

Exploring Exponential Runge-Kutta Methods: A Survey

Alessia Andò^{1,*}, Nicolò Cangini² and Mattia Sensi³

¹ *Department of Mathematics, Computer Science and Physics, University of Udine, 33100 Udine, Italy.*

² *Department of Mathematics, Politecnico di Milano, Campus Leonardo, 20133 Milan, Italy.*

³ *Department of Mathematics, University of Trento, 38123 Povo (Trento), Italy.*

Received 12 February 2026; Accepted 3 April 2026

Abstract. In this survey, we provide an in-depth investigation of exponential Runge-Kutta methods for the numerical integration of initial-value problems. These methods offer a valuable synthesis between classical Runge-Kutta methods, introduced more than a century ago, and exponential integrators, which date back to the 1960s. This manuscript presents both a historical analysis of the development of these methods up to the present day and several examples aimed at making the topic accessible to a broad audience.

AMS subject classifications: 65L03, 65L04, 65L05, 65L20, 65M12

Key words: Exponential integrators, exponential Runge-Kutta methods, exponential time-differencing, integrating factor, semilinear equations.

1 Introduction

Numerical integration of systems of ordinary differential equations (ODEs) is one of the fundamental pillars of numerical analysis. In particular, many models

*Corresponding author. *Email addresses:* alessia.ando@uniud.it (A. Andò), nicolo.cangiotti@polimi.it (N. Cangini), mattia.sensi@unitn.it (M. Sensi)

stemming from real-world scenarios, with applications ranging from quantum mechanics to fluid dynamics, naturally lead to systems of the form

$$u'(t) = Au(t) + g(t, u(t)), \quad u(0) = u_0, \quad (1.1)$$

where $u(t) \in \mathbb{R}^d$, $t \geq 0$, A is a linear operator (typically a matrix) describing the (typically stiff) part of the problem, while g represents the (typically non-stiff) nonlinear component. Remarkably, despite its fundamental importance, there is no univocal and universally accepted definition of stiffness. Some mathematicians characterize it through the presence of components of the solution that evolve on very different time scales, others define it in terms of the ratio between the eigenvalues of the Jacobian matrix of the system, while still others relate it to the practical difficulties that arise when using explicit numerical methods. This multiplicity of definitions reflects the intrinsic complexity of the phenomenon and the different perspectives from which it can be analyzed. We shall provide a more exhaustive discussion about the topic in Section 2.1.

Runge-Kutta (RK) methods represent a milestone in the development of numerical methods for ODEs. Initially introduced by Carl Runge in 1895 [88] and later refined by Martin Wilhelm Kutta in 1901 [60], these methods revolutionized the approach to the numerical solution of ODEs. Their classical formulation, based on a linear combination of function evaluations at appropriately chosen intermediate points, offers a remarkable balance between accuracy and simplicity of implementation. However, despite their historical success, classical explicit RK methods have shown significant limitations when applied to more complex or stiff problems.¹ The need to use extremely small time steps to ensure stability in the already mentioned stiff problems, with the consequent increase in computational cost, has pushed the scientific community towards the development of more sophisticated variants. Furthermore, the inability of classical RK methods to naturally preserve some important physical properties of the systems (such as invariants and energies) has motivated the search for more advanced geometric and structural methods.²

¹It is important here to make a necessary distinction between explicit and implicit RK methods. Explicit methods compute each stage using only previously computed ones, but suffer from strong stability limitations, especially in the presence of stiff problems. On the contrary, implicit methods guarantee better stability and are particularly suitable for tackling stiff problems. The systematic introduction of implicit methods dates back to the pioneering works of John Butcher [12], who provided a solid theoretical basis for their development and application. The main drawback of implicit methods is their high computational cost, since they require the knowledge of the solution for nonlinear systems of equations at each stage.

Thus, the presence of stiff terms represents a fascinating challenge for traditional numerical methods. Among the various approaches developed to address this class of problems, exponential RK (ExpRK) methods have assumed a particularly important role. As pointed out by Hochbruck *et al.* [47]:

ExpRK methods represent a breakthrough in handling stiff semilinear problems by combining the exact integration of the linear part with carefully constructed stages for the nonlinear terms, leading to exceptional stability properties without severe step size restrictions.

We emphasize that the capability of ExpRK methods to deal with stiffness makes them particularly suitable for various classes of partial differential equations (PDEs) and delay differential equations (DDEs), as will be investigated in the following sections. The fundamental idea of these methods is based on the variation-of-constant formula (VCF), which allows the solution to be rewritten using the matrix exponential e^A . Thus, the solution to the problem (1.1) can be expressed as follows:

$$u(t_0+h) = e^{hA}u(t_0) + \int_0^h e^{(h-\tau)A}g(t_0+\tau, u(t_0+\tau))d\tau. \quad (1.2)$$

The efficient implementation of these methods requires two fundamental ingredients: the computation of e^A and the approximation of the integral containing the nonlinear term. The matrix exponential can be characterized and computed in different ways: through the classical power series, through spectral decomposition when possible, or through integral representations in the complex plane (see, for instance, [43]). The numerical evaluation of the matrix exponential is an active area of research itself, with techniques ranging from scaling and squaring methods to approaches based on Krylov subspaces, pioneered by Saad [89]. As for the integral term, its approximation requires a suitable quadrature, i.e., a suitable combination of evaluations of the function g . The general ExpRK method can thus be expressed in the form

²We remark that in the mid-1990s, two important reviews regarding RK methods were published. The first one, dated 1994, is due to Enright *et al.* [32] and it focuses on algorithms related to explicit methods, while the second one was written by Butcher in 1996 [13] and with a more historical approach. Butcher concentrates on the evolution of the methods from their origins, with particular attention to implicit methods. Another interesting survey about Pseudo-RK methods is due to Costabile [19] and it was published in 2003. We remark that in none of these works is there any reference to exponential methods.

$$\begin{cases} u_{n+1} = e^{h_n A} u_n + h_n \sum_{i=1}^{\nu} b_i(h_n A) g(t_{n,i}, U_n^i), \\ U_n^i = e^{c_i h_n A} u_n + h_n \sum_{j=1}^{\nu} a_{ij}(h_n A) g(t_{n,j}, U_n^j), \quad i=1, \dots, \nu, \end{cases} \quad (1.3)$$

where ν is the number of stages, the abscissae c_i are non-negative, $t_{n+1,i} = t_n + c_i h_n$ and the weights $b_i(h_n A)$ and the coefficients $a_{ij}(h_n A)$ are $(d \times d)$ -matrices. In the following, the expression ExpRK method/scheme will refer to methods of the form (1.3) such that the coefficients a_{ij} and b_i only depend on A but not on g (although, when tracing their historical path from Section 3 on, we also refer to different, but related methods).³ Two main classes fit into this definition: the integrating factor (IF) methods and the exponential time differencing (ETD) methods. The former are characterized by coefficients of the form

$$b_i(h_n A) = \bar{b}_i e^{(1-c_i)h_n A}, \quad a_{ij}(h_n A) = \bar{a}_{ij} e^{(c_i-c_j)h_n A}, \quad i, j = 1, \dots, \nu,$$

where \bar{b}_i, \bar{a}_{ij} are constants. In the latter, instead, the coefficients are defined through the so-called φ -functions. These functions, which naturally generalize the matrix exponential, emerge from the analysis of the integral term and play a central role in the construction and analysis of these methods. We highlight that φ -functions can be defined in several equivalent ways. A particularly useful definition is through the following integral:

$$\varphi_k(z) := \frac{1}{(k-1)!} \int_0^1 e^{z(1-\tau)} \tau^{k-1} d\tau.$$

This definition shows that φ -functions are essentially weighted averages of the exponential over an interval, with polynomial weights. In the context of ExpRK methods, these functions emerge when we try to solve semilinear systems of the form (1.1), where the linear part is treated exactly (via the matrix exponential), while handling the nonlinear part with a RK-type approach. When building an ETD scheme, one derives formulas involving combinations of these φ -functions applied to the operator A multiplied by the time step h . For instance, in a one-stage scheme with constant step size, we might have

$$u_{n+1} = e^{hA} u_n + h\varphi_1(hA)g(u_n), \quad (1.4)$$

³It is worth noting that if one allows the coefficients a_{ij} and b_i to depend on the nonlinear term g , its Jacobian, or an approximation thereof, additional families can be cast into the same general stage structure. The most notable examples are exponential Rosenbrock methods [52], exponential W-methods [47] and exponential propagation iterative methods of RK type [92].

where $u_n \approx u(t_n)$ for a given mesh of time points $\{t_n\}_n$. Here, $\varphi_1(hA)$ acts as a sort of modified propagator for the nonlinear term. Note that using this approach when $A = 0$ we recover the standard RK methods exactly, while when $g = 0$ we get the exact solution for the linear part. In multi-stage schemes, higher-order φ -functions also come into play. These handle the correction terms needed to achieve a higher order of convergence, as we are going to explain in the following sections.

This article pursues a dual objective. Firstly, it provides a comprehensive overview of ExpRK methods, tracing their development from the earliest formulations to some of the most recent theoretical advances. The goal is to offer a clear and coherent account of how these methods have evolved, why they remain of current interest, and what their key characteristics are, while keeping the presentation concise and focused on the essential points. Secondly, the article is intended as a resource for readers encountering these methods for the first time. Starting from the fundamental theoretical principles, the exposition gradually progresses to more advanced topics, accompanied by illustrative examples that clarify the main features and practical significance of the methods. By combining these two perspectives, the survey aims to be both a thorough reference for specialists and an accessible introduction for newcomers, providing a presentation that is at once comprehensive, structured, and approachable.

The manuscript is organized as follows. Section 2 focuses on the key theoretical aspects of ExpRK methods. Section 3 presents a historical overview of the fundamental theoretical results, starting from the early works up to the most recent review of exponential methods by Hochbruck and Ostermann [50]. In Section 4, we recapitulate the most recent developments of ExpRK methods, from their applications to high stiff order and delay differential equations up to their usage in several modeling frameworks. Finally, Section 5 is devoted to the numerical simulations of three test cases, two of which of low dimension, that illustrate the discussed theoretical properties.

2 Basic definitions and construction of ExpRK methods

2.1 Stiffness in a nutshell

Despite the word stiff having been used for several decades, defining stiffness in precise mathematical terms seems to be an impossible task. To the best of the authors' knowledge, the term first appeared in [21], where the phenomenon

of stiffness was shown by means of a scalar example. One could try to define stiffness in terms of the inherent characteristics of the problem at hand, without explicitly referring to its numerical approximation. From this perspective, two large (not mutually exclusive) groups of problems may be classified as stiff:

- Problems whose Jacobian matrices have eigenvalues λ with very large negative real part.
- Problems which involve quantities evolving at very different time scales.

Naturally, the above are not rigorous definitions, nor are they truly independent of the numerical methods considered. In particular, the word large in the first group is somehow related to the step size h : it would be more accurate to say that a problem is stiff if $|\lambda h|$ is large, or, in other words, if it requires “unreasonable restrictions on h ” [81]. As for the second group, observe that the fast and slow components are usually associated to the large and small (in modulus) eigenvalues of the Jacobian, respectively [23] and, indeed, the ratio between the largest and the smallest real parts - the stiffness ratio - was once considered a possible measure of stiffness [61].

While both characterizations above concern properties which are typical of stiff problems, it has become clear over time that neither represents an exhaustive explanation of the complex phenomenon. More pragmatically, it is now deemed more appropriate to define stiffness in terms of the numerical issues arising from the phenomenon. In other words [40]:

Stiff equations are problems for which explicit methods do not work.

Or, at the very least, problems for which explicit methods perform worse than implicit ones. For an insightful discussion on the topic of stiffness, we refer the interested readers to [62]. It is shown therein, by means of numerical experiments, that the stiffness ratio cannot be interpreted as a measure of stiffness and the classic pseudodefinitions are neither sufficient nor necessary conditions for stiffness. Moreover, the pragmatic definition is also rephrased so as to emphasize that stiffness varies with the specific interval over which the problem is integrated:

If a numerical method with a finite region of absolute stability, applied to a system with any initial conditions, is forced to use in a certain interval of integration a step-length which is excessively small in relation to the smoothness of the exact solution in that interval, then the system is said to be stiff in that interval.

2.2 Constructing ExpRK methods

The approximation of the exponential of a matrix represents a crucial aspect in the implementation of exponential methods, due to its high computational costs.

At the end of the 1980s, Gallopoulos and Saad [35, 36] were among the first to emphasize that in the context of numerical integration one does not need to actually compute exponentials of matrices, but rather only products of the form $e^{-A}v$, where v is a vector. This led them to explore the possibility to apply Krylov methods in this context. The approach consists in approximating $e^{-A}v$ with a member of the Krylov subspace

$$K_m := \text{span}\{v, Av, \dots, A^{m-1}v\}.$$

The dimension m of the space is typically much smaller than the dimension of the matrix. Given the matrix $V_m = [v_1 v_2 \cdots v_m]$ with $v_1 = v$, whose columns form an orthogonal basis for K_m , and the orthogonal projection H_m (i.e. the Hessenberg matrix) of A on K_m , the resulting approximation reads

$$e^{-A}v \approx \|v\|_2 V_m e^{-H_m} \mathbf{1}_m, \quad \mathbf{1}_m = [1, 1, \dots, 1]^T \in \mathbb{R}^m. \quad (2.1)$$

Thus, the problem reduces to computing the exponential of a much smaller matrix than the original one. A different approach proposed in [35] consists in approximating the exponential with a rational function. As highlighted in [36], the two approaches can be combined, applying the rational approximation to e^{-H_m} while making use of (2.1). Eventually, Saad [89] further improved the accuracy obtained with the Krylov approach by rewriting $e^{-A}v = v - A\varphi_1(-A)v$ and computing a Krylov approximation for $\varphi_1(-A)$ instead of one for e^{-A} .

Later, a series of works focused on improving existing methods to approximate the full exponential (or related functions) of a matrix, rather than its action on specific vectors. The first is attributed to Beylkin *et al.* [10], who observed that the problem with Taylor truncation methods (so, also with Padé rational approximations, whose numerators and denominators are truncated Taylor series) arise when the matrix A possesses large singular values. Thus, they proposed the scaling and squaring algorithm to evaluate the exponentials, which consists in exploiting the equality $e^A = (e^{A/2^s})^{2^s}$, where s is chosen such that the largest singular value of $e^{A/2^s}$ is less than one. The authors also made use of this approach to compute all φ_k functions by means of their recursive formula

$$\varphi_0(z) = e^z, \quad \varphi_{k+1}(z) = \frac{\varphi_k(z) - 1/k!}{z}, \quad z \neq 0, \quad \varphi_{k+1}(0) = \frac{1}{(k+1)!}, \quad k \geq 0. \quad (2.2)$$

The scaling and squaring method is probably the most popular for computing matrix exponentials and it is used in MATLAB's command `expm` in combination with a Padé approximation using the same degree for the numerator and denominator [42].

In 2003, Davies and Higham [26] proposed a different approach, based on decomposing the matrix A as $A = S^{-1}BS$ such that e^B (or $\varphi_k(B)$) is much easier to compute than e^A , and S has a reasonably low condition number. In particular, their idea was to use a Schur decomposition (such that S is unitary and B is upper triangular) combined with a recurrence algorithm of Parlett [85] which exploits the fact that functions of block upper triangular matrices inherit their block structure. If A is symmetric, the approach introduced by Lu [65] for computing $\varphi_1(A)$ turns out to be more efficient. The method is again based on a matrix decomposition, obtained with an orthogonal S and a tridiagonal B . The idea in this case is to approximate e^B using a Chebyshev rational approximation for $e^{B-\lambda I}$, where λ is the largest eigenvalue of B . The algorithm has then been generalized to compute all φ -functions [79].

Another class of methods that can be employed to approximate the φ -functions is that of polynomial interpolation, such as interpolation on Leja points [16].

