than linear growth on the imaginary direction (this fact is given as an exercise in [16, Chap. IV]). In [28], the RKC

method is proposed with large damping value which reduces the length of the stability domain over the negative real axis to $\delta_s = 0.34s^2$, and leads to a growth of only $\mathcal{O}(\sqrt{s})$ for the ellipse half-height α_s. The PRKC method proposed in [29] uses the standard small damping $\eta = 0.15$ that keeps $\delta_s \approx 0.65s^2$, but α_s is fixed to 1.7 no matter how large is the number of stages s, which does not add much to the standard RKC method since this might be useful only in the small Peclet number regime. However, the main feature of the method is to reduce the number of evaluations of possible non stiff terms such as non-stiff advection or reaction terms. The PIROCK integrator [9], is notably better than the two mentioned methods RKC and PRKC. Two kinds of damping are proposed for PIROCK, the first lead to a nearly optimal length over the negative real axis $\delta_s \approx 0.81s^2$ but α_s is limited to $0.07696s + 1.878$. For the second damping, $\alpha_s = 0.5321s + 0.4996$ but δ_s reduces to $0.43s^2$. It shares as well the feature of reducing the evaluation of non diffusive terms. Nevertheless, PIROCK still have two drawbacks: the first one is the non-availability of explicit closed form formulas to compute its coefficients for a given number of stages s, the other is the limited choice of damping.

The stability region of the ARKC method changes adaptively with the spectrum of the Jacobian of the vector fields at each time step. This change is due to the damping parameter η which is not necessarily constant. In fact, the ellipse half width δ_s still grows quadratically with s, while the ellipse half height α_s grows linearly with s, in contrast to RKC and PRKC methods. For example, for the standard value of $\eta = 2/13$ used for RKC in [20] and PRKC in [29], δ_s is still equal to $0.65s^2$ and $\alpha_s \simeq 0.17s$ (see Figure 3a), whereas for PRKC α_s is fixed to 1.7 . In addition, increasing the value of the damping parameter η adds more space in the imaginary direction which is very favorable for the advection dominated problems. Figure 3c illustrates the stability region in the p-q plane for $s = 20$ and $\eta = 3$. We can see that $\delta_s = 0.5s^2$ and $\alpha_s = 0.5s$, while the best that PIROCK [9] could achieve for almost the same α_s is $\delta_s = 0.43s^2$ which makes difference for large values of s. For $\eta = 10$ we have $\delta_s = 0.35s^2$ and $\alpha_s = 0.9s$ as shown in Figure 3d, compare that with the case of standard RKC with infinite damping, considered in [28], where $\delta_s = 0.34s^2$ while $\alpha_s = \mathcal{O}(\sqrt{s})$. The last method to compare with is IMPRKC introduced in [22] where the authors provide a plot for $s = 50$ stages and $\hat{s} = 3$ additional stages. The length over the negative real axis stays almost the same as RKC at $0.65s^2$, while the width along the imaginary axis is around $0.8s$. As said before, the evaluation of the advection–reaction term at every step of IMPRKC is a disadvantage that limits its performance. For instance, for small Peclet numbers it needs time steps as much as the standard RKC scheme which makes it a bit more expensive because of the additional stages needed. In Section 5, our scheme ARKC is shown to perform better in all regimes.

4.2 Choice of damping

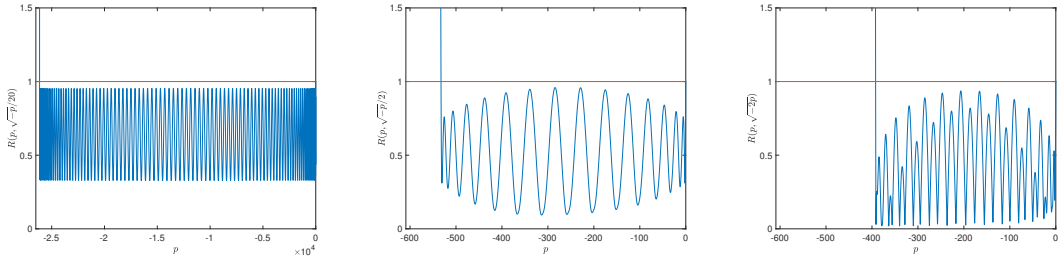
In fact, α_s is a function of s and η, this introduces additional difficulty in the estimation of the value of α_s. Therefore, we will introduce many choices of the range of Peclet number to simplify the implementation. Let ρ_D and ρ_A be the spectral radii of the Jacobians of F_D and F_A respectively. For ODEs coming from the discretization of the linear PDE (1), we have $\rho_D = 4d/\Delta x^2$ and $\rho_A = a/\Delta x$, then

$$\frac{\rho_A}{\sqrt{\rho_D}} = \frac{a}{2\sqrt{d}}.$$

We define the number

$$P'_e \coloneqq \frac{a}{\sqrt{d}},$$

hence, $\rho_A/\sqrt{\rho_D} = \frac{1}{2}P'_e$. Similarly, we can show that $q = \frac{1}{2}P'_e\sqrt{-hp}$. We will use this new number P'_e to adapt our choice of damping according to the parameters of the problem. Notice that this



(a) $\eta = 0.15$, $\delta_s = 0.65s^2$. (b) $\eta = 1$, $\delta_s = 0.59s^2$. (c) $\eta = 5$, $\delta_s = 0.4345s^2$.

