Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Efficient exponential Rosenbrock methods till order four

B. Cano [a,*,1], M.J. Moreta [b]

[a] *IMUVA, Departamento de Matemática Aplicada, Facultad de Ciencias, Universidad de Valladolid, Paseo de Belén 7, 47011 Valladolid, Spain*

[b] *IMUVA, Departamento de Análisis Económico y Economía Cuantitativa, Facultad de Ciencias Económicas y Empresariales, Universidad Complutense de Madrid, Campus de Somosaguas, Pozuelo de Alarcón, 28223 Madrid, Spain*

ARTICLE INFO

Keywords:
Exponential Rosenbrock methods
Nonlinear reaction–diffusion problems
Avoiding order reduction in time
Efficiency

ABSTRACT

In a previous paper, a technique was described to avoid order reduction with exponential Rosenbrock methods when integrating initial boundary value problems with time-dependent boundary conditions. That requires to calculate some information on the boundary from the given data. In the present paper we prove that, under some assumptions on the coefficients of the method which are mainly always satisfied, no numerical differentiation is required to approximate that information in order to achieve order 4 for parabolic problems with Dirichlet boundary conditions. With Robin/Neumann ones, just numerical differentiation in time may be necessary for order 4, but none for order ≤ 3.

Furthermore, as with this technique it is not necessary to impose any stiff order conditions, in search of efficiency, we recommend some methods of classical orders 2, 3 and 4 and we give some comparisons with several methods in the literature, with the corresponding stiff order.

1. Introduction

It is well known that stiff ordinary differential systems are typically those for which an explicit integration with standard methods is not possible due to a lack of A-stability [1]. When integrating in space partial differential equations, the space discretized system is stiff because the eigenvalues associated to the discretization of the spatial differentiation operator can be infinitely large in modulus when the space grid is refined.

Exponential methods have been developed in the literature in order to get a stable integration of stiff systems in an 'explicit' way [2]. Although exponential-type functions of matrices applied over vectors have to be calculated with these methods, the recent improvement of Krylov techniques to approximate them has made these methods a valuable tool to integrate some initial boundary value problems [3–5].

However, the phenomenon of order reduction which already turns up when integrating this type of problems with standard methods also turns up with exponential methods. More explicitly, when the boundary conditions are not periodic or do not satisfy enough conditions of annihilation on the boundary, the order of accuracy of the time integrators when integrating the space discretized system is smaller than the order of accuracy which is observed when integrating a nonstiff ODE. Some times convergence is even lost [6]. Because of that, restrictive stiff order conditions on the coefficients of exponential Runge–Kutta methods have been firstly suggested in the literature so as to achieve the desired order of accuracy when integrating semilinear parabolic problems with vanishing boundary conditions [7]. Later, another technique has been given in order to avoid that order reduction without having to impose restrictions on the coefficients [8–11]. Moreover, the latter technique is valid for time-dependent boundary conditions.

[*] Corresponding author.
E-mail addresses: bcano@uva.es (B. Cano), mjesusmoreta@ccee.ucm.es (M.J. Moreta).

[1] This research was supported by Junta de Castilla y León and Feder through project VA169P20

https://doi.org/10.1016/j.cam.2024.116158
Received 26 July 2023; Received in revised form 12 March 2024
Available online 23 July 2024

Furthermore, by assuming that the Jacobian of the vector field can be easily calculated, that extra information can be used with Rosenbrock methods so as to achieve a desired accuracy with less stages than their Runge–Kutta counterparts. Because of that, for the integration of reaction–diffusion initial boundary value problems, stiff order conditions have also been studied in [12] for exponential Rosenbrock methods. Moreover, many particular exponential Rosenbrock methods have been constructed satisfying those conditions in order to achieve a desired accuracy while trying to be as efficient as possible [3,12–16].

In contrast, in [17], a technique is suggested to avoid order reduction with any exponential Rosenbrock method of any classical order without imposing those stiff order conditions. For that, some intermediate initial boundary problems are considered for which the boundary values have to be calculated in terms of data. If analytic expressions for the latter are known, in [17] it is stated that those boundary values can be exactly calculated in order to get local order ≥2. However, in order to get local order ≥3, numerical differentiation is in general required either in space or in time. When that in space is necessary, a weak CFL condition is needed to prove convergence [17]. Although the latter condition is much less restrictive than that required when integrating with standard explicit Runge–Kutta methods, it would be better not to require it and not to resort to numerical differentiation for the ease of implementation.

Our aim in this paper is to prove that, under some simplifying assumptions (which are satisfied by mainly all already built exponential Rosenbrock methods), numerical differentiation in space is not required to achieve local order 3 and 4, so that no CFL condition is necessary then. Moreover, numerical differentiation in time is just necessary to achieve local order 4 with Robin/Neumann boundary conditions, but not with Dirichlet ones and that order and neither to achieve local order 3. We remark that, when integrating with standard Rosenbrock methods, a technique was also given in [18] to avoid order reduction and no numerical differentiation in space was either required to achieve local order ≤4. Therefore, the conclusions in this paper are similar to those in [18] for standard Rosenbrock methods. The advantage of using exponential Rosenbrock methods instead of standard ones correspond to problems where a good preconditioner is not available to solve linear systems in an efficient way and it is therefore more recommendable to use Krylov techniques to approximate exponentials of matrices applied over vectors [16,19–21].

In any case, coming back to the above considerations on exponential Rosenbrock methods, in this paper we also recommend some particular ones to get global orders 2, 3 and 4 (the latter for parabolic problems in which, by a summation-by-parts argument, the local order coincides with the global one). We remind that the local error corresponds to the error after just one step while the global error is the error after the required steps to get a final time. As no stiff order conditions have to be imposed, there are more parameters to play with in order to achieve a cheaper computation.

The paper is structured as follows. Section 2 gives some preliminaries and particularizes the obtained formulas for the full discretization in [17] in order to get local order $p+1$ for $p=1,2,3$ when the method has classical order at least p. Section 3 justifies how those formulas greatly simplify under the precise conditions on the coefficients of the method which are stated there. (That simplification must not be seen in the length of formulas, but in the fact that the required boundary values are easier to calculate and that the linear combination of φ_j-functions of matrices applied over vectors is given, so that Krylov subroutines can be directly applied.) Section 4 gives some theorems and remarks which justify that the simplifying assumptions are satisfied by basically every constructed method. Section 5 gives a recommendation for methods of order 2, 3 and 4 and, finally, Section 6 shows convergence tables and a numerical comparison in CPU time with other methods in the literature.

2. Preliminaries

We will assume that the problem to integrate can be written as

$$\begin{aligned} \dot{u}(t) &= Au(t)+\Psi(u(t))+h(t), \quad 0\le t\le T, \\ u(0) &= u_0, \\ \partial u(t) &= g(t), \end{aligned} \tag{1}$$

where $\cdot$ denotes differentiation with respect to time and where $A : D(A)\subset X\to X$ is a linear differential operator defined in a subset of the Banach space $X=L^\infty(\bar{\Omega},\mathbb{C})$, where Ω is a bounded domain in $\mathbb{R}^d$ and the supremum norm is considered. On the other hand, $\partial : X\to Y$ corresponds to a boundary operator which arrives at another Banach space $Y=L^\infty(\partial\bar{\Omega})$ with the same supremum norm. We will assume hypotheses (A1)–(A9) in [17] and, more particularly, if (1) is real and the solution stays in the interval I, that $\Psi\in C^2(I,\mathbb{R})$ and, if (1) is complex, that Ψ is holomorphic in a region where the solution stays. Moreover, we will assume that $h\in C^2([0,T],X)$.

As stated in [17,18] through [22–24], (A1)–(A4) guarantee that problem (1) is well-posed. However, more particular assumptions on the regularity of u and Ψ are stated in (A5)–(A9) and in Theorem 3.1 in [17] in order to get the desired local order $p+1$ whenever the method has non-stiff global order $\ge p$. We do not state them here for the sake of brevity, but it is there justified that all expressions which turn up in the rest of the paper exist. Another issue would be to justify in terms of the data of the problem that the exact solution is regular enough. Although that would be very interesting, it is not an aim of this paper. In any case, we refer to [25–28] for bounds of some derivatives of the exact solution in the linear case. (The last papers are focused on singularly perturbed problems, which are not necessarily the case here, but the bounds which are obtained there are also valid for not so small parameters.)

General Rosenbrock exponential methods are determined by some coefficients $c_1,\dots,c_s$ and some coefficient functions in a Butcher tableau

$$a_{i,j}(z) = \sum_{l=1}^{r}\lambda_{i,j,l}\varphi_l(c_i z),\ i=1,\dots,s,\ j=1,\dots,i-1,$$

$$b_i(z) = \sum_{l=1}^{r} \mu_{i,l}\varphi_l(z), \ i = 1, \dots, s, \tag{2}$$

where

$$\varphi_l(z) = \int_0^1 e^{(1-\theta)z} \frac{\theta^{l-1}}{(l-1)!} d\theta, \quad l \geq 1,$$

and where it is assumed that

$$\sum_{j=1}^{i-1} a_{ij}(0) = c_i. \tag{3}$$

For an autonomous ODE differential system of the type

$$\dot{U}(t) = F(U(t)), \tag{4}$$

the numerical solution U_{n+1} at time $t_{n+1} = t_n + k$ is given from the numerical solution U_n at time t_n through the following formulas

$$K_{n,i} = e^{c_i k J_n} U_n + k \sum_{j=1}^{i-1} a_{ij}(kJ_n) G_n(K_{n,j}), \quad i = 1, \dots, s, \tag{5}$$

$$U_{n+1} = e^{kJ_n} U_n + k \sum_{i=1}^{s} b_i(kJ_n) G_n(K_{n,i}), \tag{6}$$

where

$$J_n = F'(U_n), \quad G_n(U) = F(U) - J_n U. \tag{7}$$

When the problem is non-autonomous, by rewriting

$$\dot{U}(t) = F(t, U(t))$$

as an autonomous one, the method reads

$$\begin{aligned} K_{n,i} &= e^{c_i k J_n} U_n + c_i k t_n \varphi_1(c_i k J_n) V_n + k \sum_{j=1}^{i-1} \sum_{l=1}^{r} \lambda_{i,j,l} \Big[\varphi_l(c_i k J_n)[F(t_{n,j}, K_{n,j}) - t_{n,j} V_n - J_n K_{n,j}] + c_i k \varphi_{l+1}(c_i k J_n) V_n \Big], \\ U_{n+1} &= e^{kJ_n} U_n + k t_n \varphi_1(kJ_n) V_n + k \sum_{i=1}^{s} \sum_{l=1}^{r} \mu_{i,l} \Big[\varphi_l(kJ_n)[F(t_{n,i}, K_{n,i}) - t_{n,i} V_n - J_n K_{n,i}] + k \varphi_{l+1}(kJ_n) V_n \Big], \end{aligned} \tag{8}$$

where $t_{n,j} = t_n + c_j k$, and

$$V_n = \frac{\partial F}{\partial t}(t_n, U_n), \quad J_n = \frac{\partial F}{\partial U}(t_n, U_n).$$