A straightforward application of (2.2) to compute the φ -functions suffers from cancellation errors for small z . The idea implemented by Cox and Matthews [20] was to use a piecewise strategy, namely to compute the φ -functions directly with (2.2) for large z , and with truncated Taylor series for small z . There is a certain region, however, where neither approach is accurate enough. Kassam and Trefethen [57] propose a method with the idea of bridging the gap between small and large z , based on the computation of the contour integral

$$\varphi_k(z) = \frac{1}{2\pi i} \int_{\Gamma} \frac{\varphi_k(t)}{t-z} dt,$$

where the contour Γ encloses z and is separated from 0. They suggest using the trapezoidal rule to efficiently compute the contour integral. The computational cost is particularly cheap if the coefficient matrix has a sparse block structure.

In 2009, Al-Mohy and Higham [1] proposed a heuristic improvement on the standard scaling and squaring algorithm combined with Padé approximation. The goal was to solve the problem of overscaling, namely the fact that for large $\|A\|$ a value of s is chosen which is too large, causing loss of accuracy in floating-point arithmetic. The modification involves exploiting the values $\|A^k\|^{1/k}$ and the improvements are more evident for upper triangular matrices.

The (currently) most popular method to compute the action of matrix exponentials and φ -functions on vectors was proposed by the same authors in 2011 [2].

In fact, the purpose of that paper is twofold. The authors observe first that the problem of computing sums of the form $\sum_{k=0}^p \varphi_k(A)u_k$ functions defining an exponential method can be recasted as a single matrix exponential $e^{\tilde{A}}$ where \tilde{A} is obtained by suitably augmenting A with p rows and columns. Next, they propose their algorithm for computing products of matrix exponential times matrices $e^A B$ for $B \in \mathbb{R}^{n \times m}$ with $m \ll n$, as well as $e^{tA} B$ where t is an array of equally-spaced time points. It is based on the scaling part of the scaling and squaring, plus a truncated Taylor series

$$e^A B = (e^{s^{-1}A})^s B \approx B_s, \quad B_{i+1} = \left(\sum_{j=0}^m \frac{(s^{-1}A)^j}{j!} \right) B_i, \quad B_0 = B.$$

Optimal values for s and m are obtained through a backward error analysis. The algorithm incorporates the ideas previously proposed in [1] in order to minimize overscaling and turns out to be generally superior to existing ones in terms of accuracy, stability and predictability of cost, since the only convergence test involved is a test for early termination built into the evaluation of the truncated Taylor series.⁴

Toy model

Having established the theoretical framework, we now introduce a toy model aimed at reinforcing the pedagogical purpose of this manuscript. This example serves to illustrate the key aspects of the previously described numerical method, offering a clearer understanding of its implementation and highlighting its most relevant features. So, let us consider a classic one-dimensional stiff toy-model

$$u'(t) = \lambda u(t) + N(u(t)), \quad u(0) = u_0, \quad (2.3)$$

where λ is a negative real number such that $|\lambda|$ is very large (causing the stiffness of the problem) and $N(u)$ is a milder nonlinearity. Specifically, for this example we choose $\lambda = -1000$ and $N(u) = 2u/(1+u^2)$. This problem combines a strongly dissipative term ($\lambda u(t)$) with a bounded nonlinearity.

We apply the exponential time-differencing Euler method (ETD Euler) (1.4), defined by a single internal stage

$$\begin{aligned} U_1 &= u_n, \\ u_{n+1} &= e^{h\lambda} u_n + h\varphi_1(h\lambda)N(U_1). \end{aligned}$$

If we choose, for instance $h = 0.01$, then:

⁴We remark that the algorithm is implemented in the MATLAB function `expmv`.

1. We compute $e^{h\lambda} = e^{-10} \approx 4.54 \times 10^{-5}$.
2. We compute $\varphi_1(h\lambda) = (e^{h\lambda} - 1)/h\lambda \approx -1/\lambda = 1/10$.

For $u_0 = 1$, the first step is given by

$$\begin{aligned} U_1 &= 1, \\ N(U_1) &= 2 \times 1 / (1 + 1^2) = 1, \\ u_1 &= (4.54 \times 10^{-5}) \times 1 + 0.01\varphi_1(-10) \times 1 \approx 10^{-3}. \end{aligned}$$

Now, we can compare the performance of ETD Euler and the classical order 4 RK method (RK4), whose application to (2.3) gives

$$\begin{aligned} u_1 &= u_0 = 1, \\ K_1 &= h(\lambda u_1 + N(u_1)) = h(\lambda + 1) = -9.99, \\ u_2 &= U_1 + K_1/2 = -3.995, \\ K_2 &= h(\lambda U_2 + N(U_2)) = 39.9411, \\ U_3 &= U_1 + K_2/2 = 20.9705, \\ K_3 &= h(\lambda U_3 + N(U_3)) = -209.7046, \\ U_4 &= U_1 + K_3 = -208.7046, \\ K_4 &= h(\lambda U_4 + N(U_4)) \approx 2.09 \times 10^3, \\ u_1 &= u_0 + K_1/6 + K_2/3 + K_3/3 + K_4/6 \approx 290.59. \end{aligned}$$

This extremely large value provides further evidence that the equation exhibits stiff behavior and is highly sensitive on initial conditions, meaning that a smaller step size might be necessary for numerical stability. Indeed, repeating the computation with $h = 0.001$, we obtain $u_1 \approx 0.3755$.

We can highlight the following differences between the two methods: ETD Euler requires computation of $e^{h\lambda}$ and $\varphi_1(h\lambda)$, while RK4 uses only basic arithmetic operations. Moreover, as for the total function evaluations, ETD Euler requires approximately 20 evaluations of N and RK4 requires approximately 400 evaluations of N (see Table 1 for a full comparison). The advantages of ETD Euler are now clear: it remains stable with larger time steps, maintaining precision comparable to RK4. Moreover, it requires significantly fewer function evaluations.

Table 1: Computational performance comparison.

| Method | h_{\max} | Steps | Evaluations/Step | Total Evaluations | Max Error |
|-----------|------------|-------|------------------|-------------------|----------------------|
| ETD Euler | 0.01 | 10 | 2 | 20 | 3.2×10^{-8} |
| RK4 | 0.001 | 100 | 4 | 400 | 2.8×10^{-8} |

Let us analyze the stability of the methods ETD Euler and RK4 more rigorously. For the linear problem $u'(t) = \lambda u(t)$, both schemes considered yield a recurrence of the form $u_{n+1} = R(\lambda h)u_n$, and the region

$$S = \{z \in \mathbb{C} : |R(z)| < 1\}$$

is the stability domain. For example, for ETD Euler,

$$R(z) = e^z,$$

while for RK4 it can be verified that

$$R(z) = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \frac{z^4}{4!}.$$

Observe that the origin is a stable steady state of a numerical method for the linear equation whenever $h\lambda$ lies in the corresponding stability region. Thus, in general, stability of the origin depends on the choice of h . A desirable property of a numerical method is *A*-stability: a scheme is *A*-stable [22] if the entire negative complex half plane is contained in its stability domain, such that the stability of the origin is independent of the chosen step size.

We immediately observe that $|e^z| < 1$ for every z with $\Re(z) < 0$. This implies that ETD Euler is always stable, the numerical solution decreases exponentially (just like the exact one) with no restriction on h . On the other hand, the stability condition for RK4 leads to $h_{\max}\lambda \approx -2.78529$ and, for our problem with $\lambda = -1000$, h must be lower than (approximately) 2.78529×10^{-3} .

2.3 Stability

At this juncture, we shall further explore the concept of stability, as it plays a crucial role in numerical methods. In order to remain focused on the main goals of this survey, we shall mainly focus on the types of stability relevant to exponential methods.

A good starting point is provided by the paper published in 2009 by Maset and Zennaro [76], in which the authors defined the concepts of unconditional stability for ExpRK methods. Their definition is based on the concepts of contractivity and asymptotic stability for semilinear systems of the form (1.1) where g is non-stiff. Specifically, given solutions u, v of (1.1) with initial values u_0, v_0 and $\delta := u - v$, (1.1) is called contractive if

$$\|\delta(t)\| \leq \|\delta_0\|, \quad t \geq 0,$$

where $\delta_0 = \delta(0)$ and $\|\cdot\|$ is any norm in \mathbb{R}^d , and it is called asymptotically stable if

$$\|\delta(t)\| \rightarrow 0, \quad t \rightarrow \infty.$$

In a sense, unconditionally contractive (resp., asymptotically stable) methods are those preserving the corresponding property. A sufficient condition for contractivity of (1.1) is

$$\mu(A) + \gamma \leq 0, \quad (2.4)$$

where the logarithmic norm of A is defined as

$$\mu(A) := \lim_{\varepsilon \rightarrow 0^+} \frac{\|I_d - \varepsilon A\| - 1}{\varepsilon} = \lim_{\varepsilon \rightarrow 0^+} \frac{\log \|e^{\varepsilon A}\|}{\varepsilon}$$

and γ is a (moderate) Lipschitz constant for g with respect to its second argument. Similarly, (2.4) with \leq replaced by $<$ is a sufficient condition for unconditional asymptotic stability. In the following, for a given ExpRK method, u_n and v_n represent the approximations of u and v at time t_n , and $\delta_n = u_n - v_n$.

Definition 2.1. *An ExpRK method is called unconditionally contractive with respect to the norm $\|\cdot\|$ if $\|\delta_1\| \leq \|\delta_0\|$ holds for all u_0, v_0 and for all step sizes h when applied to any system of the form (1.1) satisfying (2.4). It is also called unconditionally asymptotically stable if $\|\delta_n\| \rightarrow 0$ as $n \rightarrow \infty$ whenever (2.4) is satisfied with \leq replaced by $<$.*

Maset and Zennaro [76] obtained sufficient conditions for unconditional contractivity and asymptotic stability in terms of a suitable stability function. This allows them to conclude that the ETD Euler method (1.4) and the two main classes of ExpRK methods for certain values of the parameters are unconditionally contractive and asymptotically stable. In 2013, the same authors [77] sharpened the aforementioned results and determined necessary and sufficient conditions for unconditional contractivity and asymptotic stability. This implies that no other ExpRK method with 2 stages, other than those already detected in [76], is unconditionally stable. They also analyzed the main methods with 3 to 5 stages, and observed that none of them is unconditionally stable either. However, since γ is assumed to be relatively small, conditional contractivity and stability properties are also desirable.

Definition 2.2. *An ExpRK method is called conditionally contractive (resp., asymptotically stable) with respect to the norm $\|\cdot\|$ if $\|\delta_1\| \leq \|\delta_0\|$ (resp., $\|\delta_n\| \rightarrow 0$ as $n \rightarrow \infty$) holds for all u_0, v_0 and for all step sizes $h < C/\gamma$ where C is a constant dependent on the method when applied to any system of the form (1.1) satisfying (2.4) (resp. (2.4) with \leq replaced by $<$).*

Sufficient conditions for conditional properties are provided in [77]. Without any restriction on the norm, these conditions only guarantee conditional contractivity [asymptotic stability] for those methods which are already known to be unconditionally contractive [asymptotically stable]. However, more can be said if the Euclidean norm is considered. In particular, weaker restrictions on the parameters of two-stages methods are needed in order to obtain conditional properties. The higher-order methods considered do not satisfy the sufficient conditions for general problems of the form (1.1), even when considering the Euclidean norm, but many satisfy conditional stability properties when restricted to certain classes of problems, such as those where A is symmetric.

3 The historical path of ExpRK methods

In this section, we present a brief historical overview of the development and dissemination of ExpRK methods, tracing their evolution from their inception to the 2000s. We identified two major periods in which to structure our analysis. The first period begins in the late 1950s and lays the theoretical foundations that would gain traction in the following decades, up until the second half of the 1970s. The second period marks a resurgence of interest in these methods, starting in the late 1980s. Several works have already synthesized the progress made up to the early 2000s (see e.g. [80]). Therefore, we will not dwell on the details and instead refer the reader to the relevant literature. Although the 2005 works by Hochbruck and Ostermann [48, 49] represent a significant milestone in the evolution of ExpRK methods — marking the first application of these methods to semilinear parabolic problems — we have chosen to end this phase of our analysis with their most recent comprehensive review on the topic, published in 2010 [50]. This choice allows us to provide a more rigorous and detailed examination of the last 15 years of developments, which constitutes a fundamental part of this work, and to which we dedicated Section 4.

3.1 The first approaches

We devote these paragraphs to the first decades of theory and implementation on ExpRK method. In particular, the first chronological reference dates back to 1958: in [41], Hersch introduces the term distribution to indicate a numerical method which uses matrix exponentiation according to the variation of constants formula, and applies it to a semilinear ODE. Shortly thereafter, in 1960, Certaine [17] has started the analysis around the concept of stiffness, without explicitly using this

term. The class of problems analyzed is semilinear, with a diagonal linear term, and a non-stiff nonlinear part. In the same years, Pope [86] proposed a manipulation technique to write a generic ODE in a semilinear form, thus introducing the easiest exponential Rosenbrock method (without using this name), and observing unconditional stability⁵. Rosenbrock's seminal manuscript [87] appeared in the same year, introducing the main ideas that led to a new class of RK type methods with coefficients that are not constant but rather depend on the Jacobian matrix of the right-hand side.

The years spanning the late 1960s and early 1970s were particularly fruitful for the development of Exprk theory. Indeed, in 1967, Lawson [63] extended Pope's results with a generalized RK method. This is achieved by transforming an ODE of the form $u'(t) = f(t, u(t))$ into one for the variable $z(t) = e^{-tA}u(t)$ such that $\partial f / \partial u - A$ is non-stiff (i.e. the Jacobian matrix of $f - A$ has small eigenvalues), and finally applying a classical RK scheme to the transformed system. This is the basis of the integrating factor methods (IF), as opposed to exponential time differencing methods (ETD) used by Certain and Pope [17, 86], which are based on the variation of constants formula (VCF). Here, A -stability is observed, differently from classical explicit RK methods.

In 1969, Nørsett [84] built a nonlinear A -stable method modifying Adams-Bashforth's method using VCF. The method integrates exactly $u'(t) = -Pu(t) + T(t)$, and the author observes A -stability. This method is of order $q+1$, where q is number of past steps, and $q=0$ gives the Euler method (1.4), sometimes referred to as Nørsett-Euler. Concerning the analysis of stability, Guderley and Hsu [39] provided in 1972 accuracy and stability analysis of the methods proposed by Certain in 1960, considering ODEs of the form $u'(t) = -\Lambda u(t) + f(t, u)$, with Λ diagonal, and defining a predictor-corrector method starting from an exponential formula. They remarked that, if Λ is not diagonal, then one needs an approximation for $e^{-\Lambda t}$. Four years later, Snider [90] would correct a point overlooked by Guderley and Hsu in the truncation error analysis, although the order of accuracy does not change. In these years, Chu and Berman [18] also remarked that classic implicit RK methods are sufficient for "not too large" ODE systems.

3.2 The rebirth

Interest in these topics seemed to fade during the 1980s, only to resurface at the end of the decade. Indeed, to the best of the authors' knowledge, the next contribution to the theory were provided by Gallopoulos and Saad in 1989 [35]. The

⁵To the best of the authors' knowledge, this is the first use of the expression "exponential method".

authors considered systems of the form $u'(t) = -Au(t) + r$, that arise from discretization of linear parabolic PDEs. This translates into A being typically large and sparse, and possibly stiff, and they observed that the explicit solution can be easily expressed in terms of e^{-tA} but that the latter is hard to compute for the typical A and little changes if the solution is obtained with a time-stepping procedure. Thus, they proposed to solve the system via approximations of the exponential, either with a polynomial (Krylov method, later said to be not always good for stiff systems) or with suitable rational functions (Padé or Chebyshev approximations).⁶ We refer back to Section 2.2 for a brief recap of Krylov method.

In 1996, Hochbruck *et al.* [47] claimed that exponential integrators were not a novel discovery, but were considered impractical. They observed that, when A is large, it can be harder to solve a linear system $(I - \tau A)x = v$ than to compute $e^{\tau A}v$ (which is in turn as hard as computing $\varphi_1(\tau A)v$). This fact was proved the following by Hochbruck and Lubich [46].⁷ In [46], the authors also improved Gallopoulos and Saad's bounds on the error arising from the Krylov approximations. Here, the general class of exponential integrators was also considered with general parameters, and it was introduced as a class of methods for large systems of ODEs. The authors claimed that their result "justifies renewed interest in ODE methods that use the exponential or related functions of the Jacobian" (which in hindsight was a correct claim).