Figure 4: Stability polynomial (18) for $s = 200$ (left) and $s = 30$ (middle and right), and different values of η and q.

is just another measure of the advection dominance that is proportional to the Peclet number and more convenient to use in our algorithm.

For $P'_e \leq 1/10$, i.e, $\rho_A/\sqrt{\rho_D} \leq 1/20$, we fix $\eta = 0.15$ up to $s = 200$, and for s between 200 and 500 we set $\eta = 0.6$ (we do not allow s to be more than 500). For $1/10 < P'_e \leq 1/2$ which means that $1/20 < \rho_A/\sqrt{\rho_D} \leq 1/4$, we consider the following choices for s and η:

$2 \leq s \leq 30$	$31 \leq s \leq 60$	$61 \leq s \leq 110$	$111 \leq s \leq 160$
$\eta = 0.2$	$\eta = 0.45$	$\eta = 1$	$\eta = 1.5$

$161 \leq s \leq 260$	$261 \leq s \leq 360$	$361 \leq s \leq 500$
$\eta = 2.4$	$\eta = 3$	$\eta = 4$

Table 1: $1/10 < P'_e \leq 1/2$, $1/20 < \rho_A/\sqrt{\rho_D} \leq 1/4$.

Tables for other values of P'_e are given in appendix A. Infinitely many choices could be done, but we will present and use only a few choices since they are enough for the method to perform very well. The methodology to compute the above values is easy, it is enough to plot $R_s(p, q)$ for the corresponding choice of q and to vary η in a way that we stay stable for the given value of s (see Figure 4).

4.3 Convergence analysis

In this section we will prove the second order of convergence of the scheme (16) when applied to ODEs of the form (2) arising from the discretization of advection–diffusion–reaction problems.

Theorem 4.3. *Let $T > 0$ and consider the system of ODEs (2) on the time interval $[0, T]$, where F_D and F_A are of class $\mathcal{C}^2$ and are Lipschitz continuous. Suppose in addition that the first and second derivatives of F_D and F_A are bounded. Let $N \in \mathbb{N}$, $h = T/N$, and $t_n = nh$, $n = 1, \ldots, N$, and consider the method (16) applied to (2) with step size h, such that the number of stages s and the damping parameter η are chosen appropriately to guarantee stability. Then, we have for all $n = 1 \ldots, N$,*

$$\|y(t_n) - y_n\| \leq Ch^2, \tag{19}$$

where C is independent of h and n. In other words, the method converges with order 2.

Proof. Let us prove first that the local error (the error after one step) satisfies

$$\|y(h) - y_1\| = \mathcal{O}(h^3). \tag{20}$$

Throughout the proof, $F_i(y_0)$ will be simply denoted by F_i, and $F_i'(y_0)F_j(y_0)$ will be denoted by $F_i'F_j$ with $i,j \in \{D,A\}$.

Using Taylor expansion, we can easily see that

$$G = hF_A + \frac{h^2}{2}F_A'F_A + \frac{h^2}{2}F_A'F_D + (\omega_2 - 1)\frac{h^2}{2}F_D'F_A + \mathcal{O}(h^3).$$

Now, let us suppose that

$$K_j = y_0 + \gamma_1^j hF_D + \gamma_2^j hF_A + \gamma_3^j h^2 F_D'F_D + \gamma_4^j h^2 F_D'F_A + \gamma_5^j h^2 F_A'F_D + \gamma_6^j h^2 F_A'F_A + \mathcal{O}(h^3). \quad (21)$$

The first two stages of the method satisfy:

$$K_0 = y_0 + \frac{\omega_2}{2}G, \quad K_1 = y_0 + b_1\omega_2 hF_D(y_0) + (\alpha + \frac{\omega_2}{2})G,$$

hence, we have, $\gamma_1^0 = 0$, $\gamma_1^1 = b_1\omega_2$, $\gamma_2^0 = \frac{\omega_2}{2}$, $\gamma_2^1 = \alpha + \frac{\omega_2}{2}$, $\gamma_3^0 = \gamma_3^1 = 0$, $\gamma_4^0 = \frac{\omega_2(\omega_2-1)}{4}$, $\gamma_4^1 = (\alpha + \frac{\omega_2}{2})\frac{\omega_2-1}{2}$, $\gamma_5^0 = \frac{\omega_2}{4}$, $\gamma_5^1 = \frac{1}{2}(\alpha + \frac{\omega_2}{2})$, $\gamma_6^0 = \frac{\omega_2}{4}$, and finally, $\gamma_6^1 = \frac{1}{2}(\alpha + \frac{\omega_2}{2})$. By performing a Taylor expansion of the stages K_j defined in (16) and replacing K_{j-1} and K_{j-2} by the expansion defined in(21), and finally identifying the coefficients, we get the following relations