On the other hand, we will consider a general space discretization for the differential operator A, such that, when applied to the elliptic problem

$$Au = F, \quad \partial u = g,$$

the nodal values on the grid are given by the solution $U_h \in \mathbb{C}^N$ of this system

$$A_{h,0} U_h + C_h g = P_h F + D_h \partial F,$$

where $A_{h,0}$ discretizes A restricted to Ker(∂), P_h is the nodal projection and $C_h, D_h : Y \to \mathbb{C}^N$ are other linear operators over functions on the boundary.

We will assume hypotheses (H1)–(H3) in [17] and will consider the matrix

$$J_{n,h,0} = A_{h,0} + \Psi'(U_h^n),$$

which discretizes the Jacobian with respect to U of the vector field which defines (1) at $t = t_n = nk$, where k is the timestepsize.

In [17], a technique is suggested to achieve local order $p+1$ with a method which has classical global order at least p. For the precise values $p = 1, 2, 3$, the modified exponential Rosenbrock method to achieve that goal when integrating the non-autonomous problem (1) reads as follows.

For $p = 1$,

$$\begin{aligned} K_{n,i,h} &= e^{c_i k J_{n,h,0}} U_h^n + c_i k \varphi_1(c_i k J_{n,h,0})[t_n P_h \dot{h}(t_n) + C_h \partial u(t_n)] \\ &\quad + k \sum_{j=1}^{i-1} \sum_{l=1}^{r} \lambda_{i,j,l} [\varphi_l(c_i k J_{n,h,0}) G_{n,j,h} + c_i k \varphi_{l+1}(c_i k J_{n,h,0}) P_h \dot{h}(t_n)], \\ U_h^{n+1} &= e^{k J_{n,h,0}} U_h^n + k \varphi_1(k J_{n,h,0})[t_n P_h \dot{h}(t_n) + C_h \partial u(t_n) - D_h \partial \bar{J}(t_n) u(t_n)] \\ &\quad + k^2 \varphi_2(k J_{n,h,0}) C_h \partial [\bar{J}(t_n) u(t_n) + t_n \dot{h}(t_n)] \end{aligned}$$

$$\begin{aligned}&+k\sum_{i=1}^{s}\sum_{l=1}^{r}\mu_{i,l}\big[\varphi_l(kJ_{n,h,0})G_{n,i,h}\\ &\qquad\qquad +k\varphi_{l+1}(kJ_{n,h,0})[P_h\dot{h}(t_n)+C_h\partial\hat{\hat{G}}_n]\big]\end{aligned} \tag{9}$$

where

$$G_{n,j,h}=\Psi(K_{n,j,h})+P_h h(t_{n,j})-t_{n,j}P_h\dot{h}(t_n)-\mathrm{diag}(\Psi'(U_h^n))K_{n,j,h},$$

with $t_{n,j}=t_n+c_jk$, and

$$\hat{\hat{G}}_n=\Psi(u(t_n))+h(t_n)-t_n\dot{h}(t_n)-\Psi'(u(t_n))u(t_n). \tag{10}$$

As for $p=2$, we get

$$\begin{aligned}K_{n,i,h}\ =\ & e^{c_ikJ_{n,h,0}}U_h^n\\ &+c_ik\varphi_1(c_ikJ_{n,h,0})[t_nP_h\dot{h}(t_n)+C_h\partial u(t_n)-D_h\partial\bar{J}(t_n)u(t_n)]\\ &+(c_ik)^2\varphi_2(c_ikJ_{n,h,0})C_h\partial[\bar{J}(t_n)u(t_n)+t_n\dot{h}(t_n)]\\ &+k\sum_{j=1}^{i-1}\sum_{l=1}^{r}\lambda_{i,j,l}\big[\varphi_l(c_ikJ_{n,h,0})G_{n,j,h}+c_ik\varphi_{l+1}(c_ikJ_{n,h,0})[P_h\dot{h}(t_n)+C_h\partial\hat{\hat{G}}_n]\big],\\ U_h^{n+1}\ =\ & e^{kJ_{n,h,0}}U_h^n+k\varphi_1(kJ_{n,h,0})[t_nP_h\dot{h}(t_n)+C_h\partial u(t_n)-D_h\partial\bar{J}(t_n)u(t_n)]\\ &+k^2\varphi_2(kJ_{n,h,0})\big[C_h\partial[\bar{J}(t_n)u(t_n)+t_n\dot{h}(t_n)]-D_h\partial[\bar{J}(t_n)^2u(t_n)+t_n\bar{J}(t_n)\dot{h}(t_n)]\big]\\ &+k^3\varphi_3(kJ_{n,h,0})C_h\partial[\bar{J}(t_n)^2u(t_n)+t_n\bar{J}(t_n)\dot{h}(t_n)]\\ &+k\sum_{i=1}^{s}\sum_{l=1}^{r}\mu_{i,l}\Big[\varphi_l(kJ_{n,h,0})G_{n,i,h}+k\varphi_{l+1}(kJ_{n,h,0})[P_h\dot{h}(t_n)+C_h\partial\hat{\hat{G}}_n-D_h\partial\bar{J}(t_n)\hat{\hat{G}}_n]\\ &\qquad\qquad +k^2\varphi_{l+2}(kJ_{n,h,0})C_h\partial[\bar{J}(t_n)\hat{\hat{G}}_n+\dot{h}(t_n)]\Big].\end{aligned} \tag{11}$$

Finally, for $p=3$, we have

$$\begin{aligned}K_{n,i,h}\ =\ & e^{c_ikJ_{n,h,0}}U_h^n\\ &+c_ik\varphi_1(c_ikJ_{n,h,0})[t_nP_h\dot{h}(t_n)+C_h\partial u(t_n)-D_h\partial\bar{J}(t_n)u(t_n)]\\ &+(c_ik)^2\varphi_2(c_ikJ_{n,h,0})\big[C_h\partial[\bar{J}(t_n)u(t_n)+t_n\dot{h}(t_n)]-D_h\partial[\bar{J}(t_n)^2u(t_n)+t_n\bar{J}(t_n)\dot{h}(t_n)]\big]\\ &+(c_ik)^3\varphi_3(c_ikJ_{n,h,0})C_h\partial[\bar{J}(t_n)^2u(t_n)+t_n\bar{J}(t_n)\dot{h}(t_n)]\\ &+k\sum_{j=1}^{i-1}\sum_{l=1}^{r}\lambda_{i,j,l}\big[\varphi_l(c_ikJ_{n,h,0})G_{n,j,h}+c_ik\varphi_{l+1}(c_ikJ_{n,h,0})[P_h\dot{h}(t_n)+C_h\partial\hat{\hat{G}}_n-D_h\partial\bar{J}(t_n)\hat{\hat{G}}_n]\\ &\qquad\qquad +(c_ik)^2\varphi_{l+2}(c_ikJ_{n,h,0})C_h\partial[\bar{J}(t_n)\hat{\hat{G}}_n+\dot{h}(t_n)]\big].\\ U_h^{n+1}\ =\ & e^{kJ_{n,h,0}}U_h^n+k\varphi_1(kJ_{n,h,0})[t_nP_h\dot{h}(t_n)+C_h\partial u(t_n)-D_h\partial\bar{J}(t_n)u(t_n)]\\ &+k^2\varphi_2(kJ_{n,h,0})\big[C_h\partial[\bar{J}(t_n)u(t_n)+t_n\dot{h}(t_n)]-D_h\partial[\bar{J}(t_n)^2u(t_n)+t_n\bar{J}(t_n)\dot{h}(t_n)]\big]\\ &+k^3\varphi_3(kJ_{n,h,0})\big[C_h\partial[\bar{J}(t_n)^2u(t_n)+t_n\bar{J}(t_n)\dot{h}(t_n)]-D_h\partial[\bar{J}(t_n)^3u(t_n)+t_n\bar{J}(t_n)^2\dot{h}(t_n)]\big]\\ &+k^4\varphi_4(kJ_{n,h,0})C_h\partial[\bar{J}(t_n)^3u(t_n)+t_n\bar{J}(t_n)^2\dot{h}(t_n)]\\ &+k\sum_{i=1}^{s}\sum_{l=1}^{r}\mu_{i,l}\Big[\varphi_l(kJ_{n,h,0})G_{n,i,h}+k\varphi_{l+1}(kJ_{n,h,0})[P_h\dot{h}(t_n)+C_h\partial\hat{\hat{\hat{G}}}_{n,i}-D_h\partial\bar{J}(t_n)\hat{\hat{G}}_n]\\ &\qquad\qquad +k^2\varphi_{l+2}(kJ_{n,h,0})\big[C_h\partial[\bar{J}(t_n)\hat{\hat{G}}_n+\dot{h}(t_n)]-D_h\partial[\bar{J}(t_n)^2\hat{\hat{G}}_n+\bar{J}(t_n)\dot{h}(t_n)]\big]\\ &\qquad\qquad +k^3\varphi_{l+3}(kJ_{n,h,0})C_h\partial[\bar{J}(t_n)^2\hat{\hat{G}}_n+\bar{J}(t_n)\dot{h}(t_n)]\Big].\end{aligned} \tag{12}$$

where

$$\hat{\hat{\hat{G}}}_{n,i}\ =\ \Psi(u(t_n))+h(t_n)-t_n\dot{h}(t_n)-\Psi'(u(t_n))u(t_n)+\ \frac{c_i^2k^2}{2}[\Psi''(u(t_n))\dot{u}(t_n)^2+\ddot{h}(t_n)].$$

Now we must see how to calculate the terms on the boundary taking into account that our data is just $\partial u(t)=g(t)$, $h(t)$ and u_0 in (1). As stated in [17], for $p\geq 2$, it is in principle necessary to resort to numerical differentiation either in space or in time for both Dirichlet and Robin/Neumann boundary conditions in order to approximate those boundary values. In order to avoid that as far as possible, we will see that many times some simplifications can be performed which allow to calculate the required boundaries exactly in terms of data.