In 1998, Beylkin *et al.* [10] observed that the exponential of a dense matrix can be onerous unless it is a convolution (or circular) matrix, in which case one can exploit the fast Fourier transform (FFT). In 2002, Cox and Matthews [20] proposed a more straightforward derivation of the ETD methods, compared to Beylkin's, and provided explicit formulas for the coefficients. They derived new ETD methods of order 2, 3 and 4, and were the first to extend the methods to PDEs with (after discretization) nondiagonal linear part. In 2004, Koikari [58] derived order conditions for ETD methods using the classical rooted tree analysis. To the best of the authors' knowledge, this is the first mention of strong and weak order conditions.⁸ In the same year, Berland *et al.* [6] derived order conditions for several exponential methods (ETD, IF, CF Lie groups) for autonomous semilinear ODEs and proposed a general algorithm to build exponential integrators satisfying any given order. In 2005, Kassam and Trefethen [57] introduced a new method based

⁶The work was mostly motivated by the fact that the methods are amenable of parallelization procedures. The authors concluded by conjecturing that their methods and the possibility of parallelizing would make explicit methods popular again.

⁷These works mark the beginning of Hochbruck's long list of contributions on the topic.

⁸The role of weak order conditions will be clarified (at the latest) by Hochbruck and Ostermann in the 2005 papers as we are going to explain below.

on contour integrals to compute the φ -functions in order to solve instability issues of ETDRK4 pointed out by Cox and Matthews [20]. They found that ETD methods with this modification outperform other classes of methods for PDEs, testing on a variety of models.

2005 is also the year in which two fundamental contributions by Hochbruck and Ostermann were published. Firstly, in [49] the authors developed implicit (ETD) ExpRK of collocation type, which are based on the approximation of the nonlinear term in the right-hand side with a collocation polynomial, and analyzed their order theoretically for linear and semilinear parabolic problems. They assumed that the linear operator is sectorial, requirement that holds for many typical linear operators in parabolic problems. The assumption on the nonlinear term is local Lipschitz continuity. Then, in a second paper, Hochbruck and Ostermann [48] focused on explicit (ETD) ExpRK methods in the context of semilinear parabolic problems, dedicating additional efforts to the corresponding convergence analysis. The authors derived stiff order conditions tailored to this framework up to order $p=4$. Lastly, they proved that, in the case of constant step size, the conditions of stiff order p only have to be satisfied in a weaker form (in the sense of Koikari [58]).⁹ The idea underlying the concept of stiff order, which we will refer to again in the rest of the survey, is that classical convergence theory of time-integration methods assumes that the time step is small relative to the Lipschitz constant of the right-hand side; in the case of semilinear problems (1.1) with non-stiff g , this can be translated to $\|hA\|$ being small. However, the assumption is often violated if the linear term is stiff. For a method to have stiff order p , the global error must be uniformly bounded by Ch^p , for a constant C which is also independent on the stiffness of A , and not only on the step size h .

Still in 2005, Minchev and Wright [80] produced a review on exponential integrators for problems for which an approximation of the Jacobian is known. Qualitatively, these are semilinear problems with non-stiff nonlinear part. The authors addressed implementation issues, that are difficulties concerning the computation of exponential and related functions, referring in particular to Moler [82]. In 2006, Berland *et al.* [7] compared the performance of an IF and an ETD method of non-stiff order 4 on the Schrödinger equation. In particular, they concluded that the IF method has a generally worse performance and is more sensitive to

⁹We remark that Dujardin [30] in 2009 extended the results in Hochbruck [48] to the Schrödinger equation, which is not parabolic. More recently, the same author, in collaboration with Besse and Lacroix-Violet, further developed this line of research [9] by introducing and analysing high-order exponential integrators for nonlinear Schrödinger equations, including nonautonomous problems arising from rotating Bose-Einstein condensates, with rigorous convergence and structure-preservation results.

the regularity of the potential function V , while the ETD method is more sensitive on the regularity of the initial condition. More importantly, they commented on the fact that the equation is not really amenable of the stiff order analysis by Hochbruck and Ostermann [48, 49], since it is not a parabolic problem. Indeed, the tested methods have stiff orders 1 and 2 respectively, but this is not evident from the experiments.

In 2008, Caliarì and Ostermann [15] tested interpolation on Leja points on exponential methods of Rosenbrock type (expRosenbrock) for parabolic problems. This approach seemed particularly useful since the exact Jacobian does not have a specific structure and changes at each step, and performed particularly well on problems with large advection and moderate diffusion. The authors provide order conditions for expRosenbrock methods up to order 4, remarking that they are much simpler to formulate than those for ExpRK methods. In 2009, Hochbruck *et al.* [52] reformulated the expRosenbrock methods in terms of the perturbations $D_n^i := g(U_n^i) - g(u_n)$, where g is the remainder right-hand side minus the Jacobian. Since the perturbations are small vectors, applications of matrix functions to them using Krylov subspace methods require low-dimensional subspaces and therefore it would not be as computationally expensive as it was without the reformulation, where those same matrix functions would be applied to $g(u_n)$ or $g(U_n^i)$ instead. The authors provided rigorous convergence analysis up to order 4, as well as numerical tests on an advection-diffusion-reaction problem and a Schrödinger equation with time dependent potential.

In 2010, Al-Mohy and Higham [1] provided heuristic improvements to the original scaling and squaring algorithm behind MATLAB's `expm`, motivated by the propensity for overscaling of the latter. This means that in various cases too large of a value of s (number of times that one scales and squares) is needed and causes loss in floating-point arithmetic. The improvements mainly aim at addressing overscaling for upper triangular matrices, for which the problem is somehow more visible on average. The same authors, in [44], provided a comprehensive presentation of the state of the art regarding the computation $f(A)$ where f belongs to a class of functions which include the exponential. The category of methods based on similarity is represented by the Schurr-Parlett algorithm proposed in [26]. The rational approximation category is represented by Padé approximations with the improved scaling and squaring from Al-Mohy and Higham [1]. The algorithms which compute $f(A)b$ without needing $f(A)$ are represented by those based on Krylov subspaces, where f is only needed explicitly on a smaller Hessenberg matrix. The year 2010 represents a watershed moment also for this survey, as it was in that year that Hochbruck and Ostermann [50] published the last fundamental review on exponential integrators for two types

of stiff equations. The first type of equations are those with eigenvalues of the Jacobian having very large negative real part (the usual definition of stiffness) while the second is highly oscillatory problems (purely imaginary eigenvalues with large modulus).

4 Recent advances

The year 2011 marks a decisive milestone in the study of exponential methods, thanks to the publication of the already mentioned paper by Al-Mohy and Higham [2]. As explained in Section 2.2, this work provided a fundamental improvement in the field by introducing innovative algorithms for computing the action of the matrix exponential on a vector without explicitly calculating the entire matrix exponential, improving both computational efficiency and numerical stability.

Since 2011, numerous researchers have contributed to the study of ExpRK methods, resulting in a wealth of published works in this field. For conceptual clarity, we shall abandon chronological ordering in this section and instead we divide the material into three paragraphs, providing readers with a more accessible understanding of recent developments. We emphasize that such a division is arbitrary with no claim of being definitive — it simply represents a reasonable approach to addressing this complex subject matter.

4.1 Exponential methods of high stiff order

The first part of this section is mostly devoted to summarize a series of papers published by Vu Thai Luan and various co-authors. They provide specific order conditions, fundamental to construct the numerical methods. We recall here their most important contributions, and refer back to each paper for detailed derivations and calculations.

In 2014, Luan and Ostermann coauthored three papers. In [72], they considered expRosenbrock methods under the reformulation in Hochbruck *et al.* [52]. The authors addressed the convergence analysis in a different way: namely, instead of placing the exact solution into the numerical scheme and working with defects, they analyzed the local error first. This new approach provided a more systematic procedure to derive expRosenbrock methods of higher stiff order, which the authors used to construct families of methods of orders 3, 4 and 5 (see [72, Table 1]). In [73], the authors applied the approach developed for ex-

pRosenbrock methods in [72] to build order conditions up to 5 (whereas the literature at the time provided them only up to order 4) for ExpRK methods (see [73, Table 1]). In [71], the authors proved that, in the case of constant step size, one can reach stiff order 5 by only satisfying the order 5 condition in the weak form, as it was proved in [48] for orders up to 4. They concluded by building a method with stiff order 5, which needs 8 stages. We refer to the recapitulatory table [71, Table 1]. In 2016, the same authors [74] built more accurate and efficient expRosenbrock of (stiff) order 4 and 5 (compared to their collaboration in 2014), and of order 6 (see [74, Table 1]). The method exploits parallelism by minimizing the number of sequential stages, meaning that if two of the stages can be computed independently, then they count as one, greatly improving the corresponding computational time.

The following year, Luan [66] independently built a 2-stage order 4 expRosenbrock for time-dependent PDEs. It is the second superconvergent (here, meaning $p = 2\nu$, where p is the global convergence order and ν is the number of internal stages) exponential method ever built, after expRosenbrock-Euler. The performance was tested against other expRosenbrock methods and MATLAB's ode15s on systems which are either non-stiff or semilinear with non-stiff nonlinear part.

In 2020, Luan [67] proposed a construction for more accurate and efficient ExpRK methods of (stiff) order 4 and 5, compared to the existing ones obtained in [48, 71]. This was achieved by minimizing the number of sequential stages, similarly to what was done in [74] for expRosenbrock. Moreover, he exploited both results from [2]: efficient algorithm for computing $e^{tA}b$ and augmented matrix \tilde{A} such that $e^{\tilde{A}}$ is equivalent to computing linear combinations of φ -functions. To do so, the new methods are built such that the i -th stage is defined by a single linear combination of terms $\varphi_k(c_i h A)$, where $\{c_i\}_i$ are the abscissas of the method. The resulting ExpRK of order 4 has 6 stages instead of 5, but 4 sequential stages. The 5th order one has now 10 stages instead of 8, but 5 sequential stages. Refer to [67, Table 1 & Section 6] for a detailed explanation. The implementation of the methods makes use of their own code `phipm_simul_iom`, publicly available. In the same year, Luan *et al.* [69] introduced multirate ExpRK, using different time scales for the linear term (fast) and the nonlinear term (slow) of a semilinear (possibly abstract) ODE. The methods need to be such that the i -th stage is defined by a single linear combination of terms of the form $\varphi_k(c_i h A)$ (as in [67]) which is a nontrivial requirement. The idea is to approximate the nonlinear term in both internal and final stages with a polynomial built assuming a small time step h and then solve the ODE with the polynomials replacing the nonlinear terms, using a large time step H . The idea builds on [51], where this approach was applied to exponential multistep methods (of Adams type).

Shifting our focus to the most recent advances, in 2024, Luan *et al.* [75] explored applications to gene regulatory networks, which are defined by a certain number of semilinear ODEs where the linear part is diagonal and the nonlinear part can be more or less stiff. In the examples considered in the paper, the ODEs are low dimensional, the linear part may be stiff in the sense of highly oscillatory and the nonlinear one is only moderately stiff (in these models, the nonlinear term can become very stiff when many gene products are involved). The authors build an order 2 exponential method which is Rosenbrock-like, in the sense that derivatives of the right-hand side appear in the coefficients. They compare accuracy (error vs. h) and efficiency (error vs. CPU time) with other exponential methods and classical RK. The proposed method is the most efficient and second most accurate after `expRosenbrock-Euler`. The better performance than classical RK may be partly due to the matrix being diagonal (thus, exponential being even easier to compute). This latter study thus aligns with the literature on the application of `ExpRK` methods to mathematical models in biology and physics. A detailed analysis of the relevant contributions will be provided in the final part of this section. In the same year, Luan and Alhsmy [68] developed the first order 6 `ExpRK` methods for the time integration of stiff parabolic PDEs, leveraging the exponential B-series theory extended by Luan and Ostermann for stiff problems [70]. They derived 36 stiff order conditions, see [68, Table 1], and remarked that despite these methods requiring a large number of stages, their computational cost is similar to that of a 6 stages method.

We conclude this section with a very recent development concerning (high-order) Lawson schemes. In these schemes, internal stages typically require matrix-vector multiplications with matrices of the form $e^{(c_j - c_i)hA}$, where the $\{c_i\}_i$ are again the abscissas of the method. For this reason, having equispaced abscissas is a desirable property, since it requires the computation of only one matrix exponential. Lawson observed this in his seminal paper [63] with a focus on `RK4`, while Golden [37] recently explored the idea in detail. Using the Newton-Raphson method to establish the coefficients, he arrived at developing a sixth order method with eight stages. Finally, he provided evidence of its efficiency by testing it on the 2D incompressible Navier-Stokes equation.

4.2 Exponential methods for DDEs

The second part is dedicated to the analysis of recent advances in applications of `ExpRK` methods to delay differential equations (DDEs). Indeed, a series of papers published in the past 15 years demonstrate the recent interest in applying `ExpRK`

methods to DDEs, namely, equations of the form

$$u'(t) = G(t, u_t), \quad (4.1)$$

where u_t is a function defined on an interval $[-\tau, 0]$ for a finite delay $\tau > 0$ as

$$u_t(\theta) = u(t + \theta), \quad \theta \in [-\tau, 0]$$

and represents the history of u at time t . The simplest examples of DDEs that are found in applications are defined by a single discrete delay, i.e., they can be written as

$$u'(t) = g(t, u(t), u(t - \tau)). \quad (4.2)$$

Observe that, in terms of implementation of time-integration methods, the delayed term (4.2) does not, in principle, entail any issue provided that the step size h used is less than τ . In this case, indeed, the past value of the function at a certain time step can be retrieved from the approximation obtained at previous time steps. While choosing $\tau = mh$ for some integer m can be especially convenient in the case of a single discrete delay, in general we can write $\tau = (m - \delta)h$ for some integer m and some $\delta \in [0, 1)$.

However, the presence of the delay term may affect stability of the underlying numerical methods [5], which has motivated researchers to investigate the stability of Exprk methods for DDEs of the form (4.2). Xu *et al.* [96] study the case of a scalar, linear equation of the form

$$u'(t) = \lambda u(t) + \mu u(t - \tau), \quad u_{t_0} = \phi, \quad (4.3)$$

and analyze stability properties of Exprk methods when applied to DDEs with a discrete delay, for which (4.3) plays the role of a test function, and its delayed term is treated as nonlinear. Whenever $|\lambda| + \text{Re}(\mu) < 0$, the solution of (4.3) is asymptotically stable [54] and P -stable methods for DDEs are those which preserve this property when h is such that $\tau = mh$ for an integer $m > 0$. Note that the assumption $\tau = mh \geq h$ implies that the delayed argument lies entirely outside the current step interval, which means that explicit schemes retain their explicit computational structure. A numerical method for DDEs is GP -stable if it preserves asymptotic stability regardless of the value of $h > 0$. As shown in [96], the concepts of P - and GP -stability coincide in the case of Exprk methods for DDEs. Some years later, Zhao *et al.* [99] studied various contractivity and stability concepts for explicit Exprk methods in the context of semilinear DDEs of the form

$$u'(t) = Lu(t) + g(t, u(t), u(t - \tau)). \quad (4.4)$$

Whenever $|\lambda| + \operatorname{Re}(\mu) \leq 0$, the solution of (4.3) is contractive, that is, it satisfies $|u(t)| \leq \|\phi\|_\infty$ for all $t \geq t_0$. P -contractive methods for DDEs are those which preserve this property when h is such that $\tau = mh$ [93] and GP -contractive methods for DDEs are those which preserve this property for any choice of h . Zhao *et al.* provide sufficient conditions for GP -contractivity of explicit Exprk methods and prove GP -contractivity for various methods of order 1 and 2. They also extend the concept of GP -stability to nonautonomous equations, which is satisfied by the Magnus method described in [38]. Finally, they analyze RN - and GRN -stability, introduced again in [93], and derived sufficient conditions on L, g in (4.4) such that, for all u, v solutions of (4.4) with initial conditions $u_{t_0} = \phi, v_{t_0} = \psi$, the following holds:

$$\|u - v\|_\infty \leq \max_{t \in [t_0 - \tau, t_0]} |\phi(t) - \psi(t)|.$$

RN -stable methods for DDEs are those preserving this property for $\tau = mh$, while GRN -stable ones do not need the restriction on h . Zhao *et al.* provide sufficient conditions on the coefficients for GRN -stability of explicit Exprk methods.