$$\begin{aligned}
\gamma_1^j &= \mu_j(1 - a_{j-1}) + \nu_j\gamma_1^{j-1} + \kappa_j\gamma_1^{j-2},\\
\gamma_2^j &= \nu_j\gamma_2^{j-1} + \kappa_j\gamma_2^{j-2} + (1 - \nu_j - \kappa_j)\frac{\omega_2}{2},\\
\gamma_3^j &= \mu_j\gamma_1^{j-1} + \nu_j\gamma_3^{j-1} + \kappa_j\gamma_3^{j-2},\\
\gamma_4^j &= \mu_j(\gamma_2^{j-1} - \frac{\omega_2}{2}) + \nu_j\gamma_4^{j-1} + \kappa_j\gamma_4^{j-2} + (1 - \nu_j - \kappa_j)\frac{\omega_2(\omega_2 - 1)}{4},\\
\gamma_5^j &= \nu_j\gamma_5^{j-1} + \kappa_j\gamma_5^{j-2} + (1 - \nu_j - \kappa_j)\frac{\omega_2}{4},\\
\gamma_6^j &= \nu_j\gamma_6^{j-1} + \kappa_j\gamma_6^{j-2} + (1 - \nu_j - \kappa_j)\frac{\omega_2}{4}.
\end{aligned}$$

In order to prove (20), it is sufficient to show that $\gamma_1^s = \gamma_2^s = 1$ and $\gamma_3^s = \gamma_4^s = \gamma_5^s = \frac{1}{2}$. Obviously, $\gamma_6^j = \gamma_5^j$ for all $j \geq 0$. We will provide proofs for the first 3 coefficients, the other two can be done using similar arguments.

- For γ_1^j we have: $\gamma_1^0 = 0$, $\gamma_1^1 = b_1\omega_2$, and $\gamma_1^j = \mu_j(1 - a_{j-1}) + \nu_j\gamma_1^{j-1} + \kappa_j\gamma_1^{j-2}$, thus γ_1^j are the internal stages of the RKC method (11) applied to the problem $\dot{y} = 1$, $y(0) = 0$ with step size $h = 1$. This means that for all $0 \leq j \leq s$, $\gamma_1^j = c_j$, where c_j is the j^{th} node of the RKC method. Therefore, $\gamma_1^s = c_s = 1$. Moreover, according to [20, Sect. 2], for all $j = 2 \ldots, s$ we have
$$\gamma_1^j = c_j = \omega_2\frac{T_j''(\omega_0)}{T_j'(\omega_0)}, \qquad c_1 = \frac{c_2}{T_2'(\omega_0)} = b_1\omega_2, \quad c_0 = 0.$$

- It can be proved that γ_2^j, $j = 2, \ldots, s$ are given by $\gamma_2^j = \frac{\omega_2}{2} + \alpha\frac{b_j}{b_1}P_j(\omega_0)$, where $P_j(x)$ are polynomials that satisfy the following two term recurrence relation
$$P_0(x) = 0, \quad P_1(x) = 1, \quad P_j(x) = 2xP_{j-1}(x) - P_{j-2}(x), \ j \geq 2.$$
Comparing with the relation (5), it can be easily seen that for all $j \geq 1$,
$$P_j(x) = U_{j-1}(x) = T_j'(x)/j.$$
Hence, $\gamma_2^s = \frac{\omega_2}{2} + \alpha\frac{b_s}{sb_1}T_s'(\omega_0) = \frac{\omega_2}{2} + (1 - \frac{\omega_2}{2})\omega_2 b_s T_s'(\omega_0) = 1$.

- The proof for γ_3^j is very similar to that of γ_2^j. Indeed, we can prove for each $j = 0, \ldots, s$ the equality $\gamma_3^j = 2\omega_2^2 b_j Q_j(\omega_0)$, where

$$Q_0(x) = 0, \quad Q_1(x) = 0, \quad Q_j(x) = T'_{j-1} + 2xQ_{j-1}(x) - Q_{j-2}(x), \ j \geq 2.$$

  Using the relation (4), we observe that for all $j \geq 0$, $Q_j(x) = \frac{T''_j(x)}{4}$, which implies that $\gamma_2^s = 2\omega_2^2 b_s T''_s(\omega_0)/4 = 1$.

