SECOND-ORDER ROSENBROCK-EXPONENTIAL (ROSEXP) METHODS FOR PARTITIONED DIFFERENTIAL EQUATIONS [*]

VALENTIN DALLERIT [†], TOMMASO BUVOLI [‡], MAYYA TOKMAN [†], AND STÉPHANE GAUDREAULT [§]

Abstract. In this paper, we introduce a new framework for deriving partitioned implicit-exponential integrators for stiff systems of ordinary differential equations and construct several time integrators of this type. The new approach is suited for solving systems of equations where the forcing term is comprised of several additive nonlinear terms. We analyze the accuracy and stability of the new integrators and compare their performance with existing schemes for such systems using several numerical examples. We also propose a novel approach to visualizing the linear stability of the partitioned schemes, which provides a more intuitive way to understand and compare the stability properties of various schemes. Our new integrators are A-stable, 2^{nd} order methods that require only one call to the linear system solver and one exponential-like matrix function evaluation per time step. In addition to comparing the new integrators to previously proposed schemes, our numerical experiments validate the convergence and efficiency of the new methods.

Key words. Implicit–Exponential, Stiff differential equations, Exponential integrators, Rosenbrock method,

MSC codes. 65L20, 65L04, 65L05

1. Introduction.

Many scientific and engineering problems involve dynamics driven by several processes of different natures. Often such systems are modeled by differential evolution equations with a forcing term that is comprised of several additive components. These additive terms can represent the influence of each of the driving mechanisms. A well-known example of such a system is an advection-diffusion equation where the evolution is governed by the advective and diffusive forces modeled by two additive terms with first-order and second-order derivatives, respectively. In general, a two-term forcing model can be written as an initial-value problem of the form

$$y' = f_1(y) + f_2(y) \tag{1.1a}$$

$$y(t_0) = y_0. \tag{1.1b}$$

Frequently, the forcing terms $f_i(y)$ represent processes occurring over a wide range of temporal scales. As a result, the differential equations modeling such a system are stiff, with stiffness arising from either or both of the additive forcing terms. Such additive forcing structure can be exploited in constructing an efficient temporal numerical integrator to solve the model equations. This is often accomplished through the use of a splitting [16, 7] or a partitioned approach [4, 5]. Both of these approaches have advantages and disadvantages and construction of methods of either type, particularly

[*] Submitted to the editors June 16, 2023

Funding: The work was supported in part by the NSF grants DMS-1720495 and DMS-2012875 and through the summer internship at Lawrence Livermore National Laboratory.

[†] School of Natural Sciences, University of California, 5200 N. Lake Road, Merced, CA 95343, United States
(vdallerit@ucmerced.edu, mtokman@ucmerced.edu).

[‡] Department of Mathematics, Tulane University, 6823 St. Charles Avenue, New Orleans, Louisiana 70118, United States (tbuvoli@tulane.edu)

[§] Recherche en prévision numérique atmosphérique, Environnement et Changement climatique Canada, 2121 Route Transcanadienne, Dorval, Québec, H9P 1J3, Canada
(stephane.gaudreault@ec.gc.ca).

of higher-order, is still an active area of research. In this paper, we focus on the partitioning approach to develop new methods.

Some of the best-known partitioned integrators are implicit-explicit (IMEX) methods [3] which have been used for a wide range of applications [12, 19, 13, 11]. IMEX techniques treat one component of the forcing term implicitly and the other explicitly, thus these methods are appropriate for problems where one of the forcing terms is responsible for stiffness in the system. For problems with stiffness present in both forcing terms, partitioning approach has been extended to implicit-implicit methods [22] and implicit-exponential (IMEXP) integrators [14, 2, 8]. For such problems, however, more attention has been dedicated to systems where one of the forcing terms is linear. Fewer options have been introduced for problems with nonlinear-nonlinear additive stiff forcing structure [2, 8].

In this work, we present a novel way to construct implicit-exponential-type methods for precisely such systems. In other words, we develop a new way to construct partitioned time integration schemes that treat f_1 implicitly and f_2 exponentially for problems where both of these functions are nonlinear and their Jacobians $J_{1,n} = \frac{\partial f_1}{\partial y}$ and $J_{2,n} = \frac{\partial f_2}{\partial y}$ are stiff.