Zhao *et al.* [100] analyzed D -convergence. An Exprk method for DDEs is D -convergent of order p if, when applied to (4.4) with initial condition $y_0 = \phi$, it satisfies

$$|u_n - u(t_n)| \leq Ch^p, \quad n \geq 0, \quad h \leq \bar{h},$$

where \bar{h} depends only on the regularity properties of g and the coefficients of the method, and the constant C depends on $n\tau$, the regularity properties of g , the bounds on the first p derivatives of the exact solution and the coefficients of the method. The authors extend the concepts of diagonal and algebraic stability to Exprk methods and prove that the combination of both, together with having stage order p , implies D -convergence of order p , provided that the interpolation used to reconstruct the continuous solution is accurate enough. They also prove that an Exprk method for DDEs which is exponentially algebraically stable and diagonally stable is also GDN -stable, namely, if u, v are solutions of (4.4) with initial conditions ϕ, ψ , they satisfy

$$|u_n - v_n| \leq C \max_{t \in [t_0 - \tau, t_0]} |\phi(t) - \psi(t)|$$

for all step sizes $h \leq \bar{h}$, where the constant C depends on τ , the time-integration interval length, the regularity properties of g , the bounds on the first p derivatives of the exact solution and the coefficients of the method. They observe that the ETD Euler method, as well as a certain family of methods of stiff order 2, are D -convergent (of orders 1 and 2, respectively) and GDN -stable.

Besides stability, some research has been devoted to the accuracy of ExpRK methods for DDEs. In particular, Fang and Zhan [34] follow the abstract approach in [48] to derive stiff order conditions up to order 5 for semilinear DDEs of the form (4.4), and prove the analogous convergence theorem and support the results with numerical tests on PDEs whose semi-discretization in space gives DDEs of the target form.

More recently, Andò and Vermiglio [3, 4] investigated the application of ExpRK methods for the time-integration of general DDEs of the form (4.1), as well as renewal equations (REs), namely, equations of the form

$$u(t) = F(t, u_t),$$

where F is Lipschitz continuous from $\mathbb{R} \times L^1([-\tau, 0]; \mathbb{R}^N)$ to \mathbb{R}^N for a suitable $N > 0$, which in practice means that F acts on the history through integration against a kernel (possibly composed with a smooth nonlinearity) rather than through pointwise evaluation. Andò and Vermiglio consider, in particular, the reformulation of DDEs and REs as abstract (ordinary) differential equations (ADEs), which comes from the sun-star theory [27, 28]. The ADEs obtained are semilinear regardless of the right-hand sides of the original delay equations, which makes them amenable of time-integration via ExpRK methods. Since the abstract setting provided by the sun-star calculus includes both DDEs and REs, the goal in [4] is to define a unifying framework to study the derivation of ExpRK methods for delay equations, as well as their convergence properties. Another approach to solve problems defined by DDEs is to discretize the resulting ADEs into a finite-dimensional system of ODEs and apply ExpRK methods to the latter, as done in [3]. Therein, the ODEs are obtained via pseudospectral discretization [11], which gives rise to a stiff linear term, with eigenvalues having large negative real parts. Several numerical tests with EXPINT, a MATLAB package for exponential integrators [8] provide experimental validation of the theoretical convergence order for some selected ExpRK methods and show that, when the linear part of the DDE right-hand side is stiff, it must be explicitly extracted and treated via the exponential integrator together with the term obtained by the pseudospectral discretization, rather than lumped into the nonlinear part.

The most recent contribution on the applications of ExpRK methods to DDEs is represented by the extension of the ExpRK methods of collocation type introduced in [49] to semilinear (abstract) parabolic problems such that the nonlinear term is defined by a discrete delay [53]. The delay can be time-dependent, as long as it is non-vanishing and that the map $t \mapsto \tau(t)$ is strictly increasing. This produces a mesh of non-equispaced breaking points, i.e., points of discontinuity of some derivative of the solution. The authors propose two ways to extend the

methods in [49] to the new classes of equations, both taking the breaking points into account to define a continuous approximation of the solution in a piecewise fashion. They prove that the methods are convergent with order s , the number of stages, and obtain superconvergence results provided some additional restrictions on the parameters are satisfied.

4.3 Exponential methods for mathematical modeling

In this third and final part, we present an analysis of the literature employing ExpRK methods in modeling contexts, which has garnered significant interest in recent years.¹⁰ The first contribution we examine is from 2011 and is authored by Dimarco and Pareschi [29]. In this work, the authors, motivated by the computational challenges arising from rarefied gas dynamics, introduce a class of ExpRK methods for stiff kinetic equations, with particular emphasis on the space homogeneous Boltzmann equation

$$\partial_t f = \frac{1}{\varepsilon} Q(f, f),$$

where $Q(f, f)$ is the Boltzmann collision operator, which describes the evolution of particle distribution functions in a gas by incorporating the effects of collisions. In the fluid regime, the stiffness of the problem makes standard explicit methods inefficient, as the equation takes the form

$$Y' = F(Y) + \frac{1}{\varepsilon} R(Y), \quad (4.5)$$

where $0 < \varepsilon \ll 1$ is a small parameter (enough to define two different time scales) and $R(Y)$ is a dissipative relaxation operator. To avoid solving nonlinear systems, the authors propose the following decomposition of the collision operator:

$$R(Y) = N(Y) + A(E(y) - Y), \quad (4.6)$$

where $E(y)$ represents the local equilibrium and A is a parameter chosen to linearize the problem on asymptotically large time scales. ExpRK methods treat the linear part exactly using exponential functions and the nonlinear part explicitly, yielding schemes of the form

$$Y_{n+1} = e^{-\lambda} Y_n + \lambda \sum_{i=1}^s W_i(\lambda) \left(\frac{P(Y_i, Y_i)}{\mu} - M \right) + (1 - e^{-\lambda}) M. \quad (4.7)$$

¹⁰For the reader's convenience, given that the equations pertain to physical models, we have opted to retain the original notation used in each cited paper throughout this section.

Here, M represents the local Maxwellian equilibrium, $P(f, f) = Q(f, f) + \mu f$ is the non-negative gain part of the collision operator obtained by adding a linear term to enable the exponential splitting, $\lambda = \mu \Delta t / \varepsilon$, and $W_i(\lambda)$ are suitable coefficients. This approach ensures unconditional stability, correct asymptotic preservation, and key properties such as positivity and entropy conservation, making these methods suitable for both deterministic schemes and Monte Carlo simulations. Finally, the authors highlight how their approach is closely related to the IF methods, sharing the same mathematical structure but offering greater flexibility in designing high-order schemes. In 2014, Michels *et al.* [78] applied ExpRK methods to the modeling framework of Newton's equations of motion for elastodynamic systems. These systems are described by second-order differential equations of the form

$$M \cdot x''(t) + D \cdot x'(t) + K \cdot (x(t) - x_0) + F_{\text{nl}}(x(t)) = 0, \quad (4.8)$$

where M is the mass matrix, D the damping term, K the stiffness matrix and F_{nl} represents the nonlinear forces. The presence of very high frequencies in the solution spectrum makes standard explicit methods ineffective due to time step restrictions, while implicit schemes suffer from uncontrollable numerical viscosity. To overcome these difficulties, the authors propose the use of exponential integrators that solve the linear part exactly using the exponential function of the matrix associated with the system

$$x(t + \Delta t) = e^{-\Delta t A} x(t) + \int_0^{\Delta t} e^{-(\Delta t - s)A} F_{\text{nl}}(x(t + s)) ds, \quad (4.9)$$

where $A = M^{-1}K$ represents the linear dynamics of the system. This scheme allows to treat much larger time steps than classical implicit methods, ensuring rigorous energy conservation and long-term stability. Application to complex models such as fabrics, lattice structures and materials with friction demonstrates the effectiveness of this methodology in the simulation of rigid elastodynamic systems.

It is in the last years, however, that we have witnessed a significant increase in the number of studies on the application of ExpRK methods across various domains. This growing body of research reflects the increasing interest in these numerical techniques and their potential advantages.

Recently, renewed attention has been devoted to the numerical integration of the nonlinear Schrödinger equation, particularly in connection with high-order exponential integrators and structure-preserving techniques. For instance, in 2022, the Schrödinger equation was revisited through a scalar auxiliary variable (SAV) formulation combined with IF methods, allowing the construction of linearly implicit schemes of order up to four that exactly preserve a discrete modi-

fied energy functional [56]. The SAV approach is based on the introduction of an additional scalar variable associated with the nonlinear part of the Hamiltonian, which enables efficient time integration while avoiding the solution of nonlinear systems. The proposed methods were shown to be competitive in terms of accuracy versus computational cost when compared with other SAV-based and energy-oriented schemes.¹¹ More recently, Li and Li [64] proposed multiple relaxation ExpRK methods for the nonlinear Schrödinger equation, extending the relaxation framework to exponential integrators. Their approach introduces several relaxation parameters that are determined at each time step so as to enforce the exact preservation of multiple invariants of the continuous problem, including mass and energy, while maintaining arbitrary high-order accuracy. The resulting schemes remain explicit in the linear part and avoid nonlinear solvers, while achieving simultaneous invariant preservation at the fully discrete level. In a different but related direction, Ji *et al.* [55] developed low regularity exponential-type integrators for the derivative nonlinear Schrödinger equation. Their analysis is based on a careful reformulation of the equation that isolates resonant interactions and allows for sharp error bounds under minimal regularity assumptions on the solution. The proposed methods are shown to converge without requiring classical Sobolev smoothness, thereby addressing a major limitation of standard exponential integrators when applied to derivative nonlinearities.

Various authors have focused on the development of high-order time integration schemes for gradient flows and semilinear parabolic equations that can simultaneously achieve high accuracy and preserve intrinsic structural properties, such as energy dissipation, equilibria, and inequality constraints. ExpRK methods are particularly attractive due to their stability properties, but their practical implementation often faces challenges related to the use of exponential operators, including excessive damping of high-frequency modes and difficulties in maintaining structure-preserving features at higher order. Within this context, Teng and Zhang [91] proposed structure-preserving reconstructions of IF schemes and developed an exponential-free framework achieving up to third-order accuracy. Wang *et al.* [94] further extended this line of research by constructing an up-to-fourth-order ExpRK framework for a large class of phase-field models, and by establishing unconditional energy dissipation for the resulting schemes. Their approach replaces the exponential functions in the IF framework with recursive ap-

¹¹We note that the SAV-based schemes preserve a modified energy rather than the original Hamiltonian of the Schrödinger equation. While this modification does not affect the continuous problem and is shown numerically to keep the true energy well controlled over moderate time intervals, it may lead to qualitative differences in long-time simulations. The authors justify this choice by the resulting linear implicitness, high-order accuracy, and favorable efficiency-error balance.

proximations, or with a combination of exponential and linear functions, thereby avoiding the excessive damping of high-frequency modes while preserving equilibria. These works complement earlier studies [97, 98], in which fourth-order reconstructions of IF schemes were developed to preserve maximum principles or more general inequality constraints.

Starting from 2024, the literature has seen several contributions aimed at improving the efficiency of exponential integrators for specific yet widespread classes of problems, as well as at investigating and mitigating order reduction effects. Within this line of research, Caliari *et al.* [14] proposed a fundamental acceleration technique for ExpRK integrators and Lawson schemes to efficiently solve semi-linear advection-diffusion-reaction equations. The model under consideration is of the form

$$\partial_t u(t, x) = \nabla \cdot (a(x) \nabla u(t, x)) + \nabla \cdot (b(x) u(t, x)) + r(t, x, u(t, x)), \quad (4.10)$$

where $a(x)$ is the diffusion tensor, $b(x)$ is the velocity field, and $r(t, x, u)$ represents the reaction term. The main computational challenge of exponential integrators lies in efficiently evaluating matrix functions such as $e^{\tau A}$ and $\varphi_1(\tau A)$, whose definition can be obtained from (2.2). To improve efficiency, the authors propose choosing a linear operator A with constant coefficients so that the computation of these functions can be performed more efficiently, for example, using FFT in cases with constant diffusion. The operator A is defined as

$$Au = \lambda \sum_{\mu=1}^d \max_x a_{\mu\mu}(x) \partial_{x_\mu}^2 u + \beta \cdot \nabla u, \quad (4.11)$$

where the parameter λ is determined to ensure stability based on a spectral analysis of the Laplace operator. For instance, when solving pure diffusion equations, the authors demonstrate that the ETD Euler (1.4) is unconditionally stable for $\lambda \geq 1/2$. Meanwhile, the Lawson-Euler scheme:

$$u^{n+1} = e^{\tau\lambda\Delta} (u^n + \tau(1-\lambda)\Delta u^n), \quad (4.12)$$

where Δ denotes the Laplacian operator, is stable for $\lambda \geq 0.218$, offering greater flexibility in the choice of the operator. This approach is validated numerically on advection-diffusion-reaction equations in 1D, 2D, and 3D, demonstrating significant computational efficiency improvements over IMplicit-EXplicit (IMEX) schemes, which treat implicitly only the linear part of the system, and traditional exponential methods.

Again in 2024, multiple works were published on the subject of order reduction of ExpRK methods, focusing specifically on problems with non-commutative

operators. In a first work, Hoang [45] studied the application of ExpRK methods to nonlinear parabolic equations with non-commutative operators. The model under consideration is an equation of the form

$$\partial_t u - \Delta u = f(\nabla u, u), \quad u(0) = u_0, \quad (4.13)$$

where f is a nonlinear function. To understand the numerical behavior of such methods, the analysis focuses on the simplified problem

$$u'(t) + Au(t) = Bu(t), \quad u(0) = u_0, \quad (4.14)$$

where A is an operator generating an analytic semigroup, while B is relatively bounded¹² with respect to A . The method follows an approach where A is treated exactly and B is integrated explicitly. The work focuses on the phenomenon of order reduction, which occurs when the method's convergence is lower than theoretically expected. The analysis shows that, in the presence of non-commutative operators, higher-order methods suffer from accuracy degradation unless suitable regularity conditions on the initial data are met. Through theoretical results and numerical simulations, the study highlights the limitations and possible improvements of exponential methods in handling such equations.

In a second work authored by Dang and Hoang [24], the authors investigated the issue of order reduction in third-order ExpRK methods applied to problems with non-commutative operators. In particular, they proved that if specific order conditions are not satisfied, the method's accuracy may drop to approximately 2.5 instead of 3. To overcome this problem, the authors propose a 4 stages ExpRK method that meets all the necessary conditions to preserve the expected order. The studied method takes the form

$$\begin{aligned} U_n^i &= e^{-c_i \tau A} u_n + \tau \sum_{j=1}^{i-1} a_{ij} (-\tau A) B U_n^j, \\ u_{n+1} &= e^{-\tau A} u_n + \tau \sum_{i=1}^s b_i (-\tau A) B U_n^i \end{aligned} \quad (4.15)$$

¹²In general, an operator B is said to be relatively bounded with respect to another operator A if there exist constants $C \geq 0$ and $\alpha \geq 0$ such that

$$\|Bu\| \leq C\|Au\| + \alpha\|u\|, \quad \forall u \in D(A).$$

This, in practice, means that B is not necessarily bounded, but its growth is controlled by A . This property is crucial in spectral analysis and numerical methods, ensuring stability and well-behaved approximations in the presence of unbounded operators.

with the coefficients $a_{ij}(-\tau A)$ and $b_i(-\tau A)$ defined through linear combinations of the functions $\varphi_k(-\tau A)$. A detailed convergence analysis is conducted, based on a semigroup formulation in a Banach space, showing that adopting higher regularity on the initial data helps to avoid order reduction. Numerical tests confirm the validity of the approach, demonstrating significant improvements in accuracy and stability compared to standard three-stage methods.