Thus, (20) is proved, and using regularity assumptions made on the vector fields, [15, Theorem 3.6] implies the global convergence estimate (19). □
□

4.4 Variable step size control and the fully adaptive algorithm

We introduce the following local error estimator that allows us to adaptively select the time step size in order to reach a given accuracy,

$$Est_{n+1} = C(12(y_n - y_{n+1}) + 6h(F_D(y_n) + F_A(y_n) + F_D(y_{n+1}) + F_A(y_{n+1}))),$$

where $C = 1/6 - c_2 + (1/2 - c_1)\zeta - 1/6\zeta$, with $\zeta = 0$ if $F_A \equiv 0$ and $\zeta = 1$ otherwise, and

$$c_1 = \frac{\omega_2}{2}\left(1 - \frac{\omega_2}{2}\right)\left(1 + \omega_2 \frac{U''_{s-1}(\omega_0)}{U_{s-1}(\omega_0)}\right), \quad c_2 = sb_s U''_{s-1}(\omega_0)\frac{\omega_2^3}{6}.$$

To get an intuition about the above coefficients, compare the third order term in the exact polynomial which is $(p + iq)^3/6 = p^3/6 - iq^3/6 + iqp^2/2 - pq^2/2$ and the third order term of stability polynomial (14) that is equal to $c_2 p^3 + c_1 iqp^2 - pq^2/2$. Note that the above estimator is inspired by the one considered for RKC in the paper [20], and they coincide for $\zeta = 0$. In contrast to the estimators introduced for PRKC [29] and PIROCK [9], we do not consider two separate estimators for F_D and F_A, since our method is defined in a different way. Indeed, for improved stabilization, the advection-reaction terms are computed at the beginning and not separately at the end, which makes the method more similar to RKC. We adopt the standard step size selection strategy proposed in [20] for RKC (see also [15, page 167]). Now, we are ready to present our fully adaptive algorithm,

Algorithm 4.4 ($y_0 \mapsto y_1$)**.** *Given a time step size h and an initial value y_0:*

- *Calculate ρ_A and ρ_D at the current value of the solution.*
- *Calculate $\rho_A/\sqrt{\rho_D}$ and choose the corresponding table among Tables 1, 3, 4, 5, 6, 7.*
- *Search the minimum s (and the corresponding η) in the chosen table such that $\frac{1+\omega_0}{\omega_1} > h\rho_D$.*
- *Generate the coefficients* (10) *and* (12) *and apply the recurrence* (16) *to calculate y_1.*
- *Update the step size according to the automatic step size selection procedure and repeat until reaching the final time.*

Remark 4.5. *To ensure stability, it is enough to choose s such that $\frac{1+\omega_0}{\omega_1} > h\rho_D$, because the relation between η and the corresponding range for s in each table is built to ensure that, once $-h\rho_D$ lies inside the stability domain, the whole ellipse containing the eigenvalues for the given spectral radii fits inside.*

Remark 4.6. *In the case where the eigenvalues have nonzero imaginary part and very small real part, this is close to purely advective regime, which is out of the scope of the paper. However, this case can be treated in two different ways, either by using the adaptive algorithm without any modification and then the error estimator will choose a very small time step, or by modifying the algorithm to integrate such systems (extremely large Peclet number) with RK3 or RK4 (for which the stability domain includes a part of the imaginary axis).*

For the calculation of the spectral radii ρ_D and ρ_A, we use the Matlab function "eig". In other programming languages, one can use nonlinear power method as in [3] for example. However, the cost of such methods is a different issue and is beyond the scope of the present paper. The code of the ARKC integrator as well as the drivers that reproduce the numerical experiments will be made publicly available on the page: https://sites.google.com/view/ ibrahim-almuslimani.

5 Numerical experiments

We will compare ARKC with IMPRKC, PIROCK, and PRKC as they were shown to outperform the other stabilized methods for advection–diffusion–reaction problems where the reaction term is not stiff.

5.1 Linear 1D advection–diffusion problem

An excellent example to compare the performance of ARKC with IMPRKC, PIROCK, and PRKC is the following 1-dimensional advection–diffusion equation with periodic boundary conditions

$$\begin{aligned} \partial_t u + a\partial_x u &= \partial_x^2 u, \\ u(x,0) &= \sin(2\pi x), \\ u(0,t) &= u(1,t), \end{aligned} \tag{22}$$

where a is a positive constant ($P_e = a$), $x \in [0,1]$, and $t \geq 0$. We discretize the space interval $[0,1]$ to a uniform grid $\{x_k\}_{k=0}^N$ with $x_k = kh$, and $h = 1/N$. We use second order central differences for the advection and the diffusion terms. We denote by $u_k(t)$ the approximation of $u(x_k,t)$, and the periodic boundary conditions propose that $u_0(t) = u_N(t)$. The eigenvalues of the obtained matrix are

$$\lambda_k = \frac{2}{h^2}(\cos(2k\pi h) - 1) - \frac{ia}{h}\sin(2k\pi h), \quad k = 1, \ldots, N,$$

and are located in an ellipse in the left half-plane $\mathbb{C}^-$, which makes the problem typical for the comparison of the three schemes.