This approach is particularly advantageous when one of the forcing terms can be treated implicitly in a very efficient way, e.g., when a fast preconditioner exists for this portion of the Jacobian. We extend the work in [15] to problems where both f_1 and f_2 are nonlinear and introduce a new ansatz for constructing such partitioned implicit-exponential integrators which can potentially be extended to higher order methods. We also describe a convenient way to visualize and assess the stability of the methods and choose schemes with favorable stability properties. The efficiency and accuracy of the new techniques are demonstrated on a set of test problems in a numerical study which also includes a thorough comparison of the performance of the new methods with previously introduced partitioned schemes for such problems. In the following, it is assumed that the partition into f_1 and f_2 is consistent with the problem to be solved and the Jacobians corresponding to each of the forcing terms are well-defined matrices such that their matrix exponentials or their inverses can be approximated in a stable manner.

The article is organized as follows. The first section briefly reviews the exponential and Rosenbrock methods, which serve as a building block of our new techniques. Section 3 introduces the novel ansatz for the partitioned implicit-exponential methods and presents the construction of the new second-order schemes of this type. Linear stability analysis of the new methods is included in section 4 where we also show that some of our schemes are A-stable. Finally, in section 5 we validate and compare the performance of our methods to other techniques using several numerical test problems.

2. Review of basic exponential and Rosenbrock methods.

The new partitioned methods which will be introduced in section 3 use both exponential and Rosenbrock-type integration to advance (1.1a) in time. Thus here we present a brief overview of these two approaches as the building blocks of our new schemes.

Consider the following (unpartitioned) system of ordinary differential equations (ODEs)

$$\frac{dy}{dt} = f(y), \qquad y(t_0) = y_0, \qquad y \in \mathbb{R}^N,\ f : \mathbb{R}^N \to \mathbb{R}^N \tag{2.1}$$

where y represents some unknown dynamically changing properties of the system, and f describes all forces driving the system. Suppose we are interested in computing the solution to this system over an interval $t \in [t_0, T]$. Letting h be the discretization step size and $y_n = y(t_n)$ denote the approximate solution at $t_n = t_0 + hn$, one can expand (2.1) in a Taylor series to obtain:

$$\frac{dy}{dt}(t) = f(y_n) + J_n \cdot (y(t) - y_n) + r(y(t)), \tag{2.2}$$

where $J_n = \dfrac{df}{dy}(y_n)$ and

$$r(z) = f(z) - f(y_n) - J_n \cdot (z - y_n). \tag{2.3}$$

Using the integrating factor $e^{-J_n\, t}$ on equation (2.2) we can write it in the form

$$\frac{d}{dt}\left(e^{-J_n\, t} y(t)\right) = e^{-J_n\, t}(f(y_n) - J_n y_n) + e^{-J_n\, t} r(y(t)). \tag{2.4}$$

Integrating over the time interval $[t_n, t_n + h]$ and multiplying by $e^{J_n(t_n+h)}$ leads to the integral form

$$y(t_n + h) = y_n + \varphi_1(hJ_n)hf(y_n) + \int_{t_n}^{t_n+h} e^{J_n(t_n+h-t)} r(y(t))dt, \tag{2.5}$$

where the matrix function φ_1 is defined as $\varphi_1(A) = (e^A - I)A^{-1}$ and I is the identity matrix. This equation, called the Volterra equation among its other names, is the starting point for the construction of different exponential integrators by introducing approximations to the terms of the right-hand side to estimate $y_{n+1} \approx y(t_n + h)$.

For instance, a second-order exponentially fitted Euler method (EPI2) [18] can be constructed by neglecting the nonlinear integral in (2.5), e.g.:

$$y_{n+1} = y_n + \varphi_1(hJ_n)hf(y_n). \tag{2.6}$$

The action of the matrix function φ_1 on a vector can be either evaluated exactly or approximated depending on the properties of the matrix J_n. For example, $\varphi_1(hJ_n)$ can be calculated exactly if J_n is small or diagonal. When the Jacobian is large and sparse, a variety of approximation techniques such as Taylor expansions [1], Krylov-based algorithms [9] or Leja methods [6] can be used.

A one-stage second-order Rosenbrock scheme [20], denoted here ROS2, could be derived by analogously neglecting the integral in (2.5) but also replacing the φ_1 function by its Padé approximant of order [0/1] :

$$y_{n+1} = y_n + h\left(I - \frac{h}{2}J_n\right)^{-1} f(y_n) \tag{2.7}$$

It is worth mentioning that the ROS2 scheme is very close in its formulation (differing only by a factor $\frac{1}{2}$ in front of the Jacobian) to the linearized Euler method

$$y_{n+1} = y_n + h\left(I - hJ_n\right)^{-1} f(y_n) \tag{2.8}$$

However, the linearized Euler method is only first-order.