In an additional work, Einkemmer *et al.* [31] studied the efficiency of exponential integrators for advection-dominated problems. They compared Leja and Krylov-based methods for computing matrix exponential actions, evaluating their performance against explicit RK schemes. Their results show that while exponential integrators perform similarly to explicit methods in fully advection-dominated regimes, they offer advantages when parts of the domain exhibit diffusion. Among the tested approaches, Leja-based methods proved more efficient than Krylov iterations, particularly on modern supercomputers where inner product computations are costly. The paper focused on performance modeling: counting core operations to understand theoretical cost ratios between exponential integrators and explicit schemes. The recent complementary work [25], instead, measured real CPU time in numerical experiments. It showed Krylov-based methods can work better for small steps in real CPU usage – a nuanced result not apparent from the theoretical model.

In 2024, Wang *et al.* [95] presented a unified theoretical framework for analyzing the energy dissipation properties of three classes of RK methods applied to phase-field Crystal (PFC) models, which describe crystalline dynamics at atomic scales. The fundamental model is given by the gradient flow

$$\partial_t \Phi = M \frac{\delta E}{\delta \Phi}, \quad (4.16)$$

where the Swift-Hohenberg free energy functional is

$$E[\Phi] := \int_{\Omega} \left[\frac{1}{2} \Phi (I + \Delta)^2 \Phi + F(\Phi) \right] dx \quad (4.17)$$

with

$$F(\Phi) = \frac{1}{4} \Phi^4 - \frac{\varepsilon}{2} \Phi^2, \quad f(\Phi) = -F'(\Phi).$$

Two common choices of M lead to different evolution equations: the L^2 gradient flow ($M = -I$) results in the Swift-Hohenberg equation, while the H^{-1} gradient flow ($M = \Delta$) leads to the conservative volume PFC equation. A key challenge is ensuring maximum norm boundedness of solutions without assuming global

Lipschitz continuity of $f(\Phi)$. To address this, a linear stabilization technique is introduced

$$L_\kappa \Phi := (I + \Delta)^2 \Phi + \kappa \Phi, \quad f_\kappa(\Phi) := \kappa \Phi + f(\Phi), \quad (4.18)$$

where $\kappa \geq 0$ is a stabilization parameter. The study provides a unified framework for analyzing energy dissipation across different RK methods. The main result states that numerical solutions maintain maximum norm boundedness at all stages, provided that the differentiation matrix associated with the RK method satisfies a positivity condition. Among the three classes analyzed, ExpRK methods, including explicit and ETD RK methods, are particularly effective, as they treat stiff linear terms exactly while handling nonlinear terms explicitly, improving stability and computational efficiency. The other two classes examined are implicit-explicit RK and corrected integrating factor RK methods. The novelty of the work lies in proving that these methods preserve the original energy dissipation laws if their differentiation matrices are positive definite. This is achieved using a differential form approach, discrete orthogonal convolution kernels, and mathematical induction, paving the way for rigorous energy stability in dissipative semilinear parabolic problems.

5 Examples

To conclude our manuscript, we present three examples that illustrate the general functionality of ExpRK methods. To this end, we consider standard methods of two different orders (2 and 4) and compare an ExpRK method with a Rosenbrock method and a classical explicit RK method of the same order. The purpose of these examples is not to draw general conclusions in favor of exponential methods, nor to test specific features or provide an exhaustive assessment of their performance. Rather, our goal is to gain insight into which classes of equations could benefit from the use of ExpRK methods, by means of comparison with well-established classical schemes. In this perspective, the examples are meant to have a primarily illustrative and didactic value, offering a practical introduction to the behavior of ExpRK methods in representative settings.

We remark that, in 2020, Montanelli and Bootland [83] published a comprehensive review of exponential methods, comparing various formulas for solving stiff PDEs in multiple dimensions. Their analysis, focused on periodic boundary conditions and semilinear stiff PDEs, concluded that the ETDRK4 scheme by Cox and Matthews [20] (denoted as ExpRK4 in the following, for ease and uniformity of notation) remains one of the most effective approaches despite the proliferation of more complex alternatives. Their work is particularly notable for its examination of both diffusive problems (characterized by large negative eigenvalues) and

dispersive problems (distinguished by their highly oscillatory nature), as well as its discussion of computational techniques for the φ -functions based on contour integrals. In line with Montanelli and Bootland's review [83], we restrict our attention to explicit ExpRK methods. These methods are by far the most widely employed in practice: they often yield superior performance when stiffness is predominantly carried by the linear dynamics while the nonlinear contribution remains comparatively mild. Moreover, they avoid the extra complications associated with Newton-type iterations, such as the need to solve nonlinear systems at every step and to tune iterative solver tolerances; these steps bring additional computational cost and potential convergence issues. For these reasons, explicit exponential formulations are a natural and pragmatic choice in many applications. Nevertheless, in this work we also deliberately test them on less typical, small-scale examples to probe their behavior in unconventional settings. We remark that, due to the small size, the treatment of our test cases is not expected to particularly benefit from any of the advanced techniques to approximate matrix exponentials described in this survey (and in particular in Section 2.2), which is why we choose to compute all relevant matrix exponential using MATLAB's `expm`.

The three didactic examples we propose have dimension 1, 2 and 100 (where here the word dimension refers to the size of the system of ODEs), respectively. Specifically, for the 1D case, we revisit the introductory example presented by Cox and Matthews in 2002 [20, Section 4.1]; for the 2D case, we examine a classical model from nonlinear mechanics and dynamical systems, namely the Duffing oscillator¹³; the last example is the result of applying the pseudospectral discretization method [11] to an Ikeda DDE [33]. In what follows, we briefly introduce three order 2 methods and three order 4 methods for solving stiff systems of ODEs. As already noticed, stiff ODEs require specialized numerical techniques due to their rapidly changing components existing alongside slowly evolving ones. The six methods under consideration represent different approaches to handling stiffness, while all maintaining second-order [resp. fourth-order] accuracy. Their properties suggest different performance characteristics when applied to stiff problems.

For each test case examined, we conduct a convergence analysis by evaluating the relative error against the analytical solution at the final time for various time steps. We recall that the relative error is calculated as

$$\text{Error} = |(u_{\text{numerical}} - u) / u|. \quad (5.1)$$

¹³For a comprehensive analysis of the nonlinear Duffing oscillator and its applications, we recommend the volume [59] to the interested reader.

In the first example we present, there exists an analytical expression for the solution u of the one-dimensional ODE, and we use this exact expression in the formula above. In the remaining examples, such an explicit, exact expression does not exist, and we compare the methods against a reference expression for the solution u . Additionally, we measure the computational time required by each method, enabling a comprehensive comparison of both accuracy and efficiency. To estimate the elapsed time, we perform each simulation several times (1000 in the first test, 10 in the second and 100 in the third) and reported the average. This dual assessment, as emphasized earlier, is particularly important when dealing with stiff systems, where the choice of numerical method significantly impacts the simulation's precision and performance. Through this analysis, we aim to further illustrate the practical considerations discussed in previous sections and provide additional insights into the behavior of these methods across different problem types and stiffness regimes. All simulations were performed in MATLAB, version R2024a, on a 64-bit Microsoft Windows system.

5.1 The six methods

Here, we describe the six numerical methods which we have used for our simulations, namely for solving (in one case non-autonomous, in the other cases autonomous) systems of ODEs of the form

$$\frac{d\mathbf{u}}{dt} = \mathbf{F}(t, \mathbf{u}). \quad (5.2)$$

In particular, we consider three methods of order 2 and three methods of order 4.

5.1.1 Explicit Runge-Kutta 2 method (RK2)

RK2 updates the solution as follows:

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{F}(t_n, \mathbf{u}_n), \\ \mathbf{k}_2 &= \mathbf{F}\left(t_n + \frac{h}{2}, \mathbf{u}_n + \frac{h}{2}\mathbf{k}_1\right), \\ \mathbf{u}_{n+1} &= \mathbf{u}_n + h\mathbf{k}_2. \end{aligned}$$

5.1.2 Exponential Runge-Kutta 2 method (ExpRK2)

We can rewrite the equation as follows:

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u} + \mathbf{g}(t, \mathbf{u}), \quad (5.3)$$

where $A\mathbf{u}$ is the linear term. The ExpRK2 method updates the solution as follows:

$$\begin{aligned}\mathbf{a} &= e^{hA}\mathbf{u}_n + h\varphi_1(hA)\mathbf{g}(t_n, \mathbf{u}_n), \\ \mathbf{u}_{n+1} &= \mathbf{a} + h\varphi_2(hA)(\mathbf{g}(t_n + h, \mathbf{a}) - \mathbf{g}(t_n, \mathbf{u}_n)),\end{aligned}$$

where φ_1 and φ_2 are defined from (2.2).

5.1.3 Rosenbrock 2 method (Rb2)

The Rosenbrock method is an implicit method designed for stiff ODEs, incorporating a Jacobian-based correction. The update rule is given by

$$\begin{aligned}\mathbf{k} &= (I - \gamma h \mathbf{J}_n)^{-1} \mathbf{F}(t_n, \mathbf{u}_n), \\ \mathbf{u}_{n+1} &= \mathbf{u}_n + h\mathbf{k},\end{aligned}$$

where

$$\mathbf{J}_n = \left. \frac{\partial \mathbf{F}}{\partial \mathbf{u}} \right|_{(t_n, \mathbf{u}_n)}$$

is the Jacobian matrix, and γ is a stabilization parameter; in particular, $\gamma = 0.5$ provides a method of order 2.

5.1.4 Explicit Runge-Kutta 4 method (RK4)

The classical order 4 method that we consider is RK4, which we already applied to our toy model in Section 2.2. For the general equation, RK4 updates the solution as follows:

$$\begin{aligned}\mathbf{k}_1 &= \mathbf{F}(t_n, \mathbf{u}_n), \\ \mathbf{k}_2 &= \mathbf{F}\left(t_n + \frac{h}{2}, \mathbf{u}_n + \frac{h}{2}\mathbf{k}_1\right), \\ \mathbf{k}_3 &= \mathbf{F}\left(t_n + \frac{h}{2}, \mathbf{u}_n + \frac{h}{2}\mathbf{k}_2\right), \\ \mathbf{k}_4 &= \mathbf{F}(t_n + h, \mathbf{u}_n + \mathbf{k}_3), \\ \mathbf{u}_{n+1} &= \mathbf{u}_n + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4).\end{aligned}$$

5.1.5 Exponential Runge-Kutta 4 method (ExpRK4)

Among the order 4 exponential methods, we consider ExpRK4 [20, Eqs. (26)-(29)]. Considering again the equation,

$$\frac{d\mathbf{u}}{dt} = A\mathbf{u} + \mathbf{g}(t, \mathbf{u}), \quad (5.4)$$

ExpRK4 updates the solution as follows:

$$\begin{aligned}
\mathbf{g}_1 &= \mathbf{g}(t_n, \mathbf{u}_n), \\
\mathbf{a} &= e^{hA/2} \mathbf{u}_n + \frac{h}{2} \varphi_1(hA/2) \mathbf{g}_1, \\
\mathbf{g}_2 &= \mathbf{g}(t_n + h/2, \mathbf{a}), \\
\mathbf{b} &= e^{hA/2} \mathbf{u}_n + \frac{h}{2} \varphi_1(hA/2) \mathbf{g}_2, \\
\mathbf{g}_3 &= \mathbf{g}(t_n + h/2, \mathbf{b}), \\
\mathbf{c} &= e^{hA/2} \mathbf{a} + \frac{h}{2} \varphi_1(hA/2) (2\mathbf{g}_3 - \mathbf{g}_1), \\
\mathbf{g}_4 &= \mathbf{g}(t_n + h, \mathbf{c}), \\
b_1(hA) &= \varphi_1(hA) - 3\varphi_2(hA) + 4\varphi_3(hA), \\
b_2(hA) &= 2\varphi_2(hA) - 4\varphi_3(hA), \\
b_3(hA) &= -\varphi_2(hA) + 4\varphi_3(hA), \\
\mathbf{u}_{n+1} &= e^{hA} \mathbf{u}_n + h(b_1(hA) \mathbf{g}_1 + b_2(hA) (\mathbf{g}_2 + \mathbf{g}_3) + b_3(hA) \mathbf{g}_4),
\end{aligned}$$

where φ_1 , φ_2 and φ_3 are defined from (2.2).

5.1.6 Rosenbrock 4 method (Rb4)

The order 4 Rosenbrock-Wanner method we consider is defined by four stages. Without loss of generality, for ease of notation we assume the problem is autonomous and let $\mathbf{J}_n = \partial \mathbf{F} / \partial \mathbf{u} |_{\mathbf{u}_n}$. The update rule is given by

$$\begin{aligned}
(I + \gamma_{ii} h \mathbf{J}_n) \mathbf{k}_i &= h f \left(\mathbf{u}_n + \sum_{j=1}^{i-1} \alpha_{ij} \mathbf{k}_j \right) + h \mathbf{J}_n \sum_{j=1}^{i-1} \gamma_{ij} \mathbf{k}_j, \quad i = 1, \dots, 4, \\
\mathbf{u}_{n+1} &= \mathbf{u}_n + \sum_{i=1}^4 \beta_i \mathbf{k}_i,
\end{aligned}$$

for constant $\alpha_{ij}, \beta_i, \gamma_{ij}$ which are chosen in order to satisfy the order 4 conditions (see, e.g., [40, Section IV.7]).

5.2 1D. C&M starting example

Following the same approach presented by Cox and Matthews [20] in their analysis of ETD methods, we consider a non-autonomous ODE model with rapid decay, characteristic of stiff systems

$$u'(t) = ku(t) + \sin(t), \quad u(0) = u_0 = 1, \quad (5.5)$$

where we have chosen $k = -100$ to generate the rapid linear decay typical of stiff problems, in contrast with the $\mathcal{O}(1)$ time scale of the forcing term. This equation, while simple in form, captures the essential features of stiff systems: a fast decaying homogeneous solution combined with a (relatively) slow varying forcing term. For subsequent evaluation of the different numerical schemes, we utilize the exact analytical solution given by

$$u(t) = -\frac{-e^{kt}(2+k^2) + \cos(t) + k\sin(t)}{1+k^2}. \quad (5.6)$$

The behavior of this differential equation can be divided into two distinct phases: an initial fast transient phase, during which the solution rapidly approaches the slow manifold, followed by a phase in which the solution evolves along this slow manifold. As noticed in [20], when $k < 0$ and $|k| \gg 1$, the solution quickly approaches a slow manifold on which

$$u \sim -\frac{\sin(t)}{k} - \frac{\cos(t)}{k^2} + \dots$$

A robust numerical method must accurately capture both of these phases and work effectively when the time step h is of the order of $1/|k|$. This test case allows us to evaluate how well our three order 2 and three order 4 methods handle the challenges posed by stiffness, particularly the balance between stability and accuracy when using relatively large time steps compared to the fastest time scale in the system. We examined the performance of the methods over the interval $[0, \pi/2]$ using various time steps ranging from $h = 0.0001$ to $h = 0.1$, comparing both the accuracy of the numerical solutions against the analytical solution and the computational efficiency of each method.

The results presented in Fig. 1 and Tables 2-3 highlight the distinct behaviors of the three order 2 and the three order 4 (respectively) numerical methods under analysis.

RK2 and RK4 demonstrate stability only for small step sizes, specifically for values of $h \leq 0.01$. However, for $h \geq 0.05$, the relative error grows uncontrollably. This behavior confirms a strict stability limit around $h = 0.01$, beyond which the method fails to provide reliable results. On the other hand, both exponential and Rosenbrock methods exhibit excellent stability across all tested step sizes. Even for larger ones, the methods maintain a reasonably small relative error. However, both exponential methods demonstrate higher accuracy than their Rosenbrock counterparts, and the difference becomes significant in the case of methods of order 4.