For the numerical experiments, we take $N = 150$, $t \in [0, 1/2]$, and we fix $Atol = Rtol = tol$. The initial step is fixed to 10^{-3}. The number of rejected steps is always very small and thus neglected. We can clearly see in Table 2 that PRKC can compete with our scheme ARKC only in the very small Peclet number regime, while For moderate and large Peclet number, ARKC is notably better. On the other hand, the cost of ARKC is close to that of PIROCK for moderate P_e with small advantage for the latter when using large tolerance 10^{-2} . However, ARKC becomes cheaper and more accurate at the same time for large values of P_e (in our experiment for $P_e = 10$ and 12). For small tolerance 10^{-5} ARKC outperforms PIROCK in all Peclet number regimes. This is expected because the damping (and so the vertical with of the stability region) is fixed for PIROCK (when $F_A \neq 0$, PIROCK has $\delta_s = 0.43s^2$ and $\alpha_s \simeq 0.53s + 0.5$). The adaptivity of ARKC allows to continuously increase the vertical width of its stability region which lets it to be more flexible with respect to the change in P_e. One should not forget that the explicit availability

Method	a	*Steps*	F_D evals	F_A evals	*s_max*	L_∞ error at $t = 1/2$
PRKC	0.1	14|71	862|1860	56|284	126|106	4.8×10^{-4}|3×10^{-7}
PIROCK	0.1	13|237	789|3654	39|711	150|104	1.8×10^{-3}|7.5×10^{-7}
IMPRKC	0.1	15|83	953|2150	953|2150	139|118	3.4×10^{-4}|3.3×10^{-7}
ARKC	0.1	14|79	886|2098	42|237	145|97	4.3×10^{-4}|3.3×10^{-7}
PRKC	0.5	27|77	1341|2067	108|308	57|57	8.7×10^{-9}|3.8×10^{-9}
PIROCK	0.5	13|237	789|3648	39|711	150|104	1.8×10^{-3}|7.3×10^{-7}
IMPRKC	0.5	15|83	953|2152	953|2152	112|118	3.5×10^{-4}|2×10^{-7}
ARKC	0.5	13|79	909|2132	39|237	142|117	2.5×10^{-4}|2.2×10^{-7}
PRKC	1	47|89	1803|2342	188|356	40|40	9×10^{-10}|6.1×10^{-10}
PIROCK	1	13|237	849|3648	39|711	200|104	1.7×10^{-4}|6.6×10^{-7}
IMPRKC	1	15|84	949|2147	949|2147	138|121	3.7×10^{-4}|4.5×10^{-7}
ARKC	1	11|74	896|2104	33|222	194|153	2×10^{-4}|3.6×10^{-7}
PRKC	2	90|120	2575|2893	360|480	29|29	1.2×10^{-10}|1×10^{-10}
PIROCK	2	13|237	934|3720	39|711	200|150	$2 \times 10^{-6}2.5 \times 10^{-8}$
IMPRKC	2	15|86	942|2174	942|2174	141|108	4.5×10^{-4}|2.8×10^{-7}
ARKC	2	10|56	995|2267	30|168	228|172	4.8×10^{-5}|1.8×10^{-7}
PRKC	5	222|229	3980|4049	888|916	18|18	4.1×10^{-11}|4×10^{-11}
PIROCK	5	14|238	1043|3826	42|714	182|150	$1.8 \times 10^{-5}1.7 \times 10^{-7}$
IMPRKC	5	16|103	966|2350	966|2350	148|111	5.5×10^{-4}|1.6×10^{-7}
ARKC	5	12|59	1272|2764	36|177	237|184	1.9×10^{-6}|2.9×10^{-8}
PRKC	10	442|453	5738|5818	1768|1812	13|13	6.2×10^{-11}|6×10^{-11}
PIROCK	10	24|246	1574|4086	72|738	125|125	1.3×10^{-3}|1.2×10^{-5}
IMPRKC	10	23|149	1227|2819	1227|2819	106|111	9×10^{-4}|2.1×10^{-6}
ARKC	10	15|84	1359|3207	45|252	234|160	5.4×10^{-6}|7.3×10^{-8}
PRKC	12	530|543	6355|6436	2120|2172	12|12	6.4×10^{-11}|6.2×10^{-11}
PIROCK	12	28|255	1750|4306	84|765	104|104	1.1×10^{-2}|4.3×10^{-6}
IMPRKC	12	27|170	1353|3036	1353|3036	92|92	$2.1 \times 10^{--3}$|3.1×10^{-6}
ARKC	12	18|104	1557|3593	54|312	196|150	3.5×10^{-5}|4.3×10^{-7}

Table 2: Comparison of ARKC, PRKC, and PIROCK for the linear advection–diffusion problem (22). The numbers on the left correspond to $tol = 10^{-2}$, while those on the right correspond to $tol = 10^{-5}$.

of ARKC coefficients helps a lot in making the scheme adaptive with respect to the change in P_e, the feature that is missing in PIROCK.