3. Further simplifications and calculation of required boundaries in terms of data

In this section, we will see that, under the assumptions

$$\sum_{i=1}^{s} \mu_{i,1} = 1, \qquad \sum_{i=1}^{s} \mu_{i,l} = 0 \quad (l = 2,\dots,r), \tag{13}$$

$$\sum_{j=1}^{i-1} \lambda_{i,j,1} = c_i, \qquad \sum_{j=1}^{i-1} \lambda_{i,j,l} = 0 \quad (l = 2,\dots,r), \quad i = 1,\dots,s, \tag{14}$$

some terms in the general expressions (9), (11), (12) can be simplified. In a first place, the required terms on the boundary can be calculated in a much more direct way in terms of data, i.e. without resorting to numerical differentiation either in space nor in time, except for the case of Robin/Neumann boundary conditions and $p = 3$, for which numerical differentiation for the first derivative in time will be required. Secondly, some other terms not concerning the boundary also simplify under those assumptions and moreover, the evaluation of $G_{n,i,h}$ can be performed except for terms which lead to $O(k^{p+2})$-residues in U_h^{n+1} and therefore do not change either the local neither the global order. (This last simplification just concerns $p = 1$ for both the equations on the stages and the solution and $p = 2$ for the stages. It corresponds to the use of $\tilde{G}_{n,i,h}$ in the formulas below.) Moreover, we will gather together all terms in the same φ_j so that Krylov subroutines can be directly applied to calculate a linear combination of those matrix functions applied over the corresponding vectors.

3.1. $p = 1$

We notice that, in this case, for the stages, just the term $\partial u(t_n) = g(t_n)$ on the boundary is required (see (9)). Moreover, taking into account that

$$G_{n,j,h} = \Psi(K_{n,j,h}) + P_h h(t_n) - t_n P_h \dot{h}(t_n) - \mathrm{diag}(\Psi'(U_h^n))K_{n,j,h} + O(k^2), \tag{15}$$

and the first part of (14), the terms in $\varphi_1(c_i k J_{n,h,0})$ without boundaries can be simplified to

$$k\varphi_1(c_i k J_{n,h,0})[c_i P_h h(t_n) + \sum \lambda_{i,j,1}\tilde{G}_{n,j,h}],$$

where

$$\tilde{G}_{n,j,h} = \Psi(K_{n,j,h}) - \mathrm{diag}(\Psi'(U_h^n))K_{n,j,h}.$$

On the other hand, the term in $\varphi_2(c_i k J_{n,h,0})$, considering the second part of (14) is

$$k\varphi_2(c_i k J_{n,h,0})\big[\sum \lambda_{i,j,2}\tilde{G}_{n,j,h} + c_i^2 k P_h \dot{h}(t_n)\big].$$

Summing up,

$$\begin{aligned} K_{n,i,h} \;=\; & e^{c_i k J_{n,h,0}} U_h^n + k\varphi_1(c_i k J_{n,h,0})\big[c_i[P_h h(t_n) + C_h \partial u(t_n)] + \sum \lambda_{i,j,1}\tilde{G}_{n,j,h}\big] \\ & + k\varphi_2(c_i k J_{n,h,0})\big[\sum \lambda_{i,j,2}\tilde{G}_{n,j,h} + c_i^2 k P_h \dot{h}(t_n)\big] \\ & + k\sum_{l=3}^{r} \varphi_l(c_i k J_{n,h,0}) \sum \lambda_{i,j,l}\tilde{G}_{n,j,h}. \end{aligned}$$

As for U_h^{n+1} in (9), multiplying $k\varphi_1(kJ_{n,h,0})C_h$, just $\partial u(t_n) = g(t_n)$ turns up again; on the other hand, the term in $D_h \partial \bar{J}(t_n) u(t_n)$ can be written as

$$-k\varphi_1(kJ_{n,h,0})D_h\partial[\dot{u}(t_n) - \Psi(u(t_n)) + \Psi'(u(t_n))u(t_n) - h(t_n)]. \tag{16}$$

This term can be exactly calculated when considering Dirichlet boundary conditions and, with Robin/Neumann boundary conditions, it can be approximated through the numerical approximation at the boundary given by the space discretization of (1) itself, but without resorting to numerical differentiation. On the other hand, notice that, by using the first part of (13), some terms cancel and the term multiplying $k^2\varphi_2(kJ_{n,h,0})C_h$ is just $\partial\dot{u}(t_n)$. Therefore, that boundary can be calculated exactly in terms of data as $\dot{g}(t_n)$. As for the terms in $k^2\varphi_{l+1}(kJ_{n,h,0})C_h$ with $l \geq 2$, notice that they vanish in (9) because of the second part of (13). Moreover, considering (15) and the first part of (13), the terms in $\varphi_1(kJ_{n,h,0})$ without boundaries can be simplified to

$$k\varphi_1(kJ_{n,h,0})\big[P_h h(t_n) + \sum \mu_{i,1}\tilde{G}_{n,i,h}\big].$$

In a similar way, but using now also the second part of (13), the terms in $\varphi_2(kJ_{n,h,0})$ and $\varphi_l(kJ_{n,h,0})$ without boundaries, with $l \geq 3$, can be simplified to

$$\begin{aligned} & k\varphi_2(kJ_{n,h,0})\big[\sum \mu_{i,2}\tilde{G}_{n,i,h} + kP_h\dot{h}(t_n)\big] \\ & k\varphi_l(kJ_{n,h,0})\big[\sum \mu_{i,l}\tilde{G}_{n,i,h}\big], \quad l \geq 3. \end{aligned}$$

Summing up, under assumptions (13), U_h^{n+1} in (9) can be simplified to

$$U_h^{n+1} \;=\; e^{kJ_{n,h,0}}U_h^n + k\varphi_1(kJ_{n,h,0})\big[P_h h(t_n) + \sum \mu_{i,1}\tilde{G}_{n,i,h} + C_h\partial u(t_n) - D_h\partial[\dot{u}(t_n) - \Psi(u(t_n)) + \Psi'(u(t_n))u(t_n) - h(t_n)]\big]$$

$$+ k\varphi_2(kJ_{n,h,0})\big[\sum \mu_{i,2}\tilde{G}_{n,i,h} + k[P_h\dot{h}(t_n) + C_h\partial\dot{u}(t_n)]\big] + k\sum_{l=3}^{r}\varphi_l(kJ_{n,h,0})\sum \mu_{i,l}\tilde{G}_{n,i,h}.$$

3.2. $p = 2$

With similar arguments as those for $p = 1$, the stages for $p = 2$ in (11) can be simplified to the following formulas:

$$\begin{aligned}K_{n,i,h} &= e^{c_ikJ_{n,h,0}}U_h^n \\ &\quad + k\varphi_1(c_ikJ_{n,h,0})\Big[\sum \lambda_{i,j,1}\tilde{G}_{n,j,h} + c_i\big[P_hh(t_n) + C_h\partial u(t_n) - D_h\partial[\dot{u}(t_n) - \Psi(u(t_n)) + \Psi'(u(t_n))u(t_n) - h(t_n)]\big]\Big] \\ &\quad + k\varphi_2(c_ikJ_{n,h,0})\big[\sum \lambda_{i,j,2}\tilde{G}_{n,j,h} + c_i^2k[P_h\dot{h}(t_n) + C_h\partial\dot{u}(t_n)]\big] + k\sum_{l=3}^{r}\varphi_l(c_ikJ_{n,h,0})\sum \lambda_{i,j,l}\tilde{G}_{n,j,h}.\end{aligned}$$