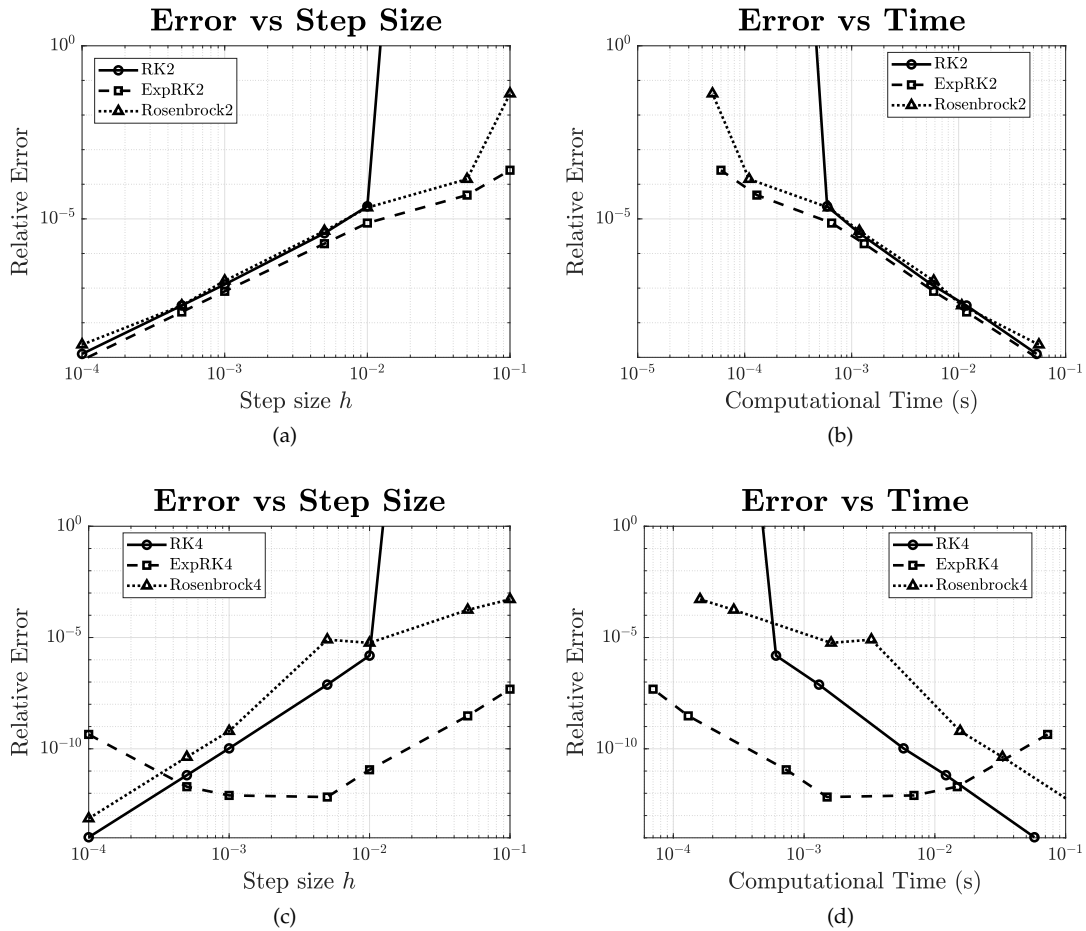


Figure 1: Relative error as a function of step size, showing how accuracy of each method of order 2 (a) and order 4 (c) evolves with different step choices and relative error as a function of computational time, highlighting the efficiency of methods of order 2 (b) and order 4 (d).

Regarding computational efficiency, differences in computation times among the methods of order 2 are negligible, making observations about performance insignificant. Concerning instead the methods of order 4, we remark that, while RK4 has the lowest computational cost per step, ExpRK4 requires the least computational time to reach a target error. It is worth noting that a relative error close to machine precision is already reached using $h = 0.005$, which makes the use of smaller step sizes not sensible for this example.

Final considerations. Considering both accuracy and efficiency, ExpRK2 emerges as the best-balanced method among those of order 2. It sustains high accuracy even with large step sizes, allowing for significantly larger steps compared to the

Table 2: Numerical methods performance for various step sizes (order 2 methods).

| h | RK2 Error | ExpRK2 Error | Rb2 Error | RK2 Time(s) | ExpRK2 Time(s) | Rb2 Time(s) |
|--------------------|-------------------------|--------------------------|-------------------------|-------------|----------------|-------------|
| 1×10^{-4} | 1.2551×10^{-9} | 8.3276×10^{-10} | 2.3212×10^{-9} | 0.05355 | 0.05829 | 0.05625 |
| 5×10^{-4} | 3.1432×10^{-8} | 2.0438×10^{-8} | 3.1446×10^{-8} | 0.01183 | 0.01188 | 0.01070 |
| 1×10^{-3} | 1.2779×10^{-7} | 8.0986×10^{-8} | 1.6274×10^{-7} | 0.00556 | 0.00584 | 0.00585 |
| 5×10^{-3} | 3.8530×10^{-6} | 1.9199×10^{-6} | 4.4691×10^{-6} | 0.00118 | 0.00131 | 0.00118 |
| 1×10^{-2} | 2.3082×10^{-5} | 7.5741×10^{-6} | 2.1139×10^{-5} | 0.00059 | 0.00065 | 0.00061 |
| 5×10^{-2} | 7.0246×10^{30} | 4.8629×10^{-5} | 1.4075×10^{-4} | 0.00010 | 0.00013 | 0.00011 |
| 1×10^{-1} | 2.9523×10^{27} | 2.5459×10^{-4} | 4.1347×10^{-2} | 0.00005 | 0.00006 | 0.00005 |

Table 3: Numerical methods performance for various step sizes (order 4 methods).

| h | RK4 Error | ExpRK4 Error | Rb4 Error | RK4 Time(s) | ExpRK4 Time(s) | Rb4 Time(s) |
|--------------------|--------------------------|--------------------------|--------------------------|-------------|----------------|-------------|
| 1×10^{-4} | 1.0583×10^{-14} | 4.4262×10^{-10} | 7.3560×10^{-14} | 0.05772 | 0.07283 | 0.17238 |
| 5×10^{-4} | 6.5026×10^{-12} | 1.9799×10^{-12} | 4.2579×10^{-11} | 0.01216 | 0.01484 | 0.03288 |
| 1×10^{-3} | 1.0440×10^{-10} | 8.0308×10^{-13} | 6.2264×10^{-10} | 0.00576 | 0.00692 | 0.01561 |
| 5×10^{-3} | 7.6433×10^{-8} | 6.7939×10^{-13} | 8.0284×10^{-6} | 0.00130 | 0.00150 | 0.00327 |
| 1×10^{-2} | 1.5387×10^{-6} | 1.1421×10^{-11} | 5.7421×10^{-6} | 0.00061 | 0.00073 | 0.00161 |
| 5×10^{-2} | 6.4073×10^{36} | 3.0104×10^{-9} | 1.7416×10^{-4} | 0.00011 | 0.00013 | 0.00029 |
| 1×10^{-1} | 5.8631×10^{40} | 4.7821×10^{-8} | 5.2099×10^{-4} | 0.00006 | 0.00007 | 0.00016 |

other methods. Although it has a slight computational overhead per step, the reduced number of required steps makes it more efficient overall.

RK2, in contrast, proves to be the least suitable method for this stiff problem, as it requires extremely small time steps to maintain stability and lacks competitiveness in both accuracy and computational efficiency.

These findings align with the conclusions drawn by Cox and Matthews in [20], confirming the superiority of exponential methods for solving stiff problems. In particular, ExpRK2 offers a significant advantage in terms of stability and accuracy, permitting the use of much larger step sizes than conventional methods. The superiority of ExpRK4 with respect to the tested methods of the same order is even more evident. For stiff problems such as the one studied here, characterized by a parameter $k = -100$, exponential methods represent the optimal choice among the three methods of the corresponding order, striking an ideal balance between accuracy, stability, and computational efficiency.

5.3 2D. Duffing oscillator

The system under consideration in this second example is the Duffing oscillator, a fundamental model in nonlinear dynamics describing a forced harmonic oscillator with a cubic nonlinearity in the restoring force. The governing differential equation is given by

$$u'' + \omega u + ku^3 = 0, \quad (5.7)$$

where u represents the position, ω is the natural frequency, and k is the nonlinear coefficient. To analyze its temporal evolution, we rewrite the system as a first-order autonomous system by introducing the state variable $v = u'$, leading to the phase-space formulation

$$\begin{cases} u' = v, \\ v' = -\omega u - ku^3. \end{cases} \quad (5.8)$$

This formulation allows us to study the system as a two-dimensional dynamical problem. For numerical integration, it is convenient to express this system in a matrix-vector form

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ -ku^3 \end{bmatrix}. \quad (5.9)$$

This matrix formulation highlights the linear and nonlinear contributions to the system's dynamics. The stiffness of the system becomes significant for large values of k , where the nonlinear term induces rapid variations in the solutions. This necessitates the use of numerical schemes capable of handling both slow dynamics and the sharp transitions associated with high-curvature regions in the phase space trajectories.

The results presented in Fig. 2 and Tables 4-5 provide a comprehensive analysis of six numerical methods of order 2 and 4 applied to the Duffing oscillator system¹⁴ integrated over the time interval $[0,20]$ with $\omega = 1$ and $k = 100$.

RK2 demonstrates severe stability limitations when applied to this problem. For step sizes $h \leq 0.001$, the method produces reasonable errors, but as h increases to 0.01, the relative error grows to approximately 2.97 and, most importantly, for larger step sizes ($h \geq 0.05$), the method fails entirely, producing NaN values. On the other hand, stability constraints on RK4 are not as evident. This suggests that the problem, with these parameters, can be considered only mildly stiff.

Rb2 exhibits significantly better stability characteristics than RK2: it maintains reasonable accuracy for step sizes up to $h = 0.01$, with errors remaining on the order of 10^0 .

¹⁴All relative errors are computed with respect to a high-precision numerical solution obtained using MATLAB's `ode15s` solver, as no analytical solution is available for this nonlinear system.

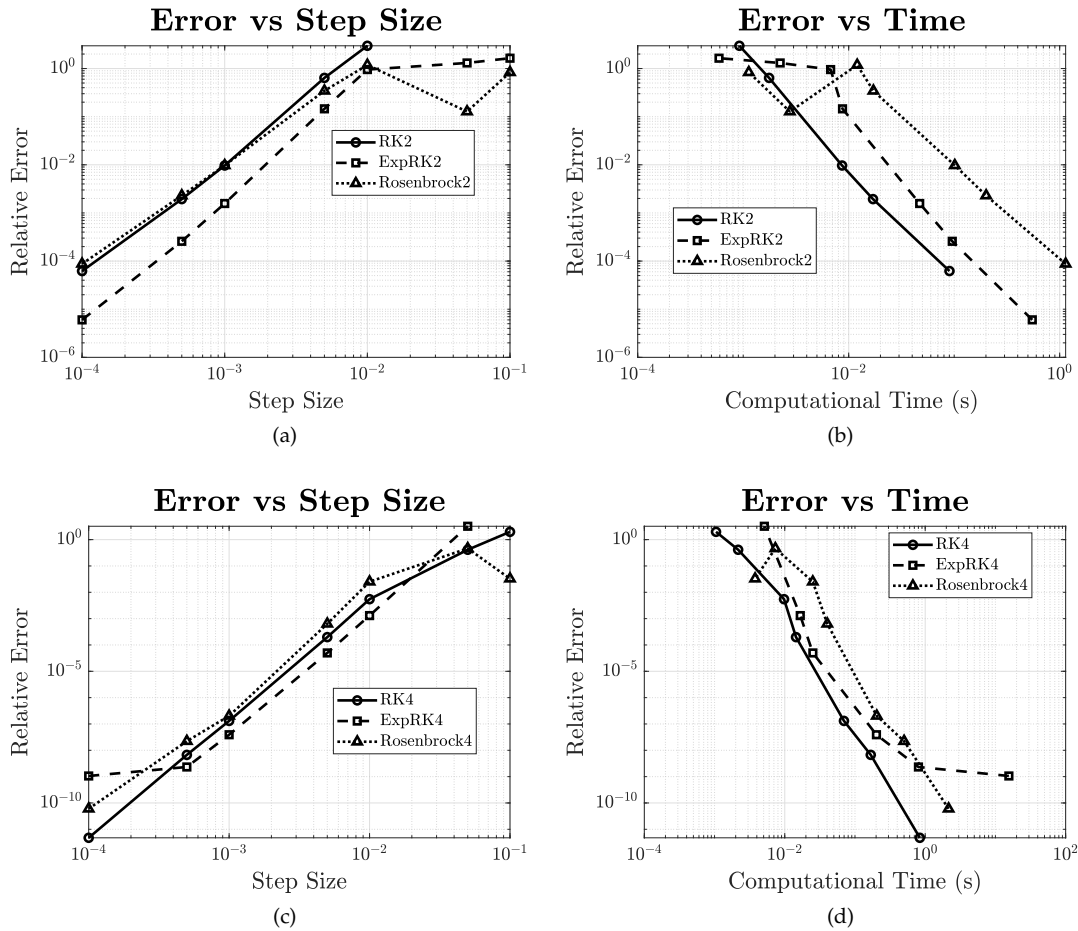


Figure 2: Relative error as a function of step size, showing how accuracy of each method of order 2 (a) and order 4 (c) evolves with different step choices and relative error as a function of computational time, highlighting the efficiency of methods of order 2 (b) and order 4 (d).

ExpRK2 demonstrates superior accuracy for most step sizes up to $h=0.01$, particularly excelling at moderate step sizes where it maintains errors significantly lower than the other methods. For $h = 5 \times 10^{-4}$ and $h = 1 \times 10^{-3}$, ExpRK2 shows errors approximately an order of magnitude smaller than both RK2 and Rb2, despite not being designed so as to capture stiffness phenomena caused by the nonlinear terms. However, for larger step sizes ($h \geq 0.05$), the errors increase substantially, exceeding the error of Rb2 at these larger step sizes. ExpRK4 exhibits similar behavior, in that it provides more accurate results for the intermediate step sizes. However, the difference in this case is not as significant.

Regarding computational efficiency, across all step sizes small enough for RK2 to obtain a solution, RK2 is the fastest method. ExpRK2 occupies an intermediate

Table 4: Numerical methods performance for various step sizes (order 2 methods).

| h | RK2 Error | ExpRK2 Error | Rb2 Error | RK2 Time(s) | ExpRK2 Time(s) | Rb2 Time(s) |
|--------------------|-------------------------|-------------------------|-------------------------|-------------|----------------|-------------|
| 1×10^{-4} | 6.2072×10^{-5} | 6.0105×10^{-6} | 8.7547×10^{-5} | 0.09026 | 0.54986 | 1.13880 |
| 5×10^{-4} | 1.9363×10^{-3} | 2.5767×10^{-4} | 2.2968×10^{-3} | 0.01701 | 0.09588 | 0.20080 |
| 1×10^{-3} | 9.6628×10^{-3} | 1.5697×10^{-3} | 9.7327×10^{-3} | 0.00866 | 0.04709 | 0.10134 |
| 5×10^{-3} | 6.3873×10^{-1} | 1.4548×10^{-1} | 3.4409×10^{-1} | 0.00176 | 0.00874 | 0.01715 |
| 1×10^{-2} | 2.9671×10^0 | 9.5176×10^{-1} | 1.1835×10^0 | 0.00092 | 0.00674 | 0.01204 |
| 5×10^{-2} | NaN | 1.3049×10^0 | 1.2752×10^{-1} | 0.00008 | 0.00224 | 0.00277 |
| 1×10^{-1} | NaN | 1.6391×10^0 | 8.3403×10^{-1} | 0.00002 | 0.00059 | 0.00113 |

Table 5: Numerical methods performance for various step sizes (order 4 methods).

| h | RK4 Error | ExpRK4 Error | Rb4 Error | RK4 Time(s) | ExpRK4 Time(s) | Rb4 Time(s) |
|--------------------|--------------------------|-------------------------|--------------------------|-------------|----------------|-------------|
| 1×10^{-4} | 4.7521×10^{-12} | 1.0584×10^{-9} | 6.0232×10^{-11} | 0.83861 | 15.54832 | 2.15542 |
| 5×10^{-4} | 6.7006×10^{-9} | 2.3246×10^{-9} | 2.2207×10^{-8} | 0.16769 | 0.80160 | 0.50081 |
| 1×10^{-3} | 1.3075×10^{-7} | 3.8983×10^{-8} | 2.0253×10^{-7} | 0.06966 | 0.20216 | 0.20446 |
| 5×10^{-3} | 1.9809×10^{-4} | 4.9790×10^{-5} | 6.3571×10^{-4} | 0.01447 | 0.02514 | 0.04033 |
| 1×10^{-2} | 5.4686×10^{-3} | 1.3051×10^{-3} | 2.5171×10^{-2} | 0.00977 | 0.01663 | 0.02508 |
| 5×10^{-2} | 4.1232×10^{-1} | 3.2182×10^0 | 4.6251×10^{-1} | 0.00217 | 0.00512 | 0.00734 |
| 1×10^{-1} | 1.9771×10^0 | NaN | 3.2802×10^{-2} | 0.00105 | 0.00300 | 0.00378 |

position, with computation times around 3 times higher than RK2 for a given (relative) error. Rb2 emerges as the most computationally intensive method, requiring instead around 10 times the time required by RK2 for a given error. Among the methods of order 4, the classical RK is the fastest again, but the exponential method ExpRK4 is faster than the Rosenbrock method of the same order for intermediate step sizes. In any case, the difference among the three methods in terms of performance is not as significant as within the methods of order 2.