In the small Peclet number regime, IMPRKC performs very similar to standard RKC method, which is expected, and thus it is outperformed by ARKC and the other two schemes. For large Peclet numbers, IMPRKC outperforms PRKC and its cost is close to PIROCK, while ARKC performs much better due to the very small number of evaluations of the advection term compared to that of IMPRKC.

The smaller number of steps needed in ARKC compared to the three other methods, leads to a significantly lower number of F_A evaluations no matter how big is the number of stages.

5.2 Burgers equation with a nonlinear reaction term

As an example of a PDE with variable Peclet number, we consider the following Burgers equation with a nonlinear reaction term and periodic boundary conditions

$$\begin{aligned} \partial_t u + 10u\partial_x u &= \partial_x^2 u + \sin(u^2), \quad (x,t) \in [0,1] \times [0,1/2], \\ u(x,0) &= 1 + \sin(2\pi x), \\ u(0,t) &= u(1,t) \quad \forall\, t \in [0,1/2]. \end{aligned} \tag{23}$$

We discretize the above equation in space using second order central differences with $\Delta x = 10^{-2}$, into $N+1$ grid points with $N = 100$. We calculate a reference solution using the Radau IIA method of order 5 [16].

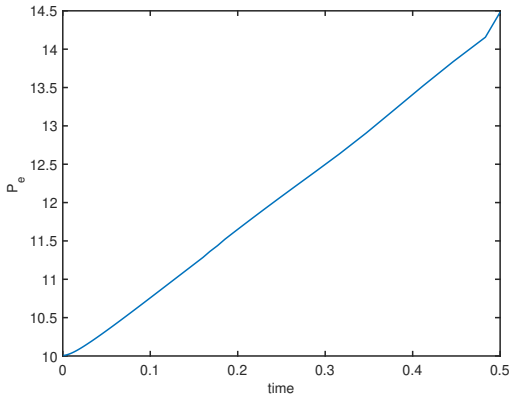


Figure 5: Change in Peclet number with respect to time in problem (23).

In Figure 5, we plot the Peclet number of equation (23) as a function of time. we see that it is variable and of quite large magnitude.

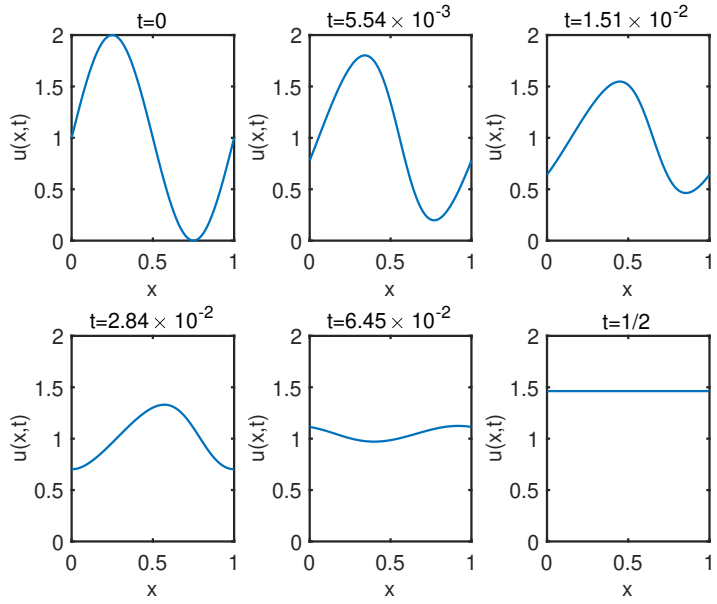
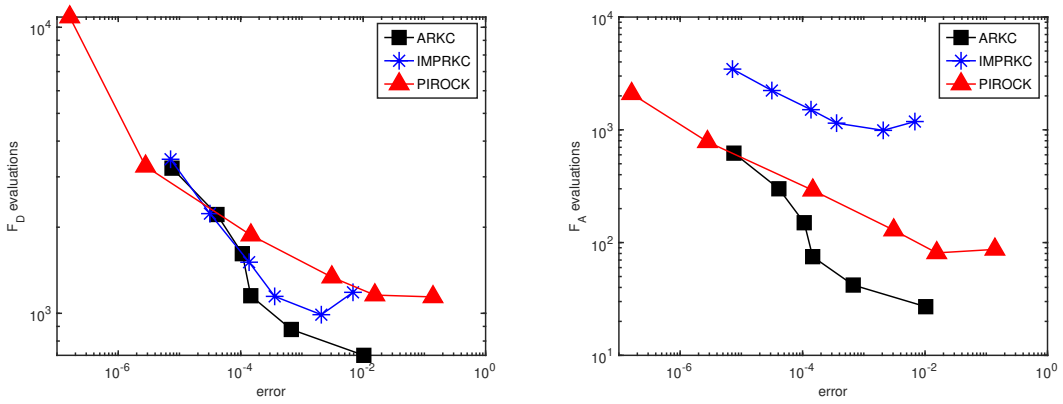


Figure 6: Solution of equation (23) obtained using ARKC method (16) at different time moments.