Both EPI2 and ROS2 require evaluation of matrix functions of the full (unpartitioned) Jacobian J_n and share similar properties in terms of linear stability. The choice between the two, therefore, depends on the nature of the problem to be solved.

For example, when an efficient solver for a linear system of equations is available (e.g., a direct solver or a preconditioned iterative method), then the ROS2 scheme may be a judicious choice. Otherwise, exponential approximation of $\varphi_1(hJ_n)f_n$ might be more efficient when used with a fast algorithm such as the KIOPS method [9]. The case where an efficient linear solver is only available for a portion of the Jacobian will be discussed in the next section.

3. New nonlinear-nonlinear partitioned Rosenbrock-Exponential (ROSEXP) methods.

3.1. General framework.

3.1. General framework. Below we introduce a framework for developing efficient numerical schemes for solving nonlinear-nonlinear partitioned problems of the form:

$$y' = f(y) = f_1(y) + f_2(y), \qquad y(t_0) = y_0, \tag{3.1}$$

where f_1 and f_2 are both stiff. To develop such schemes, we use the idea of generalized EPI methods introduced in [23]. Specifically, in [23] it was proposed to construct approximation to the solution of (3.1) in the form

$$y_{n+1} = y_n + \sum_i \psi_i(hJ_n)f(z_i) \tag{3.2}$$

where $\psi_i(hJ_n)$ are functions of a matrix J_n which in some way approximates the Jacobian or a portion of the Jacobian, and z_i's are vectors approximating the solution on some nodes. The functions ψ_i are chosen to construct integrators of a particular type. For example, these functions can be exponential or rational depending on whether an exponential, an implicit, or a hybrid method is being built.

We extend this idea to the case of a partitioned right-hand side and allow these functions to be a product of exponential or rational functions, each applied to either the Jacobian of f_1 or f_2. For low order methods, this idea can be expressed using the following ansatz:

$$y_{n+1} = y_n + Q_{1,1}(hJ_{1,n})Q_{2,1}(hJ_{2,n})hf_1(y_n) + Q_{1,2}(hJ_{1,n})Q_{2,2}(hJ_{2,n})hf_2(y_n) \tag{3.3}$$

where: $Q_{i,j}$ are analytic functions (rational or exponential-like functions), $J_{1,n}$ and $J_{2,n}$ are respectively the Jacobians of f_1 and f_2 evaluated at y_n. Note that the multiplication order of the functions $Q_{1,i}$ and $Q_{2,i}$ in the above ansatz can be changed to derive different schemes. Since matrices $J_{1,n}$ and $J_{2,n}$ do not necessarily commute, we can also consider the following flipped ansatz that reverses the order of application of the functions:

$$y_{n+1} = y_n + Q_{2,1}(hJ_{2,n})Q_{1,1}(hJ_{1,n})hf_1(y_n) + Q_{2,2}(hJ_{2,n})Q_{1,2}(hJ_{1,n})hf_2(y_n) \tag{3.4}$$

Because the functions $Q_{i,j}$ are only applied to either $J_{1,n}$ or $J_{2,n}$, availability of efficient solvers that estimate $Q_{i,j}$'s applied to each of these matrices separately for some problems can result in significant computational savings compared to a method which involves only the full Jacobian $J_n = J_{1,n} + J_{2,n}$. This ansatz is very general and allows the construction of many methods. In the following section, we present the derivation of several efficient second-order schemes.

3.2. Construction of second-order schemes. In this section, we derive the classical order conditions necessary for a scheme based on the ansatz (3.3) to have second order of convergence. To do so, we assume that the numerical solution at time t_n is exact ($y(t_n) = y_n$) and match the numerical solution at the next time step y_{n+1} to $y(t_{n+1})$, the exact solution at time t_{n+1} up to and including second order terms. This will add some restrictions on the functions $Q_{i,j}$ that will be used to derive second-order schemes.