Final considerations. The analysis of these numerical methods for the Duffing oscillator reveals important trade-offs between accuracy, stability, and computational efficiency. RK2, while computationally inexpensive, suffers from severe stability limitations that render it unsuitable for step sizes larger than 0.01. Rb2 is the most accurate for step sizes larger than 0.01, while it becomes less accurate than ExpRK2 for smaller step sizes. Its non-monotonic error behavior at larger step sizes warrants further investigation but could potentially be exploited in specific applications. As for the methods of order 4, the Rosenbrock one appears again to be the most accurate for larger step sizes, while ExpRK4 may be a better choice for applications with higher accuracy constraints.

For the Duffing oscillator with the given parameters, the exponential methods appear to offer the most favorable combination of stability, accuracy, and computational efficiency among the methods of the same order. They maintain acceptable accuracy even at large step sizes and does so with reasonable computational cost. This makes them particularly suitable for applications where a balance between accuracy and efficiency is required. However, in applications where low latency is of outmost importance, the Rosenbrock methods may be preferred for moderate step sizes where its superior accuracy is most evident.

These findings highlight the importance of considering the specific characteristics of the differential system when selecting a numerical method, as the relative performance of different methods can vary significantly depending on the nature and parameters of the problem being solved.

5.4 Ikeda DDE

Lastly, we consider the Ikeda DDE, which models a variety of passive optical systems subject to a delayed feedback [33]. The general form of the DDE reads

$$x'(t) = -\lambda x(t) + f(\mu, x(t-\tau))$$

with λ and μ positive parameters, and f a continuous function. We analyze the following Cauchy problem, obtained by choosing $f(\mu, y) = \mu(1 - \sin(y))$ and $\tau = \pi/2$:

$$\begin{cases} x'(t) = -\lambda x(t) + \mu \left(1 - \sin \left(x \left(t - \frac{\pi}{2} \right) \right) \right), \\ x(t) = \cos(t), \quad t \in \left[-\frac{\pi}{2}, 0 \right]. \end{cases} \quad (5.10)$$

In order to make (5.10) amenable to our analysis, we need to first discretize the DDE into a finite number M of ODEs. We opt in particular for the pseudospectral discretization technique [11]. In fact, the ODEs obtained using this approach are defined by a semilinear right-hand side with stiff linear part, regardless of the right-hand side of the DDE. The resulting non-linear term is defined by only one nonzero component, which corresponds to the non-linear term in the original right-hand side (see [3] for details on this approach).

For our experiments, we chose $M = 50$, $\lambda = 40$ and $\mu = 2$, and integrated the equation over the time interval $[0, 4]$. The dimension of the resulting ODE is thus much larger than the previous examples, yet smaller than many applications which involve the discretization of a PDE. The linear part is dominant but, again contrary to the majority of examples analyzed in the literature of exponential methods, it is not sparse.

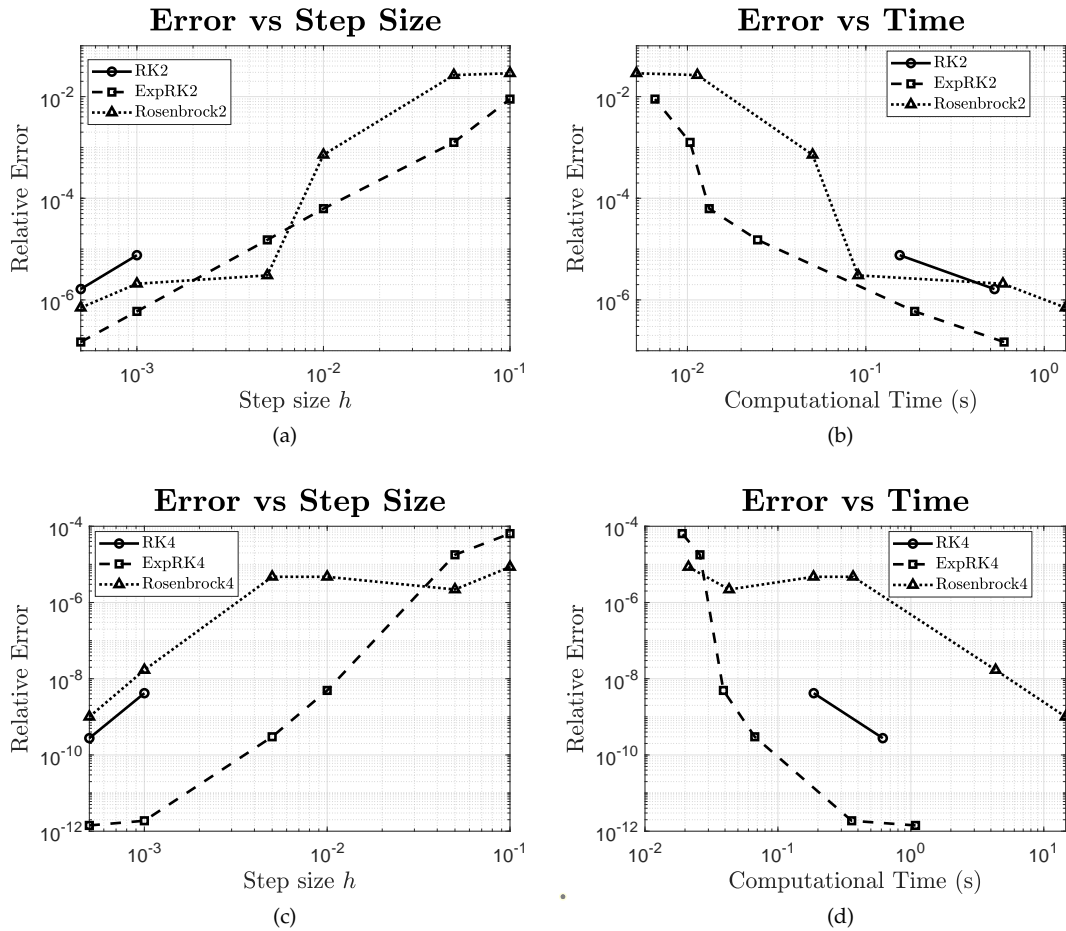


Figure 3: Relative error as a function of step size, showing how accuracy of each method of order 2 (a) and order 4 (b) evolves with different step choices and relative error as a function of computational time, highlighting the efficiency of methods of order 2 (c) and order 4 (d).

We remark that we analyze this test case as an example of a large stiff ODE, and that the investigation of the convergence of the solutions of the approximating ODE to the solutions of the original DDE is beyond the scopes of the survey. Therefore, we compute the error with respect to a reference solution of the approximating ODE obtained with `ode15s`.

Consistently with our previous test cases, we present our results in Fig. 3 and Tables 6-7.

The behavior of the classical RK methods is a clear confirmation of the overall stiffness of the problem. Indeed, not only are they affected by stability issues when used with larger step sizes, but their accuracy is consistently lower than the ExpRK methods of the same order by several order of magnitudes. The accuracy

Table 6: Numerical methods performance for various step sizes (order 2 methods).

| h | RK2 Error | ExpRK2 Error | Rb2 Error | RK2 Time(s) | ExpRK2 Time(s) | Rb2 Time(s) |
|--------------------|-------------------------|-------------------------|-------------------------|-------------|----------------|-------------|
| 5×10^{-4} | 1.6338×10^{-6} | 1.4835×10^{-7} | 7.0250×10^{-7} | 0.52600 | 0.59553 | 1.3176 |
| 1×10^{-3} | 7.5799×10^{-6} | 5.9468×10^{-7} | 2.0798×10^{-6} | 0.15484 | 0.18780 | 0.58667 |
| 5×10^{-3} | NaN | 1.5167×10^{-5} | 3.0350×10^{-6} | 0.01216 | 0.02473 | 0.09058 |
| 1×10^{-2} | NaN | 6.2445×10^{-5} | 7.1566×10^{-4} | 0.00534 | 0.01325 | 0.05029 |
| 5×10^{-2} | NaN | 1.2584×10^{-3} | 2.6518×10^{-2} | 0.00207 | 0.01035 | 0.01137 |
| 1×10^{-1} | NaN | 8.9412×10^{-3} | 2.8803×10^{-2} | 0.00050 | 0.00658 | 0.00517 |

Table 7: Numerical methods performance for various step sizes (order 4 methods).

| h | RK4 Error | ExpRK4 Error | Rb4 Error | RK4 Time(s) | ExpRK4 Time(s) | Rb4 Time(s) |
|--------------------|--------------------------|--------------------------|-------------------------|-------------|----------------|-------------|
| 5×10^{-4} | 2.7586×10^{-10} | 1.4164×10^{-12} | 1.0040×10^{-9} | 0.61501 | 1.0845 | 14.597 |
| 1×10^{-3} | 4.1724×10^{-9} | 1.8776×10^{-12} | 1.6948×10^{-8} | 0.18597 | 0.35891 | 4.3405 |
| 5×10^{-3} | NaN | 2.9999×10^{-10} | 4.7226×10^{-6} | 0.01883 | 0.06698 | 0.36765 |
| 1×10^{-2} | NaN | 4.9315×10^{-9} | 4.7182×10^{-6} | 0.00839 | 0.03890 | 0.18527 |
| 5×10^{-2} | NaN | 1.7857×10^{-5} | 2.1931×10^{-6} | 0.00261 | 0.02591 | 0.04295 |
| 1×10^{-1} | NaN | 6.4117×10^{-5} | 8.5589×10^{-6} | 0.00090 | 0.01906 | 0.02124 |

of the ExpRK methods is also, in both cases, much higher than the corresponding Rosenbrock methods, although the discrepancy is more evident among the methods of order 4, than of order 2.

The exponential methods are also the most advantageous in terms of efficiency, regardless of the order.

Final considerations. The effectiveness of ExpRK2 and ExpRK4 in this example should not come as a surprise, since the chosen value of μ results in a mildly stiff nonlinear part, whereas the linear term is the main responsible of the overall stiffness. In other words, the problem formulation makes the choice of exponential methods particularly well-suited despite the density of the matrix defining the linear term, which makes the computation of exponential somewhat more onerous.

6 Conclusions

In this review, we provided a comprehensive historical and theoretical account of the development of ExpRK methods over more than sixty years of research. We highlighted how the methods have evolved from their early formulations in

the 1960s to the state-of-the-art developments in stiff order conditions, efficient implementations, and novel applications to partial and delay differential equations. This dual historical and conceptual reconstruction not only situates ExpRK methods within the broader landscape of numerical integration techniques but also clarifies the reasons why they continue to attract substantial attention in both theory and practice.

At the same time, this survey was conceived with a pedagogical aim: to serve as an accessible entry point for readers encountering exponential integrators for the first time. To this end, we have complemented the historical and technical overview with a gradual introduction to the core concepts, supported by carefully selected didactic examples. Our numerical experiments were not intended to showcase the most advanced algorithms or optimized implementations, but rather to illustrate in a transparent and approachable way how ExpRK methods work in practice, and to provide newcomers with concrete tools to grasp their strengths and limitations. In this sense, the manuscript is both a review for specialists and a tutorial for learners. While we focused uniquely on explicit methods for our numerical section, to maintain a coherent presentation, a systematic investigation of their implicit exponential counterparts, together with the study of their robustness properties, remains promising direction for, potentially, a future work.

The dual nature of this contribution, at once a scholarly survey and a didactic introduction, represents, we believe, its distinctive value. On the one hand, specialists may find here a structured synthesis of the major milestones, including recent advances such as high stiff order methods, parallelization strategies, and applications in modeling contexts. On the other hand, students and non-experts may use the text as a stepping stone, moving from the basic principles to more sophisticated topics without losing sight of the underlying intuition. The coexistence of these two layers reflects the current state of the field itself, which remains both historically rich and dynamically evolving.

Looking ahead, it is clear that ExpRK methods will continue to play a central role in computational mathematics. Their capacity to treat stiff linear terms exactly, while retaining the flexibility and structure of RK schemes, ensures their relevance in a wide range of applications, from fundamental scientific modeling to large-scale simulations. Current research trends on stiff order conditions, energy preservation, and efficient implementations for large systems suggest that the methods are still far from reaching a definitive form. On the contrary, they remain a fertile ground for both theoretical inquiry and practical innovation. For these reasons, we expect ExpRK methods to maintain, and even strengthen, their role as a powerful paradigm for the numerical exploration of stiff problems.

Acknowledgements

All authors thank the reviewers for their insightful feedback, which has significantly helped improve the final version of this paper.

A.A is a member and acknowledges the support of Gruppo Nazionale per il Calcolo Scientifico of Istituto Nazionale di Alta Matematica. Her work was partially supported by the Italian Ministry of University and Research through the PRIN 2022 project (No. 20229P2HEA) “Stochastic numerical modelling for sustainable innovation”, Unit of Udine (CUP G53C24000710006).

N.C is a member and acknowledges the support of Gruppo Nazionale per l’Analisi Matematica, la Probabilità e le loro Applicazioni of Istituto Nazionale di Alta Matematica and moreover acknowledges the support of the MIUR-PRIN 2022 project “Nonlinear dispersive equations in presence of singularities” (No. 20225ATSTP).

M.S is a member and acknowledges the support of Gruppo Nazionale di Fisica Matematica of Istituto Nazionale di Alta Matematica.

References

- [1] A. H. Al-Mohy and N. J. Higham, *A new scaling and squaring algorithm for the matrix exponential*, SIAM J. Matrix Anal. Appl. 31 (2010), 970–989.
- [2] A. H. Al-Mohy and N. J. Higham, *Computing the action of the matrix exponential, with an application to exponential integrators*, SIAM J. Sci. Comput. 33 (2011), 488–511.
- [3] A. Andò and R. Vermiglio, *Exponential time integration for delay differential equations via pseudospectral discretization*, 18th IFAC Workshop on Time Delay Systems TDS 2024, IFAC-PapersOnLine 58 (2024), 190–195.
- [4] A. Andò and R. Vermiglio, *Exponential Runge-Kutta methods for delay equations in the sun-star abstract framework*, Discrete Contin. Dyn. Syst. Ser. B 30 (2025), 1842–1858.
- [5] A. Bellen and M. Zennaro, *Numerical Methods for Delay Differential Equations*, Numerical Mathematics and Scientific Computing series, Oxford University Press, 2003.
- [6] H. Berland, B. Owren, and B. Skaflestad, *B-series and order conditions for exponential integrators*, SIAM J. Numer. Anal. 43 (2005), 1715–1727.
- [7] H. Berland, B. Owren, and B. Skaflestad, *Solving the nonlinear Schrödinger equation using exponential integrators*, Model. Identif. Control 27 (2006), 201–217.
- [8] H. Berland, B. Skaflestad, and W. M. Wright, *EXPINT—A MATLAB package for exponential integrators*, ACM Trans. Math. Softw. 33 (2007).
- [9] C. Besse, G. Dujardin, and I. Lacroix-Violet, *High order exponential integrators for nonlinear Schrödinger equations with application to rotating Bose-Einstein condensates*, SIAM J. Numer. Anal. 55 (2017), 1387–1411.