As for U_h^{n+1} in (11), now the term multiplying $k^2\varphi_2(kJ_{n,h,0})C_h$ is again $\partial\dot{u}(t_n)$. As for $k^2\varphi_2(kJ_{n,h,0})D_h$, we have

$$\begin{aligned}&-\partial\big[\bar{J}(t_n)^2u(t_n) + \bar{J}(t_n)[\Psi(u(t_n)) + h(t_n) - \Psi'(u(t_n))u(t_n)]\big] \\ &\quad = -\partial\bar{J}(t_n)\dot{u}(t_n) = -\partial[\ddot{u}(t_n) - \dot{h}(t_n)],\end{aligned}$$

where, for the last equality, we have just considered the differentiation of Eq. (1). On the other hand, the term multiplying in $k^3\varphi_3(kJ_{n,h,0})C_h$ is

$$\begin{aligned}&\partial\big[\bar{J}(t_n)^2u(t_n) + \bar{J}(t_n)[\Psi(u(t_n)) + h(t_n) - \Psi'(u(t_n))u(t_n)] + \dot{h}(t_n)\big] \\ &\quad = \partial[\bar{J}(t_n)\dot{u}(t_n) + \dot{h}(t_n)] = \partial\ddot{u}(t_n),\end{aligned}$$

and the term in $\varphi_3(kJ_{n,h,0})D_h$ vanishes because $\sum \mu_{i,2} = 0$. In a similar way, the terms in $\varphi_l(kJ_{n,h,0})C_h$ and $\varphi_l(kJ_{n,h,0})D_h$ for $l \geq 4$ vanish because of the second part of (13). Considering this, U_h^{n+1} in (11) can be calculated as

$$\begin{aligned}U_h^{n+1} &= e^{kJ_{n,h,0}}U_h^n + k\varphi_1(kJ_{n,h,0})\ \big[\ \sum \mu_{i,1}\tilde{\tilde{G}}_{n,i,h} + C_h\partial u(t_n) - D_h\partial[\dot{u}(t_n) - \Psi(u(t_n)) + \Psi'(u(t_n))u(t_n) - h(t_n)]\ \big] \\ &\quad + k\varphi_2(kJ_{n,h,0})\big[\sum \mu_{i,2}\tilde{\tilde{G}}_{n,i,h} + k[P_h\dot{h}(t_n) + C_h\partial\dot{u}(t_n) - D_h\partial[\ddot{u}(t_n) - \dot{h}(t_n)]]\big] \\ &\quad + k\varphi_3(kJ_{n,h,0})[\sum \mu_{i,3}\tilde{\tilde{G}}_{n,i,h} + k^2C_h\partial\ddot{u}(t_n)] \\ &\quad + k\sum_{l=4}^{r}\varphi_l(kJ_{n,h,0})\sum \mu_{i,l}\tilde{\tilde{G}}_{n,i,h},\end{aligned}$$

where

$$\tilde{\tilde{G}}_{n,j,h} = \Psi(K_{n,j,h}) - \mathrm{diag}(\Psi'(U_h^n))K_{n,j,h} + P_hh(t_{n,j}) - c_jkP_h\dot{h}(t_n).$$

We notice that again a term like (16) turns up, which can either be calculated exactly or approximated without resorting to numerical differentiation. As for the other terms on the boundary, they can be calculated in terms of data with both Dirichlet and Robin/Neumann boundary conditions since $\partial\dot{u}(t) = \dot{g}(t)$ and $\partial\ddot{u}(t) = \ddot{g}(t)$.

3.3. $p = 3$

Similarly to the calculation of U_h^{n+1} with $p = 2$, but using (14), the stages in (12) can be simplified to

$$\begin{aligned}K_{n,i,h} &= e^{c_ikJ_{n,h,0}}U_h^n + k\varphi_1(c_ikJ_{n,h,0})\Big[\sum \lambda_{i,j,1}\tilde{\tilde{G}}_{n,j,h} + c_i\big[C_h\partial u(t_n) - D_h\partial[\dot{u}(t_n) - \Psi(u(t_n)) + \Psi'(u(t_n))u(t_n) - h(t_n)]\big]\Big] \\ &\quad + k\varphi_2(c_ikJ_{n,h,0})\Big[\sum \lambda_{i,j,2}\tilde{\tilde{G}}_{n,j,h} + c_i^2k\big[P_h\dot{h}(t_n) + C_h\partial\dot{u}(t_n) - D_h\partial[\ddot{u}(t_n) - \dot{h}(t_n)]\big]\Big] \\ &\quad + k\varphi_3(c_ikJ_{n,h,0})\big[\sum \lambda_{i,j,3}\tilde{\tilde{G}}_{n,j,h} + c_i^3k^2C_h\partial\ddot{u}(t_n)\big] \\ &\quad + k\sum_{l=4}^{r}\varphi_l(c_ikJ_{n,h,0})\sum \lambda_{i,j,l}\tilde{\tilde{G}}_{n,j,h}.\end{aligned} \tag{17}$$

As for the terms concerning boundaries to calculate U_h^{n+1} in (12), using the left part of (13) and (14), the term in $\varphi_2(kJ_{n,h,0})C_h\partial$ can be simplified to

$$k^2\varphi_2(kJ_{n,h,0})C_h\partial\left[\dot{u}(t_n) + \frac{k^2}{2}(\sum \mu_{i,1}c_i^2)[\Psi''(u(t_n))\dot{u}(t_n)^2 + \ddot{h}(t_n)]\right],$$

which can again be calculated exactly in terms of data with Dirichlet boundary conditions and can be approximated with Robin/Neumann ones considering the error from the approximation itself at the boundary and resorting to numerical differentiation just for the first time derivative $\dot{u}$.

On the other hand, the term in $\varphi_2(kJ_{n,h,0})D_h\partial$ can be simplified to

$$-k^2\varphi_2(kJ_{n,h,0})D_h\partial\big[\bar{J}(t_n)^2u(t_n) + \bar{J}(t_n)[\Psi(u(t_n)) - \Psi'(u(t_n))u(t_n) + h(t_n)]\big]$$

$$= -k^2\varphi_2(kJ_{n,h,0})D_h\partial\bar{J}(t_n)\dot{u}(t_n) = -k^2\varphi_2(kJ_{n,h,0})D_h\partial[\ddot{u}(t_n) - \dot{h}(t_n)],$$

which is exactly calculable in terms of data for both Dirichlet and Robin/Neumann boundary conditions.

As for the terms in $\varphi_3(kJ_{n,h,0})C_h\partial$, they can be written as

$$\begin{aligned}
&k^2\varphi_3(kJ_{n,h,0})C_h\partial\Big[\frac{k^2}{2}(\sum\mu_{i,2}c_i^2)[\Psi''(u(t_n))\dot{u}(t_n)^2 + \ddot{h}(t_n)] \\
&\qquad\qquad + k\big[\bar{J}(t_n)^2u(t_n) + \bar{J}(t_n)[\Psi(u(t_n)) - \Psi'(u(t_n))u(t_n) + h(t_n)] + \dot{h}(t_n)\big]\Big] \\
&= k^2\varphi_3(kJ_{n,h,0})C_h\partial\big[\frac{k^2}{2}(\sum\mu_{i,2}c_i^2)[\Psi''(u(t_n))\dot{u}(t_n)^2 + \ddot{h}(t_n)] + k[\bar{J}(t_n)\dot{u}(t_n) + \dot{h}(t_n)]\big] \\
&= k^2\varphi_3(kJ_{n,h,0})C_h\partial\big[\frac{k^2}{2}(\sum\mu_{i,2}c_i^2)[\Psi''(u(t_n))\dot{u}(t_n)^2 + \ddot{h}(t_n)] + k\ddot{u}(t_n)\big].
\end{aligned}$$

Similarly, the term in $\varphi_3(kJ_{n,h,0})D_h\partial$ can be written as

$$\begin{aligned}
&-k^3\varphi_3(kJ_{n,h,0})D_h\partial\big[\bar{J}(t_n)^3u(t_n) + \bar{J}(t_n)^2[\Psi(u(t_n)) + h(t_n) - \Psi'(u(t_n))u(t_n)] + \bar{J}(t_n)\dot{h}(t_n)\big] \\
&= -k^3\varphi_3(kJ_{n,h,0})D_h\partial\big[\bar{J}(t_n)[\bar{J}(t_n)\dot{u}(t_n) + \dot{h}(t_n)]\big] \\
&= -k^3\varphi_3(kJ_{n,h,0})D_h\partial\bar{J}(t_n)\ddot{u}(t_n) \\
&= -k^3\varphi_3(kJ_{n,h,0})D_h\partial[\dddot{u}(t_n) - \ddot{h}(t_n) - \Psi''(u(t_n))\dot{u}(t_n)^2],
\end{aligned}$$

where the last equality comes from differentiating (1) three times.

In a similar way, it can be deduced that the term in $\varphi_4(kJ_{n,h,0})C_h$ is

$$k^4\varphi_4(kJ_{n,h,0})C_h\partial\Big[\frac{1}{2}(\sum\mu_{i,3}c_i^2)[\Psi''(u(t_n))\dot{u}(t_n)^2 + \ddot{h}(t_n)] + \dddot{u}(t_n) - \ddot{h}(t_n) - \Psi''(u(t_n))\dot{u}(t_n)^2\Big]$$

but that in $\varphi_4(kJ_{n,h,0})D_h\partial$ vanishes, as well as the possible terms $\varphi_l(kJ_{n,h,0})C_h\partial$ and $\varphi_l(kJ_{n,h,0})D_h\partial$ for $l \geq 5$.

Summing up, under the assumptions in (14), U_h^{n+1} in (12) can be written as

$$\begin{aligned}
U_h^{n+1} &= e^{kJ_{n,h,0}}U_h^n + k\varphi_1(kJ_{n,h,0})\big[\sum\mu_{i,1}\tilde{\tilde{G}}_{n,i,h} + C_h\partial u(t_n) - D_h\partial[\dot{u}(t_n) - \Psi(u(t_n)) + \Psi'(u(t_n))u(t_n) - h(t_n)]\big] \\
&+ k\varphi_2(kJ_{n,h,0})\Big[\sum\mu_{i,2}\tilde{\tilde{G}}_{n,i,h} + k\big[P_h\dot{h}(t_n) + C_h\partial\big[\dot{u}(t_n) + \frac{k^2}{2}(\sum\mu_{i,1}c_i^2)[\Psi''(u(t_n))\dot{u}(t_n)^2 + \ddot{h}(t_n)]\big] - D_h\partial[\ddot{u}(t_n) - \dot{h}(t_n)]\big]\Big] \\
&+ k\varphi_3(kJ_{n,h,0})\Big[\sum\mu_{i,3}\tilde{\tilde{G}}_{n,i,h} + k^2\big[C_h\partial\big[\frac{k}{2}(\sum\mu_{i,2}c_i^2)[\Psi''(u(t_n))\dot{u}(t_n)^2 + \ddot{h}(t_n)] + \ddot{u}(t_n)\big] \\
&\qquad\qquad\qquad - D_h\partial[\dddot{u}(t_n) - \ddot{h}(t_n) - \Psi'(u(t_n))\dot{u}(t_n)^2]\big]\Big] \\
&+ k\varphi_4(kJ_{n,h,0})\Big[\sum\mu_{i,4}\tilde{\tilde{G}}_{n,i,h} + k^3C_h\partial\big[\dddot{u}(t_n) + (\frac{1}{2}\sum\mu_{i,3}c_i^2 - 1)[\Psi''(u(t_n))\dot{u}(t_n)^2 + \ddot{h}(t_n)]\big]\Big] \\
&+ k\sum_{l=5}^{r}\varphi_l(kJ_{n,h,0})\sum\mu_{i,l}\tilde{\tilde{G}}_{n,i,h}
\end{aligned} \tag{18}$$

Again all the terms at the boundary in this expression can be exactly calculated in terms of data with Dirichlet boundary conditions and in an approximated way with Robin/Neumann boundary ones taking into account the approximated values of the space discretization of (1) at the boundary and the approximation of $\dot{u}$ through numerical differentiation in time.