Figure 6 shows the solution $u(x,t)$ of equation (23) obtained using ARKC method (16) at different time moments.



(a) F_D evaluations (b) F_A evaluations

Figure 7: Comparison between ARKC, IMPRKC, and PIROCK in terms of cost for problem (23).

In Figure 7, we compare the number of functions evaluations needed to obtain a given accuracy for the solution of the Burgers equation (23) using 3 different numerical methods: ARKC, IMPRKC, and PIROCK. The results are obtained for $tol = 10^{-r}$, $r = 1, \ldots, 6$. The advantage of our scheme ARKC (16) is very clear. For IMPRKC, the number of F_D evaluations is reasonable, while that of F_A evaluations is very large compared to ARKC and PIROCK. For PIROCK, the fixed damping increases the number of steps and of functions evaluations with respect to ARKC. The flexibility of ARKC gives it remarkable advantage over the other schemes. We can also see that even when the number of F_D evaluations for ARKC is close to that of the other two schemes, the number of evaluations of F_A containing the nonlinear advection and reaction terms stays much lower, that is because of the low number of time steps needed.

6 Conclusion

In this paper, we have constructed a fully adaptive second order explicit stabilized Runge–Kutta–Chebyshev time integrator for advection–diffusion–reaction PDEs, called ARKC. The new scheme is implemented using an algorithm that is able to adaptively choose the step size, the number of stages and the damping parameter of the method according to the Peclet number. The new scheme is shown to outperform existing methods in the literature for the same type of problems. This high performance is a result of the full adaptivity of Algorithm 4.4, in particular, the adaptive damping that allows significant control of the form of the stability region as a function of the Peclet number.

Acknowledgements The author is grateful to Gilles Vilmart for helpful discussions and comments.

References

[1] A. Abdulle. *Chebyshev methods based on orthogonal polynomials*. PhD Thesis, University of Geneva, Department of Mathematics. University of Geneva, 2001.

[2] A. Abdulle. Fourth order Chebyshev methods with recurrence relation. *SIAM J. Sci. Comput.*, 23(6):2041–2054, 2002.

[3] A. Abdulle. ROCK2 and ROCK4: software for stiff differential equations (discretized parabolic problems). *Codes available under* `http://anmc.epfl.ch/`, 2002.

[4] A. Abdulle. *Explicit Stabilized Runge–Kutta Methods*, pages 460–468. Encyclopedia of Applied and Computational Mathematics, Springer Berlin Heidelberg, 2015.

[5] A. Abdulle, I. Almuslimani, and G. Vilmart. Optimal explicit stabilized integrator of weak order 1 for stiff and ergodic stochastic differential equations. *SIAM/ASA J. Uncertain. Quantif.*, 6(2):937–964, 2018.

[6] A. Abdulle and G. R. de Souza. Explicit stabilized multirate method for stiff stochastic differential equations. *arXiv:2010.15193*, 2020.

[7] A. Abdulle and T. Li. S-ROCK methods for stiff Ito SDEs. *Commun. Math. Sci.*, 6(4):845–868, 2008.

[8] A. Abdulle and A. Medovikov. Second order chebyshev methods based on orthogonal polynomials. *Numer. Math.*, 90(1):1–18, 2001.

[9] A. Abdulle and G. Vilmart. PIROCK: a swiss-knife partitioned implicit-explicit orthogonal Runge-Kutta Chebyshev integrator for stiff diffusion-advection-reaction problems with or without noise. *J. Comput. Phys.*, 242:869–888, 2013.

[10] A. Abdulle, G. Vilmart, and K. C. Zygalakis. Weak second order explicit stabilized methods for stiff stochastic differential equations. *SIAM J. Sci. Comput.*, 35(4):A1792–A1814, 2013.

[11] I. Almuslimani. *Explicit Stabilized Methods for Stiff Stochastic Differential Equations and Stiff Optimal Control Problems*. University of Geneva. PhD thesis, 2020.

[12] I. Almuslimani and G. Vilmart. Explicit stabilized integrators for stiff optimal control problems. *SIAM J. Sci. Comput.*, 43(2):A721–A743, 2021.

[13] M. Bakker. Analytical aspects of a minimax problem. 1971. Technical Note TN 62 (in Dutch), Mathematical centre, Amsterdam.

[14] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. Structure-preserving algorithms for ordinary differential equations.

[15] E. Hairer, S. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems*, volume 8. Springer Verlag Series in Comput. Math., Berlin, 1993.

[16] E. Hairer and G. Wanner. *Solving ordinary differential equations II. Stiff and differential-algebraic problems*. Springer-Verlag, Berlin and Heidelberg, 1996.