- [10] G. Beylkin, J. M. Keiser, and L. Vozovoi, *A new class of time discretization schemes for the solution of nonlinear PDEs*, J. Comput. Phys. 147 (1998), 362–387.
- [11] D. Breda, O. Diekmann, M. Gyllenberg, F. Scarabel, and R. Vermiglio, *Pseudospectral discretization of nonlinear delay equations: New prospects for numerical bifurcation analysis*, SIAM J. Appl. Dyn. Syst. 15 (2016), 1–23.
- [12] J. C. Butcher, *Implicit Runge-Kutta processes*, Math. Comp. 18 (1964), 50–64.
- [13] J. C. Butcher, *A history of Runge-Kutta methods*, Appl. Numer. Math. 20 (1996), 247–260.
- [14] M. Caliari, F. Cassini, L. Einkemmer, and A. Ostermann, *Accelerating exponential integrators to efficiently solve semilinear advection-diffusion-reaction equations*, SIAM J. Sci. Comput. 46 (2024), A906–A928.
- [15] M. Caliari and A. Ostermann, *Implementation of exponential Rosenbrock-type integrators*, Appl. Numer. Math. 59 (2009), 568–581.
- [16] M. Caliari, M. Vianello, and L. Bergamaschi, *Interpolating discrete advection-diffusion propagators at Leja sequences*, J. Comput. Appl. Math. 172 (2004), 79–99.
- [17] J. Certaine, *The solution of ordinary differential equations with large time constants*, Mathematical Methods for Digital Computers, Wiley (1960), 128–132.
- [18] S. C. Chu and M. Berman, *An exponential method for the solution of systems of ordinary differential equations*, CACM 17 (1974), 699–702.
- [19] F. Costabile, *A survey of pseudo Runge-Kutta methods*, Rend. Mat. Roma 23 (2003), 217–234.
- [20] S. M. Cox and P. C. Matthews, *Exponential time differencing for stiff systems*, J. Comput. Phys. 176 (2002), 430–455.
- [21] C. F. Curtiss and J. O. Hirschfelder, *Integration of stiff equations*, Proc. Pak. Acad. Sci. A 38 (1952), 235–243.
- [22] G. Dahlquist, *A special stability problem for linear multistep methods*, BIT 3 (1963), 27–43.
- [23] G. Dahlquist, *Problems Related to the Numerical Treatment of Stiff Differential Equations*, in: Proceeding of the International Computing Symposium, North-Holland Publishing Company, 1973.
- [24] T. T. Dang and T. H. Hoang, *How to avoid order reduction in third-order exponential Runge-Kutta methods for problems with non-commutative operators?*, arXiv:2412.11920, 2024.
- [25] T. T. Dang and T. H. Hoang, *From memory model to CPU time: Exponential integrators for advection-dominated problems*, arXiv:2512.0367, 2025.
- [26] P. I. Davies and N. J. Higham, *A Schur-Parlett algorithm for computing matrix functions*, SIAM J. Matrix Anal. Appl. 25 (2003), 464–485.
- [27] O. Diekmann, P. Getto, and M. Gyllenberg, *Stability and bifurcation analysis of Volterra functional equations in the light of suns and stars*, SIAM J. Math. Anal. 39 (2008), 1023–1069.

- [28] O. Diekmann, S. A. van Gils, S. M. Verduyn Lunel, and H.-O. Walther, *Delay Equations – Functional, Complex and Nonlinear Analysis*, in: Applied Mathematical Sciences, Vol. 10, Springer Verlag, 1995.
- [29] G. Dimarco and L. Pareschi, *Exponential Runge-Kutta methods for stiff kinetic equations*, SIAM J. Numer. Anal. 49 (2011), 2057–2077.
- [30] G. Dujardin, *Exponential Runge-Kutta methods for the Schrödinger equation*, Appl. Numer. Math. 59 (2009), 1839–1857.
- [31] L. Einkemmer, T. H. Hoang, and A. Ostermann, *Should exponential integrators be used for advection-dominated problems?*, Adv. Appl. Math. Mech. 18 (2026), 815–832.
- [32] W. H. Enright, D. J. Higham, B. Owren, and P. W. Sharp, *A survey of the explicit Runge-Kutta method*.
- [33] T. Erneux, L. Larger, M. W. Lee, and J.-P. Goedgebuer, *Ikeda Hopf bifurcation revisited*, Phys. D 194 (2004), 49–64.
- [34] J. Fang and R. Zhan, *High order explicit exponential Runge-Kutta methods for semilinear delay differential equations*, J. Comput. Appl. Math. 388 (2021), Paper No. 113279.
- [35] E. Gallopoulos and Y. Saad, *On the parallel solution of parabolic equations*, in: Proceedings of the 3rd International Conference on Supercomputing, (1989), 17–28.
- [36] E. Gallopoulos and Y. Saad, *Efficient solution of parabolic equations by Krylov approximation methods*, SIAM J. Sci. Stat. Comput. 13 (1992), 1236–1264.
- [37] M. Golden, *An Explicit Sixth Order Runge-Kutta Method for Simple Lawson Integration*, 2025.
- [38] C. González, A. Ostermann, and M. Thalhammer, *A second-order Magnus-type integrator for nonautonomous parabolic problems*, J. Comput. Appl. Math. 189 (2006), 142–156.
- [39] K. G. Guderley and C.-C. Hsu, *A predictor-corrector method for a certain class of stiff differential equations*, Math. Comp. 26 (1972), 51–69.
- [40] E. Hairer and G. Wanner, *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, in: Computational Mathematics, Springer-Verlag 1991.
- [41] J. Hersch, *Contribution à la méthode des équations aux différences*, Z. Angew. Math. Phys. 9 (1958), 129–180.
- [42] N. J. Higham, *The scaling and squaring method for the matrix exponential revisited*, SIAM J. Matrix Anal. Appl. 26 (2005), 1179–1193.
- [43] N. J. Higham, *Functions of Matrices: Theory and Computation*, 2008.
- [44] N. J. Higham and A. H. Al-Mohy, *Computing matrix functions*, Acta Numer. 19 (2010), 159–208.
- [45] T. H. Hoang, *Order reduction of exponential Runge-Kutta methods: Non-commuting operators*, arXiv:2410.00470, 2024.
- [46] M. Hochbruck and C. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal. 34 (1997), 1911–1925.
- [47] M. Hochbruck, C. Lubich, and H. Selhofer, *Exponential integrators for large systems*

- of differential equations*, SIAM J. Sci. Comput. 19 (1998), 1552–1574.
- [48] M. Hochbruck and A. Ostermann, *Explicit exponential Runge-Kutta methods for semi-linear parabolic problems*, SIAM J. Numer. Anal. 43 (2005), 1069–1090.
- [49] M. Hochbruck and A. Ostermann, *Exponential Runge-Kutta methods for parabolic problems*, Appl. Numer. Math. 53 (2005), 323–339.
- [50] M. Hochbruck and A. Ostermann, *Exponential integrators*, Acta Numer. 19 (2010), 209–286.
- [51] M. Hochbruck and A. Ostermann, *Exponential multistep methods of Adams-type*, Bit Numer. Math. 51 (2011), 889–908.
- [52] M. Hochbruck, A. Ostermann, and J. Schweitzer, *Exponential Rosenbrock-type methods*, SIAM J. Numer. Anal. 47 (2009), 786–803.
- [53] Q. Huang, A. Ostermann, and G. Zhong, *Exponential Runge-Kutta methods of collocation type for parabolic equations with time-dependent delay* Bit Numer. Math. 66 (2026), Paper No. 4.
- [54] K. J. In't Hout, *A new interpolation procedure for adapting Runge-Kutta methods to delay differential equations*, BIT 32 (1992), 634–649.
- [55] L. Ji, H. Li, and A. Ostermann, *A low regularity exponential-type integrator for the derivative nonlinear Schrödinger equation*, arXiv:2601.20212, 2026.
- [56] C. Jiang, J.-C. Cui, X. Qian, and S. Song, *High-order linearly implicit structure-preserving exponential integrators for the nonlinear Schrödinger equation*, J. Sci. Comput. 90 (2022).
- [57] A.-K. Kassam and L. N. Trefethen, *Fourth-order time-stepping for stiff PDEs*, SIAM J. Sci. Comput. 26 (2005), 1214–1233.
- [58] S. Koikari, *Rooted tree analysis of Runge-Kutta methods with exact treatment of linear terms*, J. Comput. Appl. Math. 177 (2005), 427–453.
- [59] I. Kovacic and M. J. Brennan, *The Duffing Equation: Nonlinear Oscillators and Their Behaviour*, Wiley, 2011.
- [60] W. Kutta, *Beitrag zur näherungsweise integration totaler differentialgleichungen*, Z. Angew. Math. Phys. 46 (1901), 435–453.
- [61] J. Lambert, *Computational Methods in Ordinary Differential Equations*, Introductory Mathematics for Scientists and Engineers, Wiley, 1973.
- [62] J. Lambert, *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*, Wiley, 1991.
- [63] J. D. Lawson, *Generalized Runge-Kutta processes for stable systems with large Lipschitz constants*, SIAM J. Numer. Anal. 4 (1967), 372–380.
- [64] D. Li and X. Li, *Multiple relaxation exponential Runge-Kutta methods for the nonlinear Schrödinger equation*, SIAM J. Numer. Anal. 62 (2024), 2719–2744.
- [65] Y. Y. Lu, *Computing a matrix function for exponential integrators*, J. Comput. Appl. Math. 161 (2003), 203–216.
- [66] V. T. Luan, *Fourth-order two-stage explicit exponential integrators for time-dependent*

- PDEs*, Appl. Numer. Math. 112 (2017), 91–103.
- [67] V. T. Luan, *Efficient exponential Runge-Kutta methods of high order: Construction and implementation*, Bit Numer. Math. 61 (2021), 535–560.
- [68] V. T. Luan and T. Alhsmay, *Sixth-order exponential Runge-Kutta methods for stiff systems*, Appl. Math. Lett. 153 (2024), Paper No. 109036.
- [69] V. T. Luan, R. Chinomona, and D. R. Reynolds, *A new class of high-order methods for multirate differential equations*, SIAM J. Sci. Comput. 42 (2020), A1245–A1268.
- [70] V. T. Luan and A. Ostermann, *Exponential B-series: The stiff case*, SIAM J. Numer. Anal. 51 (2013), 3431–3445.
- [71] V. T. Luan and A. Ostermann, *Explicit exponential Runge-Kutta methods of high order for parabolic problems*, J. Comput. Appl. Math. 256 (2014), 168–179.
- [72] V. T. Luan and A. Ostermann, *Exponential Rosenbrock methods of order five-construction, analysis and numerical comparisons*, J. Comput. Appl. Math. 255 (2014), 417–431.
- [73] V. T. Luan and A. Ostermann, *Stiff order conditions for exponential Runge-Kutta methods of order five*, in: Modeling, Simulation and Optimization of Complex Processes-HPSC 2012: Proceedings of the Fifth International Conference on High Performance Scientific Computing, Springer (2014), 133–143.
- [74] V. T. Luan and A. Ostermann, *Parallel exponential Rosenbrock methods*, Comput. Math. Appl. 71 (2016), 1137–1150.
- [75] V. T. Luan, N. Van Hoang, and J. O. Ehigie, *Efficient exponential methods for genetic regulatory systems*, J. Comput. Appl. Math. 436 (2024), Paper No. 115424.
- [76] S. Maset and M. Zennaro, *Unconditional stability of explicit exponential Runge-Kutta methods for semi-linear ordinary differential equations*, Math. Comp. 78 (2009), 957–967.
- [77] S. Maset and M. Zennaro, *Stability properties of explicit exponential Runge-Kutta methods*, IMA J. Numer. Anal. 33 (2013), 111–135.
- [78] D. L. Michels, G. A. Sobottka, and A. G. Weber, *Exponential integrators for stiff elastodynamic problems*, ACM Trans. Graph. 33 (2014), 1–20.
- [79] B. V. Minchev, *Exponential Integrators for Semilinear Problems*, PhD Thesis, University of Bergen, 2004.
- [80] B. V. Minchev and W. Wright, *A review of exponential integrators for first order semi-linear problems*, Technical Report Preprint Numerics 2/2005, Norwegian University of Science and Technology, 2005.
- [81] W. L. Miranker, *Numerical methods for stiff equations and singular perturbation problems*, Mathematics and Its Applications, Vol. 5, Springer Dordrecht, 1981.
- [82] C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Rev. 45 (2003), 3–49.
- [83] H. Montanelli and N. Bootland, *Solving periodic semilinear stiff PDEs in 1D, 2D and 3D with exponential integrators*, Math. Comput. Simulation 178 (2020), 307–327.
- [84] S. P. Nørsett, *An A-stable modification of Adams-Bashforth's method*, in: Conference on

- the Numerical Solution of Differential Equations, Lecture Notes in Mathematics, Vol. 109, Springer, 1969.
- [85] B. N. Parlett, *Computation of Functions of Triangular Matrices*, Memorandum (University of California, Berkeley, Electronics Research Laboratory), Defense Technical Information Center, 1974.
- [86] D. A. Pope, *An exponential method of numerical integration of ordinary differential equations*, CACM 6 (1963), 491–493.
- [87] H. H. Rosenbrock, *Some general implicit processes for the numerical solution of differential equations*, Comput. J. 5 (1963), 329–330.
- [88] C. Runge, *Über die numerische auflösung von differentialgleichungen*, Math. Ann. 46 (1895), 167–178.
- [89] Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal. 29 (1992), 209–228.
- [90] A. D. Snider, *Error analysis for a stiff system procedure*, Math. Comp. 30 (1976), 216–219.
- [91] X. Teng and H. Zhang, *A third-order energy stable exponential-free Runge-Kutta framework for the nonlocal Cahn-Hilliard equation*, J. Sci. Comput. 103 (2025).
- [92] M. Tokman, *Efficient integration of large stiff systems of odes with exponential propagation iterative (epi) methods*, J. Comput. Phys. 213 (2006), 748–776.
- [93] L. Torelli, *Stability of numerical methods for delay differential equations*, J. Comput. Appl. Math. 25 (1989), 15–26.
- [94] H. Wang, J. Sun, H. Zhang, and X. Qian, *A novel up to fourth-order equilibria-preserving and energy-stable exponential Runge-Kutta framework for gradient flows*, CSIAM Trans. Appl. Math. 6 (2025), 106–147.
- [95] X. Wang, X. Zhao, and H.-L. Liao, *A unified framework on the original energy laws of three effective classes of Runge-Kutta methods for phase field crystal type models*, SIAM J. Numer. Anal. 63 (2025), 1808–1832.
- [96] Y. Xu, J. J. Zhao, and Z. N. Sui, *Stability analysis of exponential Runge-Kutta methods for delay differential equations*, Appl. Math. Lett. 24 (2011), 1089–1092.
- [97] H. Zhang, X. Qian, J. Xia, and S. Song, *Efficient inequality-preserving integrators for differential equations satisfying forward Euler conditions*, ESAIM: Math. Model. Numer. Anal. 57 (2023), 1619–1655.
- [98] H. Zhang, J. Yan, X. Qian, and S. Song, *Up to fourth-order unconditionally structure-preserving parametric single-step methods for semilinear parabolic equations*, Comput. Meth. Appl. Mech. Eng. 393 (2022), Paper No. 114817.
- [99] J. Zhao, R. Zhan, and A. Ostermann, *Stability analysis of explicit exponential integrators for delay differential equations*, Appl. Numer. Math. 109 (2016), 96–108.
- [100] J. Zhao, R. Zhan, and Y. Xu, *D-convergence and conditional GDN-stability of exponential Runge-Kutta methods for semilinear delay differential equations*, Appl. Math. Comput. 339 (2018), 45–58.