3.3.1. Concluding remarks

Remark 3.1. We notice that, in any case, no numerical differentiation in space is required to approximate the required boundary values, so that no weak CFL condition is required to prove the classical order of the method, as it was in principle necessary in the more general case [17].

Remark 3.2. Although, for the sake of brevity, we do not show the calculations here, for $p = 4$, numerical differentiation in space would be required to approximate boundary values with both Dirichlet and Robin/Neumann boundary conditions, even under assumptions (13)–(14).

4. Conditions under which the simplifying assumptions are satisfied

In this section, we will see why assumptions (13) and (14) are nearly always satisfied for methods of classical order ≤ 4. For that, we will consider the classical order conditions on the coefficients of the Rosenbrock method (2) and, from them, we will justify when (13) and (14) are guaranteed. Classical order conditions till order four were derived in [13], although just assuming that $s = 2$ and that $\sum b_i(z) = \varphi_1(z)$. In the general case, just considering Taylor expansions of (5)–(6) on the timestepsize, the fact that $G_n'(U_n) = 0$ because of (7), and comparing with the Taylor expansion of the exact solution of (4), the order conditions of Table 1 turn up.

Then, we have the following result:

Table 1
Classical order conditions for exponential Rosenbrock methods.

Order	Conditions			
1	$\sum b_i(0) = 1$			
2	$\sum b_i'(0) = \frac{1}{2}$			
3	$\sum b_i''(0) = \frac{1}{3}$	$\sum b_i(0)c_i^2 = \frac{1}{3}$		
4	$\sum b_i'''(0) = \frac{1}{4}$	$\sum b_i'(0)c_i^2 = \frac{1}{12}$	$\sum b_i(0)c_i a_{ij}'(0) = \frac{1}{8}$	$\sum b_i(0)c_i^3 = \frac{1}{4}$

Theorem 4.1. *If q denotes the classical order of an exponential Rosenbrock method ($1 \leq q \leq 4$) and r in (2) satisfies $r \leq q$, then (13) is satisfied.*

Proof. We firstly notice that, considering (2), the first column of conditions in Table 1 is equivalent to

$$
\begin{aligned}
\sum_{l=1}^{r} \frac{1}{l!} \sum_{i=1}^{s} \mu_{i,l} &= 1, \\
\sum_{l=1}^{r} \frac{1}{(l+1)!} \sum_{i=1}^{s} \mu_{i,l} &= \frac{1}{2}, \\
\sum_{l=1}^{r} \frac{1}{(l+2)!} \sum_{i=1}^{s} \mu_{i,l} &= \frac{1}{6}, \\
\sum_{l=1}^{r} \frac{1}{(l+3)!} \sum_{i=1}^{s} \mu_{i,l} &= \frac{1}{24}.
\end{aligned} \tag{19}
$$

Therefore, when $q = 1$, just the first equation must hold and, when $r = 1$, (13) follows directly.

When $q = 2$, the first two equations must hold. When $r = 1$, both equations are the same and again (13) follows immediately. When $r = 2$, we have a linear system of two equations in the two unknowns $\sum \mu_{i,1}$ and $\sum \mu_{i,2}$. The matrix associated to that system is clearly nonsingular. Because of that, there is a unique solution of that system, which obviously corresponds to $\sum \mu_{i,1} = 1$ and $\sum \mu_{i,2} = 0$.

When $q = 3$, the first three equations must hold. When $r \leq 2$, the first two of them lead to (13) with the same arguments than before and the corresponding solution happen to also satisfy the third equation. When $r = 3$, we have a linear system of three equations and three unknowns, which matrix is again non-singular and the unique solution of the system is $\sum \mu_{i,1} = 1$, $\sum \mu_{i,2} = 0$ and $\sum \mu_{i,3} = 0$.

For $q = 4$, a similar argument leads to the result. □

Let us see now under which conditions (14) is guaranteed.

Theorem 4.2.

(i) If $r = 1$ or $\lambda_{i,j,l} = 0$ for $l \geq 2$, (14) is always satisfied.
(ii) If $r = 2$ or $\lambda_{i,j,l} = 0$ for $l \geq 3$, $s = 2$ and $q = 4$, (14) is satisfied.

Proof. (i) comes directly from (3), which can be written like this considering (2)

$$
\sum_{l=1}^{r} \frac{1}{l!} \sum_{j=1}^{i-1} \lambda_{i,j,l} = c_i, \quad i = 1, \dots, s. \tag{20}
$$

In order to prove (ii), we notice that, for $q = 4$ and $s = 2$, the last two conditions in the last row of Table 1 read

$$
b_2(0)c_2 a_{21}'(0) = \frac{1}{8}, \quad b_2(0)c_2^3 = \frac{1}{4}.
$$

Considering (2) again, the latter can also be written as

$$
c_2^2 \Big(\sum_{l=1}^{r} \mu_{2,l} \frac{1}{l!}\Big)\Big(\sum_{l=1}^{r} \lambda_{2,1,l} \frac{1}{(l+1)!}\Big) = \frac{1}{8}, \quad c_2^3 \sum_{l=1}^{r} \mu_{2,l} \frac{1}{l!} = \frac{1}{4},
$$

which imply that

$$
\sum_{l=1}^{r} \lambda_{2,1,l} \frac{1}{(l+1)!} = \frac{c_2}{2}.
$$

Taking here $r = 2$ as well as in (20) with $i = 2$, a uniquely solvable linear systems of two equations and two unknowns turn up, which lead to $\lambda_{2,1,1} = c_2$ and $\lambda_{2,1,2} = 0$. □

Remark 4.3. We notice that the conditions which guarantee that the simplifying assumptions are satisfied mainly concern the maximum value r for the index l in the φ_l-functions, which must be small enough with respect to the classical order which wants to be achieved. This is not a serious drawback since, for the sake of simplicity, in the construction of methods, r is taken as small as possible. Only in case (ii) of Theorem 4.2 there is also a restriction for order $q = 4$ on the number of stages if φ_2 is turning up in the coefficients a_{ij}. However, this is not restrictive either since, as stated in the introduction, with Rosenbrock methods, very few stages are required to get a desired accuracy. In particular, classical order 4 can be obtained with just 2 stages.

Remark 4.4. We also remark that the simplifying assumptions (13)–(14) are equivalent to the simplifying assumptions in [12]

$$\sum_{i=1}^{s} b_i(z) = \varphi_1(z), \quad \sum_{j=1}^{i-1} a_{ij}(z) = c_i \varphi_1(c_i z), \quad 1 \le i \le s,$$

under which it was assured that equilibria of autonomous problems were preserved and which allowed to simplify stiff order conditions in that paper. More particularly, stiff order 2 was assured under those assumptions when integrating that type of problems when considering vanishing boundary conditions. Because of that, assumptions (13)–(14) are satisfied by mainly all already constructed methods. As distinct, in this paper, we justify through Theorems 4.1 and 4.2 that those assumptions are assured to be satisfied in many cases, without the need to resort to stiff order conditions of any kind or the preservation of equilibria.

5. Recommended methods depending on the desired accuracy

In this section, considering the results on the previous ones, we will suggest what we think is the best choice up to the moment of exponential Rosenbrock methods to achieve the particular orders of accuracy $q = 2, 3$ and 4.

5.1. $q = 2$

Rosenbrock-Euler method, which just has one stage and corresponds to $b_1(z) = \varphi_1(z)$, is well-known to have classical order 2. By looking at Table 1, we can see that not only the conditions for classical order 2 are satisfied, but also one of the conditions for classical order 3. Moreover, the other condition to achieve the latter accuracy cannot be satisfied with any method which just has one stage since, in such a case, because of (3), $c_1 = 0$. Therefore, this seems to be an unbeatable second-order method. What is more, it happens to satisfy (13) and (14). (In fact, that could also be deduced directly from Theorems 4.1 and 4.2.) Because of that, the technique in [17] with $p = 2$ can be applied to achieve local order 3 without resorting to numerical differentiation to calculate the required boundary values. (We notice that Euler-Rosenbrock method has stiff order 2 according to [12], but just shows local order 2 when implemented through the standard method of lines when the boundary condition $g(t)$ is time-dependent).

Numerical results in [17] show the big advantage in computational time of using the modified Rosenbrock-Euler method against Rosenbrock-Euler with the standard method of lines. The simplifications for that particular simple modified Rosenbrock-Euler method were already done in [17], where it was observed that the difference with the standard method of lines just consisted on adding a term of the form $k^3 \varphi_3(k J_{n,h,0}) C_h \partial \ddot{u}(t_n)$, when calculating U_h^{n+1} from U_h^n.