[17] W. Hundsdorfer and J. Verwer. *Numerical solution of time-dependent advection-diffusion-reaction equations*, volume 33 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2003.

[18] R. Jeltsch and M. Torrilhon. Flexible stability domains for explicit Runge-Kutta methods. In *Some topics in industrial and applied mathematics*, volume 8 of *Ser. Contemp. Appl. Math. CAM*, pages 152–180. Higher Ed. Press, Beijing, 2007.

[19] D. I. Ketcheson and A. J. Ahmadia. Optimal stability polynomials for numerical integration of initial value problems. *Commun. Appl. Math. Comput. Sci.*, 7(2):247–271, 2012.

[20] B. Sommeijer, L. Shampine, and J. Verwer. RKC: an explicit solver for parabolic PDEs. *J. Comput. Appl. Math.*, 88:316–326, 1998.

[21] B. P. Sommeijer and J. G. Verwer. *A performance evaluation of a class of Runge-Kutta-Chebyshev methods for solving semidiscrete parabolic differential equations.* Afdeling Numerieke Wiskunde [Department of Numerical Mathematics], 91. Mathematisch Centrum, Amsterdam, 1980.

[22] X. Tang and A. Xiao. Improved runge–kutta–chebyshev methods. *Mathematics and Computers in Simulation*, 174:59–75, 2020.

[23] M. Torrilhon and R. Jeltsch. Essentially optimal explicit Runge-Kutta methods with application to hyperbolic-parabolic equations. *Numer. Math.*, 106(2):303–334, 2007.

[24] P. Van der Houwen and B. Sommeijer. On the internal stage runge-kutta methods for largem-values. *Z Angew Math Mech*, 60:479–485, 1980.

[25] P. J. van der Houwen and B. P. Sommeijer. On the internal stability of explicit, m-stage Runge-Kutta methods for large m-values. *Z. Angew. Math. Mech.*, 60(10):479–485, 1980.

[26] J. Verwer, W. Hundsdorfer, and B. Sommeijer. Convergence properties of the runge-kutta-chebyshev method. *Numer. Math.*, 57:157–178, 1990.

[27] J. G. Verwer and B. P. Sommeijer. An implicit-explicit Runge-Kutta-Chebyshev scheme for diffusion-reaction equations. *SIAM J. Sci. Comput.*, 25(5):1824–1835, 2004.

[28] J. G. Verwer, B. P. Sommeijer, and W. Hundsdorfer. RKC time-stepping for advection-diffusion-reaction problems. *J. Comput. Phys.*, 201(1):61–79, 2004.

[29] C. J. Zbinden. Partitioned Runge-Kutta-Chebyshev methods for diffusion-advection-reaction problems. *SIAM J. Sci. Comput.*, 33(4):1707–1725, 2011.

A Damping and number of stages for some choices of Peclet number

s	< 11	< 21	< 31	< 41	< 51	< 61	< 71	< 81	< 91	< 101
η	0.15	0.6	1	1.4	1.7	2.1	2.4	2.7	3	3.3
s	< 121	< 141	< 161	< 181	< 201	< 251	< 301	< 401	< 501	
η	3.7	4.1	4.5	4.9	5.3	6	6.6	7.7	8.8	

Table 3: $1/2 < P'_e \leq 1$, $1/4 < \rho_A/\sqrt{\rho_D} \leq 1/2$.

s	< 11	< 21	< 31	< 41	< 51	< 61	< 71	< 81
η	0.7	1.5	2.3	2.9	3.5	4	4.5	4.9
s	< 91	< 101	< 141	< 181	< 251	< 301	< 401	< 501
η	5.2	5.5	6.7	7.7	8.8	9.8	11	12

Table 4: $1 < P'_e \leq 3/2$, $1/2 < \rho_A/\sqrt{\rho_D} \leq 3/4$.

s	< 11	< 21	< 31	51	< 71	< 111	< 151	< 311	501
η	1	2.5	3.5	4.8	6	7.8	9	12.5	15

Table 5: $3/2 < P'_e \leq 2$, $3/4 < \rho_A/\sqrt{\rho_D} \leq 1$.

s	< 11	< 21	< 31	51	< 71	< 111	< 151	< 311	501
η	2	3.8	5	6.8	8	10.4	12	16	19

Table 6: $2 < P'_e \leq 2\sqrt{2}$, $1 < \rho_A/\sqrt{\rho_D} \leq \sqrt{2}$.

s	< 11	< 31	< 71	< 151	< 311	< 501
η	4	9	13.5	18	23	27

Table 7: $P'_e > 2\sqrt{2}$, $\rho_A/\sqrt{\rho_D} > \sqrt{2}$.

One can get more tables and increase the adaptivity of the algorithm with respect to damping. This will for sure increase the performance of the method. However, the method performs already very well with the tables we provided.