5.2. $q = 3$

As stated before, it is impossible to get an exponential Rosenbrock method of classical order 3 with just one stage. Because of that, we look for one with two stages. Trying to be as efficient as possible, we take $c_2 = 1$ so that possible evaluations at $t = t_n + c_2 k$ can also be used at the next step. The simpler function $a_{21}(z)$ of the form (2) satisfying (3) is then $a_{2,1}(z) = \varphi_1(z)$. If we now look for functions $b_1(z)$ and $b_2(z)$ satisfying the four necessary conditions in Table 1, we can see that we can achieve that just by considering $r = 1$. Although a linear system of four equations with two unknowns is obtained, three of them are equivalent and altogether lead to

$$b_1(z) = \frac{2}{3}\varphi_1(z), \quad b_2(z) = \frac{1}{3}\varphi_1(z).$$

This method again satisfies conditions (13) and (14), as it was also assured through Theorems 4.1 and 4.2. By considering $r = 2$, a one-parameter family of methods turn up, which correspond to

$$\mu_{1,1} = \frac{2}{3} + \frac{\mu_{2,2}}{2}, \quad \mu_{1,2} = -\mu_{2,2}, \quad \mu_{2,1} = \frac{1}{3} - \frac{\mu_{2,2}}{2}. \tag{21}$$

We notice that with all these methods, the first equation in the last row of Table 1 is satisfied. As for the third and fourth equation in the same row, they are never satisfied. However, the second equation in that row is just satisfied for $\mu_{2,2} = 1$, and that leads to

$$b_1(z) = \frac{7}{6}\varphi_1(z) - \varphi_2(z), \quad b_2(z) = -\frac{1}{6}\varphi_1(z) + \varphi_2(z).$$

We may therefore expect that this leads to the smallest local errors inside the family (21).

All previous methods have stiff order 2 but not stiff order 3 according to [12] since

$$b_1(z) + b_2(z) = \varphi_1(z), \quad a_{21}(z) = c_2 \varphi_1(c_2 z), \quad b_2(z) c_2^2 \neq 2\varphi_3(z).$$

However, the technique in [17] can be applied to avoid order reduction. Again the simplifying assumptions (13) and (14) are satisfied and therefore, the advantages of the simplified formulas in Section 3 can be used.

5.3. $q = 4$

As a method of classical order 4 to apply our technique to avoid order reduction, we have chosen *exprb42N* in [13], since it seems to be the best for mildly stiff problems in that paper. It is also a two-stage method where $c_2 = 3/4$ and the coefficients which determine the method are

$$a_{21}(z) = \frac{3}{4}\varphi_1(\frac{3}{4}z),\ b_1(z) = \frac{35}{27}\varphi_1(z) - \frac{48}{27}\varphi_2(z),\ b_2(z) = -\frac{8}{27}\varphi_1(z) + \frac{48}{27}\varphi_2(z).$$

Theorems 4.1 and 4.2 can be applied, so that the simplifications in Section 3 can be performed. With our technique to avoid order reduction using $p = 3$, we manage to get order 4 although the method just has stiff order 2 according to [12].

6. Numerical comparisons with other methods

Our aim in this section is not to show an exhaustive comparison of the recommended methods (implemented without order reduction as suggested in Section 3) against others in the literature (implemented through the standard method of lines). Our aim is not either to improve the techniques to calculate the linear combination of exponential matrix functions over some vectors which turn up in all methods. Our aim is just to defend that the technique well described in Section 3 must be taken into account if one wants to integrate accurately and efficiently initial boundary value problems of the type (1) with exponential Rosenbrock methods.

For that, we have chosen a particular example of (1) with time-dependent boundary conditions, which satisfies the hypotheses in [17] so that the technique there can be applied to avoid order reduction and which also satisfies the hypotheses in [12] so that a method satisfying the corresponding stiff order conditions also lead to the classical order of the method (at least for vanishing boundary conditions). We have also considered a particular space discretization which also satisfies hypotheses (H1)–(H2) in [17], as justified there. The problem is

$$\begin{aligned} u_t(x,t) &= u_{xx}(x,t) + u^2(x,t) + h(x,t), \\ u(x,0) &= u_0(x), \\ u(0,t) &= g_0(t), \quad u(1,t) = g_1(t), \end{aligned} \tag{22}$$

where h, u_0, g_0 and g_1 are taken so that the exact solution of the problem is $u(x,t) = \cos(x+t)$. As for the space discretization of the second derivative in space, we have considered the standard second-order symmetric difference scheme, which leads to

$$A_{h,0} = \frac{1}{h^2}\text{tridiag}(1,-2,1), \quad C_h[g_0, g_1] = \frac{1}{h^2}[g_0(t), \dots, g_1(t)], \quad D_h \equiv 0.$$

We have integrated till time $T = 1$ with a fixed value of $h = 1/1000$, for which the error in space can be considered negligible when taking as timestepsizes $k = 1/5, 1/10, 1/20, \dots$

6.1. $q = 3$

For the methods suggested in Section 5.2, which corresponds to Butcher tableaux

$$\begin{array}{c|cc} 0 & & \\ 1 & \varphi_1 & \\ \hline & \frac{2}{3}\varphi_1 & \frac{1}{3}\varphi_1 \end{array}, \qquad \begin{array}{c|cc} 0 & & \\ 1 & \varphi_1 & \\ \hline & \frac{7}{6}\varphi_1 - \varphi_2 & \varphi_2 - \frac{1}{6}\varphi_1 \end{array}, \tag{23}$$

we consider the technique described in Section 3 to avoid order reduction with $p = 3$, so that local order 4 and global order 3 are achieved according to [17]. We will refer to them as methods M1 and M2 respectively. In this case, the final formulas (17) for the stages reduce to this for both methods

$$\begin{aligned} K_{n,1,h} &= U_h^n, \\ K_{n,2,h} &= e^{kJ_{n,h,0}}U_h^n + k\varphi_1(kJ_{n,h,0})[\tilde{\tilde{G}}_{n,1,h} + C_h\partial u(t_n)] \\ &\quad + k^2\varphi_2(kJ_{n,h,0})[P_h\dot{h}(t_n) + C_h\partial\dot{u}(t_n)] + k^3\varphi_3(kJ_{n,h,0})C_h\partial\ddot{u}(t_n), \end{aligned} \tag{24}$$

and the final formula (18) to this for method M1

$$\begin{aligned} U_h^{n+1} &= e^{kJ_{n,h,0}}U_h^n + k\varphi_1(kJ_{n,h,0})[\frac{1}{3}(2\tilde{\tilde{G}}_{n,1,h} + \tilde{\tilde{G}}_{n,2,h}) + C_h\partial u(t_n)] \\ &\quad + k^2\varphi_2(kJ_{n,h,0})\big[P_h\dot{h}(t_n) + C_h\partial[\dot{u}(t_n) + \frac{k^2}{6}[\Psi''(u(t_n))\dot{u}(t_n)^2 + \ddot{h}(t_n)]]\big] \\ &\quad + k^3\varphi_3(kJ_{n,h,0})C_h\partial\ddot{u}(t_n) \\ &\quad + k^4\varphi_4(kJ_{n,h,0})C_h\partial[\dddot{u}\ (t_n) - \Psi''(u(t_n))\dot{u}(t_n)^2 - \ddot{h}(t_n)] \end{aligned} \tag{25}$$

and to this for method M2

$$U_h^{n+1} = e^{kJ_{n,h,0}}U_h^n + k\varphi_1(kJ_{n,h,0})[\frac{7}{6}\tilde{\tilde{G}}_{n,1,h} - \frac{1}{6}\tilde{\tilde{G}}_{n,2,h} + C_h\partial u(t_n)]$$

Table 2
Global errors when integrating (22) with nonvanishing boundary conditions, using method (27) and method (28) with the standard method of lines, $h = 1/1000$.

k	1/5	1/10	1/20	1/40	1/80
Method (27)	1.3868e−4	1.7442e−5	2.0928e−6	2.5018e−7	3.0287e−8
Order		2.9912	3.0590	3.0644	3.0462
Method (28)	3.1444e−4	3.8446e−5	4.7354e−6	5.8548e−7	7.2448e−8
Order		3.0319	3.0213	3.0158	3.0146

$$\begin{aligned}&+k\varphi_2(kJ_{n,h,0})\Big[-\tilde{\tilde{G}}_{n,1,h}+\tilde{\tilde{G}}_{n,2,h}+k\big[P_h\dot{h}(t_n)+C_h\partial[\dot{u}(t_n)-\frac{k^2}{12}[\Psi''(u(t_n))\dot{u}(t_n)^2+\ddot{h}(t_n)]]\big]\Big]\\&+k^3\varphi_3(kJ_{n,h,0})C_h\partial\big[\frac{k}{2}[\Psi''(u(t_n))\dot{u}(t_n)^2+\ddot{h}(t_n)]+\ddot{u}(t_n)\big]\\&+k^4\varphi_4(kJ_{n,h,0})C_h\partial[\dddot{u}\,(t_n)-\Psi''(u(t_n))\dot{u}(t_n)^2-\ddot{h}(t_n)].\end{aligned}\tag{26}$$

We will compare these methods with the standard method of lines implementation of the main method in *exprb32* [12], which is also a two-stage method of classical order 3, but which shows stiff order 3. For that method, it also happens that $c_2 = 1$ and the Butcher tableau is

$$\begin{array}{c|cc}0 & & \\ 1 & \varphi_1 & \\ \hline & \varphi_1-2\varphi_3 & 2\varphi_3\end{array}.\tag{27}$$

In fact, this is the only two-stage method of stiff order 3 with $c_2 = 1$. The standard method of lines implementation of this method consists of applying (8) to

$$\dot{U}_h(t)=A_{h,0}U_h(t)+\Psi(U_h(t))+P_hh(t)+C_h\partial u(t),$$

and leads to

$$\begin{aligned}K_{n,1,h}&=U_h^n,\\K_{n,2,h}&=e^{kJ_{n,h,0}}U_h^n+k\varphi_1(kJ_{n,h,0})[\tilde{\tilde{G}}_{n,1,h}+C_h\partial u(t_n)]\\&\quad+k^2\varphi_2(kJ_{n,h,0})[P_h\dot{h}(t_n)+C_h\partial\dot{u}(t_n)],\\U_h^{n+1}&=e^{kJ_{n,h,0}}U_h^n+k\varphi_1(kJ_{n,h,0})[\tilde{\tilde{G}}_{n,1,h}+C_h\partial u(t_n)]\\&\quad+k^2\varphi_2(kJ_{n,h,0})[P_h\dot{h}(t_n)+C_h\partial\dot{u}(t_n)]\\&\quad+2k\varphi_3(kJ_{n,h,0})\big[-\tilde{\tilde{G}}_{n,1,h}+\tilde{\tilde{G}}_{n,2,h}+C_h\partial[u(t_{n+1})-u(t_n)-k\dot{u}(t_n)]\big].\end{aligned}$$

We notice that, although with this method, we just arrive at φ_3 and with the previous ones to φ_4, what is multiplying φ_3 and φ_4 in (25) and (26) is just information on the boundary, which is very cheap to calculate. This method shows order 3 when integrating (22) in spite of the fact that non-vanishing boundary conditions are considered (See Table 2).

A similar implementation with the standard method of lines can be done with the main method in *exprb42* [13] given by

$$\begin{array}{c|cc}0 & & \\ \frac{3}{4} & \frac{3}{4}\varphi_{1,2} & \\ \hline & \varphi_1-\frac{32}{9}\varphi_3 & \frac{32}{9}\varphi_3\end{array},\tag{28}$$

where $\varphi_{i,j}(z)$ denotes $\varphi_i(c_jz)$. In spite of having $c_2 \neq 1$, this method has classical order 4. Its stiff order is also 4 according to [12], but that just applies to problems with vanishing boundary conditions. With non-vanishing boundary conditions, it just shows order 3 (See Table 2).

We have implemented all methods in Matlab using directly the subroutine phipm.m in [5] but rewriting in all formulas $e^{\alpha kJ_{n,h,0}}U_h^n$ as

$$U_h^n+\alpha kJ_{n,h,0}\varphi_1(\alpha kJ_{n,h,0})U_h^n$$

and thus incorporating a term $\alpha J_{n,h,0}U_h^n$ in the term multiplying $\varphi_1(\alpha kJ_{n,h,0})$. This was recommended in [12] and later used in [13–16], although we do not think this implementation is the best when h is very small.

As we can see in Fig. 1, both methods in (23), implemented through formulas (24)–(25)–(26) behave very similarly. The global order 3 is recovered in both cases (see Tables 4 and 6) and the computational cost is better than method (28) for all the considered values of k and better than (27) for global errors less than 10^{-6}. We also remark the great improvement in computational cost with the technique being described in this paper from $p = 2$ to $p = 3$. (We remind that $p = 2$ also leads to global order 3 in parabolic problems, but just local order 3 [17], see Tables 3 and 5.) In spite of adding more terms in the formulas after full discretization, the computational cost for a fixed stepsize decreases. An explanation for that is given in [29].

We also remind that all boundary values in formulas (24)–(25)–(26) can be exactly calculated in terms of data if an analytic expression is known for them.

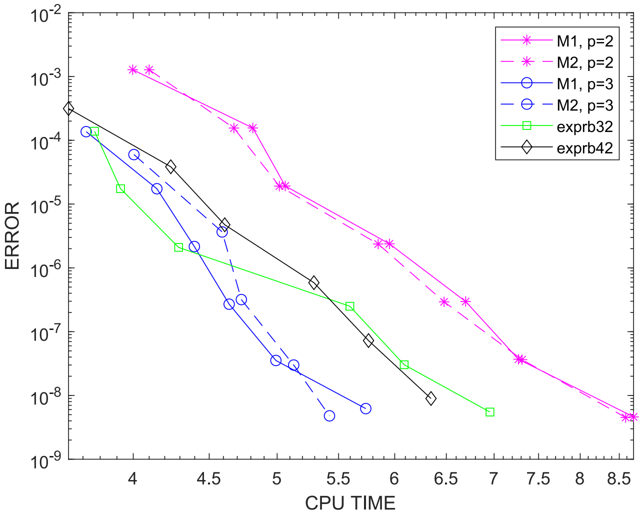

Fig. 1. Error against CPU time when integrating problem (22) with nonvanishing boundary conditions, using the first exponential Rosenbrock method in (23) (continuous line) with the suggested technique corresponding to $p = 2$ (magenta, asterisk) and $p = 3$ (blue, circle); the second exponential Rosenbrock method in (23) (discontinuous line) with the suggested technique corresponding to $p = 2$ (magenta, asterisk) and $p = 3$ (blue, circle); method (27) (green, square) and method (28) (black, diamond) with the standard method of lines.

Table 3
Local and global error when integrating (22) with nonvanishing boundary conditions, using the first exponential Rosenbrock method (M1) in (23) with the suggested technique corresponding to $p = 2$, $h = 1/1000$.

k	1/5	1/10	1/20	1/40	1/80
Local error	1.1536e−3	1.4197e−4	1.7585e−5	2.1850e−6	2.7179e−7
Order		3.0225	3.0132	3.0086	3.0071
Global error	1.2786e−3	1.5602e−4	1.9220e−5	2.3832e−6	2.9659e−7
Order		3.0348	3.0210	3.0116	3.0064

Table 4
Local and global error when integrating (22) with nonvanishing boundary conditions, using the first exponential Rosenbrock method (M1) in (23) with the suggested technique corresponding to $p = 3$, $h = 1/1000$.

k	1/5	1/10	1/20	1/40	1/80
Local error	6.6350e−5	4.1489e−6	2.5927e−7	2.0669e−8	2.1560e−9
Order		3.9993	4.0002	3.6489	3.2611
Global error	1.3650e−4	1.7394e−5	2.1603e−6	2.6970e−7	3.5321e−8
Order		2.9722	3.0093	3.0018	2.9328

Table 5
Local and global error when integrating (22) with nonvanishing boundary conditions, using the second exponential Rosenbrock method (M2) in (23) with the suggested technique corresponding to $p = 2$, $h = 1/1000$.

k	1/5	1/10	1/20	1/40	1/80
Local error	1.1519e−3	1.4171e−4	1.7540e−5	2.1770e−6	2.7038e−7
Order		3.0230	3.0142	3.0102	3.0093
Global error	1.2761e−3	1.5552e−4	1.9125e−5	2.3657e−6	2.9341e−7
Order		3.0365	3.0236	3.0152	3.0113

Table 6
Local and global error when integrating (22) with nonvanishing boundary conditions, using the second exponential Rosenbrock method (M2) in (23) with the suggested technique corresponding to $p = 3$, $h = 1/1000$.

k	1/5	1/10	1/20	1/40	1/80
Local error	6.6527e−5	4.1505e−6	2.5861e−7	1.6082e−8	1.1963e−9
Order		4.0026	4.0045	4.0073	3.7487
Global error	5.9715e−5	3.6612e−6	3.1846e−7	2.9967e−8	4.7893e−9
Order		4.0277	3.5231	3.4097	2.6455

Table 7
Global errors when integrating (22) with nonvanishing boundary conditions, using methods (30), (31), EPIRK4s3 and EPIEK4s3A with the standard method of lines, $h = 1/1000$.

k	1/5	1/10	1/20	1/40
Method (30)	4.3607e−5	2.9478e−6	1.9171e−7	1.1496e−8
Order		3.8869	3.9426	4.0597
Method (31)	1.5573e−5	8.7267e−7	5.1437e−8	3.8709e−9
Order		4.1575	4.0846	3.7321
Method EPIRK4s3	3.9766e−5	2.4662e−6	1.5065e−7	8.9167e−9
Order		4.0112	4.0330	4.0785
Method EPIRK4s3A	4.5137e−5	2.9593e−6	1.8858e−7	1.1213e−8
Order		3.9310	3.9720	4.0719

6.2. $q = 4$

For the method suggested in Section 5.3, which will be denoted by M3,

$$\begin{array}{c|cc} 0 & & \\ \frac{3}{4} & \frac{3}{4}\varphi_{1,2} & \\ \hline & \frac{35}{27}\varphi_1 - \frac{48}{27}\varphi_2 & -\frac{8}{27}\varphi_1 + \frac{48}{27}\varphi_2 \end{array}, \qquad (29)$$

we have again considered the technique in Section 3 with $p = 3$. In such a way, we get both local and global order 4. We have again considered phipm.m in [5] for the implementation of the corresponding results, writing the stages and U_h^{n+1} as linear combinations of $\{\varphi_l(\alpha J_{n,h,0})\}_{l\geq 1}$, as stated before.

In order to compare with methods which show order 4 when integrating (1) through the standard method of lines, we have chosen the main method in *pexprb43* in [15] since it seems to be the best among 4th-order methods in that paper. It is a 3-stage method in which the second and third stage can be calculated in parallel since $a_{32}(z) = 0$. More particularly, the method is

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2}\varphi_1(\frac{1}{2}\cdot) & & \\ 1 & \varphi_1 & & \\ \hline & \varphi_1 - 14\varphi_3 + 36\varphi_4 & 16\varphi_3 - 48\varphi_4 & -2\varphi_3 + 12\varphi_4 \end{array}. \qquad (30)$$

We have implemented this method in parallel using the subroutines phipm.m (as in [15]) and also with a horizontal/vertical implementation using the subroutine *phipm_simul_iom2* in [16]. It happens to show order 4 in our problem, as it can be observed in Table 7.

On the other hand, we have also chosen for the comparison the best among 5th-order methods in [14]. It is called *exprb53s3* there together with its embedded one and it happens to show just order 4 for problems where the boundary condition $g(t)$ is time-dependent, as it can be observed in Table 7. It has 3 stages and its tableau is

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2}\varphi_1(\frac{1}{2}\cdot) & & \\ \frac{9}{10} & \frac{9}{10}\varphi_{1,3} - \frac{27}{25}\varphi_{3,2} - \frac{729}{125}\varphi_{3,3} & \frac{27}{25}\varphi_{3,2} + \frac{729}{125}\varphi_{3,3} & \\ \hline & \varphi_1 - \frac{1208}{81}\varphi_3 + \frac{1120}{27}\varphi_4 & 18\varphi_3 - 60\varphi_4 & -\frac{250}{81}\varphi_3 + \frac{500}{27}\varphi_4 \end{array}. \qquad (31)$$

We have implemented the method with *phipm.m* subroutine directly and also using the subroutine *phipm_simul_iom2* in [16].

Finally, we have also considered for the comparisons the stiffly accurate 4th-order integrators EPIRK4s3 and EPIRK4s3 A in [3]. These methods are especially designed for autonomous problems, and special Krylov subroutines (called KIOPS) are designed for them so as to take profit of the fact that the φ_j matrix functions in the stages are applied over the same vector. We have then rewritten our problem (22) as an autonomous problem introducing the time variable into the equations and then we have applied the subroutines in [3] for the particular methods already mentioned. These methods also happen to show order 4 in our problem, as Table 7 shows. The results in terms of computational time turn up in Fig. 2, where we can see the comparison between all the methods and different Krylov subroutines. It is clearly observed that the recommended method (29) with the implementation corresponding to $p = 3$ in Section 3 is the most efficient of all methods for all values of the timestepsize. (Its fourth order can be also checked in Table 8.)

Moreover, we again remind that no numerical differentiation has been required to be used for the calculation of the necessary boundary values with this technique.

Table 8
Local and global error when integrating (22) with nonvanishing boundary conditions, using exponential Rosenbrock method (M3) in (29) with the suggested technique corresponding to $p = 3$, $h = 1/1000$.

k	1/5	1/10	1/20	1/40
Local error	6.6319e−5	4.1433e−6	2.5846e−7	1.6092e−8
Order		4.0006	4.0028	4.0055
Global error	4.4211e−5	2.5046e−6	1.4784e−7	8.9314e−9
Order		4.1418	4.0825	4.0490

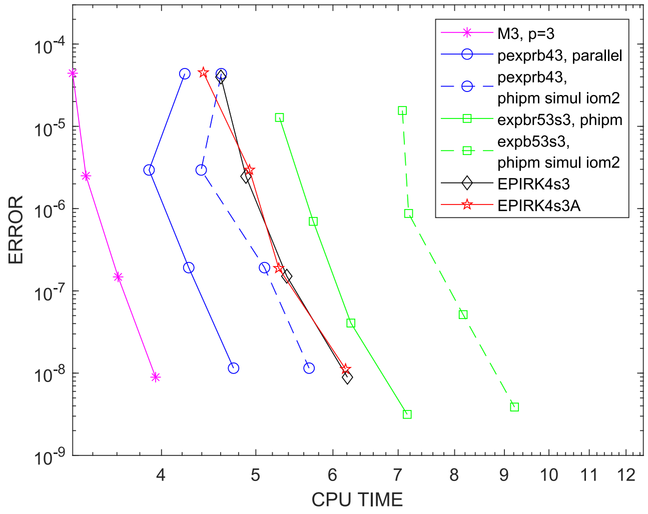

Fig. 2. Error against CPU time when integrating problem (22) with nonvanishing boundary conditions, using the exponential Rosenbrock method (29) (continuous line) with the suggested technique corresponding to $p = 3$ (magenta, asterisk); method (30) in parallel (blue, circle, continuous line) and using subroutine *phipm_simul_iom2* (blue, circle, discontinuous line); method (31) with phipm.m subroutine (green, square, continuous line) and using subroutine *phipm_simul_iom2* (green, square, discontinuous line); method EPIRK4s3 (black, diamond) and method EPIRK4s3 A (red, star).

Data availability

The codes used in this paper are available online in [30].

Acknowledgments

We are very much grateful to M. Tokman for providing us the subroutines corresponding to EPIRK4s3 and EPIRK4s3A.

References

[1] J.D. Lambert, Numerical Methods for Ordinary Differential Systems: The Initial Value Problem, Wiley, 1992.
[2] M. Hochbruck, A. Ostermann, Exponential integrators, Acta Numer. (2010) 209–286.
[3] S. Gaudreault, G. Rainwater, M. Tokman, KIOPS: A fast adaptive Krylov subspace solver for exponential integrators, J. Comput. Phys. 372 (2018) 236–255.
[4] M. Hochbruck, C. Lubich, On Krylov subspace approximations to the matrix exponential, SIAM J. Numer. Anal. 34 (5) (1997) 1911–1925.
[5] J. Niesen, W.M. Wright, Algorithm 919: a Krylov subspace algorithm for evaluating the φ-functions appearing in exponential integrators, ACM Trans. Math. Software 38 (3) (2012) 22.
[6] B. Cano, N. Reguera, CMMSE: Analysis of order reduction when lawson methods integrate nonlinear initial boundary value problems, Math. Methods Appl. Sci. (2022) 11319–11330, http://dx.doi.org/10.1002/mma.8451.
[7] M. Hochbruck, A. Ostermann, Explicit exponential Runge–Kutta methods for semilinear parabolic problems, SIAM J. Numer. Anal. 43 (2005) 1069–1090.
[8] B. Cano, M.J. Moreta, Exponential quadrature rules without order reduction for integrating linear initial boundary value problems, SIAM J. Numer. Anal. 56-3 (2018) 1187–1209.
[9] B. Cano, M.J. Moreta, Solving reaction–diffusion problems with explicit Runge–Kutta exponential methods without order reduction, ESAIM Math. Model. Numer. Anal. 58 (2024) 1053–1085.
[10] B. Cano, M.J. Moreta, Simplified explicit exponential Runge–Kutta methods without order reduction, 2024, to be published in Journal of Computational Mathematics.
[11] B. Cano, N. Reguera, How to avoid order reduction when lawson methods integrate nonlinear initial boundary value problems, BIT 62 (2022) 431–463.
[12] M. Hochbruck, A. Ostermann, J. Schweitzer, Exponential Rosenbrock-type methods, SIAM J. Numer. Anal. 47 (1) (2009) 786–803.
[13] V.T. Luan, Fourth-order two-stage explicit exponential integrators for time-dependent PDEs, Appl. Numer. Math. 112 (2017) 91–103.
[14] V.T. Luan, A. Ostermann, Exponential Rosenbrock methods of order five-construction, analysis and numerical comparisons, J. Comput. Appl. Math. 255 (2014) 417–431.

[15] V.T. Luan, A. Ostermann, Parallel exponential Rosenbrock methods, Comput. Math. Appl. 71 (2016) 1137–1150.

[16] V.T. Luan, J.A. Pudykiewicz, D.R. Reynolds, Further development of efficient and accurate time integration schemes for meteorological models, J. Comput. Phys. 376 (2019) 817–837.

[17] B. Cano, M.J. Moreta, Exponential rosenbrock methods without order reduction when integrating nonlinear initial boundary value problems, 2023, submitted for publication. https://arxiv.org/pdf/2307.12722.pdf.

[18] I. Alonso-Mallo, B. Cano, Efficient time integration of nonlinear partial differential equations by means of Rosenbrock methods, Mathematics 9 (2021) 1970, http://dx.doi.org/10.3390/math9161970.

[19] M.A. Gondal, Exponential rosenbrock integrators for option pricing, J. Comput. Appl. Math. 234 (4) (2010) 1153–1160.

[20] V.T. Luan, D.L. Michels, Exponential rosenbrock methods and their application in visual computing, in: T. Jax, A. Bartel, M. Ehrhardt, M. Günther, G. Steinebach (Eds.), Invited Chapter in Book ''Rosenbrock-Wanner-Type Methods: Theory and Applications'', Springer, 2021.

[21] M. Tokman, Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods, J. Comput. Phys. 213 (2006) 748–776.

[22] I. Alonso-Mallo, C. Palencia, On the convolutions operators arising in the study of abstract initial boundary value problems, Proc. R. Soc. Edinb. A 126A (1996) 515–539.

[23] D. Henry, Geometric Theory of Semilinear Parabolic Problems, in: Springer Lectures Notes in Mathematics, vol. 840, Springer, Berlin, Germany, 1981.

[24] C. Palencia, I. Alonso-Mallo, Abstract initial boundary values problems, Proc. R. Soc. Edinb. A 124A (1994) 879–908.

[25] C. Clavero, J.C. Jorge, A splitting uniformly convergent method for one-dimensional parabolic singularly perturbed convection–diffusion systems, Appl. Numer. Math. 183 (2023) 317–332.

[26] J.L. Gracia, F.J. Lisbona, E.O'. Riordan, A coupled system of singularly perturbed parabolic reaction–diffusion equations, Adv. Comput. Math. 32 (1) (2010) 43–61.

[27] R. Ishwariya, J.J. Miller, S. Valarmathi, Parameter uniform essentially first order convergence of a fitted mesh method for a class of parabolic singularly perturbed Robin problem for a system of reaction–diffusion equations, Int. J. Biomath. 12 (01) (2019) 19950001.

[28] S. Matthews, E. O'Riordan, G.I. Shishkin, A numerical method for a system of singularly perturbed reaction–diffusion equations, J. Comput. Appl. Math. 145 (2002) 151–166.

[29] B. Cano, N. Reguera, Why improving the accuracy of exponential integrators can decrease their computational cost? Mathematics 9 (2021) 1008, http://dx.doi.org/10.3390/math9091008.

[30] M.J. Moreta, B. Cano, CPU exponential Rosenbrock methods, Mendeley Data V1 (2024) http://dx.doi.org/10.17632/svhvwtxtzh.1.