First, we assume that the functions $Q_{i,j}$ are analytic, so that we have the following Taylor series representation

$$Q_{i,j}(hA) = \alpha_{i,j} + \beta_{i,j} hA + O(h^2)$$

Without loss of generality, we can assume that $\alpha_{i,j} = 1$. If it is not the case, the function can be rescaled. Moreover, the product of the scaling coefficients must be equal to 1 for consistency. We use these expansions of $Q_{i,j}$ to obtain the following form of the numerical solution

$$\begin{aligned}
y_{n+1} &= y_n + Q_{1,1}\left(hJ_{1,n}\right) Q_{2,1}\left(hJ_{2,n}\right) hf_1\left(y_n\right) + Q_{2,1}\left(hJ_{1,n}\right) Q_{2,2}\left(hJ_{2,n}\right) hf_2\left(y_n\right) \\
&= y_n + \left(1 + \beta_{1,1} hJ_{1,n}\right)\left(1 + \beta_{2,1} hJ_{2,n}\right) hf_1(y_n) \\
&\qquad + \left(1 + \beta_{1,2} hJ_{1,n}\right)\left(1 + \beta_{2,2} hJ_{2,n}\right) hf_2(y_n) + O\left(h^3\right) \\
&= y_n + h\left(f_1\left(y_n\right) + f_2(y_n)\right) \\
&\qquad + h^2\left[\left(\beta_{1,1} J_{1,n} + \beta_{2,1} J_{2,n}\right) f_1(y_n) + \left(\beta_{1,2} J_{1,n} + \beta_{2,2} J_{2,n}\right) f_2(y_n)\right] + O\left(h^3\right)
\end{aligned}$$

On the other side, the exact solution at time t_{n+1} can be expanded as follows,

$$y(t_{n+1}) = y(t_n) + h(f_1(y_n) + f_2(y_n)) + \frac{h^2}{2}(J_{1,n} + J_{2,n})(f_1(y_n) + f_2(y_n)) + O\left(h^3\right)$$

After matching the terms up to second order, we have the following conditions on the functions $Q_{i,j}$:

$$\beta_{1,1} = \beta_{2,1} = \beta_{1,2} = \beta_{2,2} = 1/2$$

Table 3.1 presents several schemes that satisfy these conditions. These methods were obtained by choosing the functions $Q_{i,j}$ to be exponential or rational functions similar to those found in formulas for the EPI2 and ROS2 schemes. For this reason, if we consider the extreme case partitioning $f_1 = 0, f_2 = f$ then all the schemes from Table 3.1 reduce to the EPI2 method. Likewise, if $f_1 = f, f_2 = 0$ then all the schemes simplify to the ROS2 scheme. It should be noted that we derived these methods using classical order conditions theory. In our future research we plan to investigate whether stiffly accurate methods approach of [15] can be used to build similar partitioned methods.

As mentioned above, implicit-exponential (IMEXP) schemes for linear-nonlinear partitioned problems were introduced in [15]. In particular, the scheme *HImExp2N* (equation (4.2) in [15]) is derived for problems of the type $y' = Ly + N(y)$ where L is a linear operator and N is a nonlinear operator. Interpreting this scheme in the context of our ansatz and the derived order conditions, we can easily see that the method *HImExp2N* also satisfies the order conditions (3.2). Thus, *HImExp2N* can also be used for the nonlinear-nonlinear partitioned problems and is, in fact, a second order scheme for problems of the form (1.1a). Using the notation from this article, *HImExp2N* can be written as:

Coefficients	Scheme
	RosExp2 – ansatz (3.3)
$Q_{1,1}(z) = Q_{1,2}(z) = \left(I - \frac{z}{2}\right)^{-1}$	$y_{n+1} = y_n + \left(I - \frac{h}{2} J_{1,n}\right)^{-1} \varphi_1(hJ_{2,n}) hf(y_n)$
	ExpRos2 – ansatz (3.4)
$Q_{2,1}(z) = Q_{2,2}(z) = \varphi_1(z)$	$y_{n+1} = y_n + \varphi_1(hJ_{2,n}) \left(I - \frac{h}{2} J_{1,n}\right)^{-1} hf(y_n)$
	PartRosExp2 – ansatz (3.3)
$Q_{1,1}(z) = Q_{1,2}(z) = \left(I - \frac{z}{2}\right)^{-1}$	$y_{n+1} = y_n + \left(I - \frac{h}{2} J_{1,n}\right)^{-1} \left[\frac{1}{2}\left(e^{hJ_{2,n}} + I\right) hf_1(y_n) + \varphi_1(hJ_{2,n}) hf_2(y_n)\right]$
$Q_{2,1}(z) = \frac{1}{2}\left(e^z + I\right)$	**PartExpRos2** – ansatz (3.4)
$Q_{2,2}(z) = \varphi_1(z)$	$y_{n+1} = y_n + \frac{1}{2}\left(e^{hJ_{2,n}} + I\right) \left(I - \frac{h}{2} J_{1,n}\right)^{-1} hf_1(y_n) + \varphi_1(hJ_{2,n}) \left(I - \frac{h}{2} J_{1,n}\right)^{-1} hf_2(y_n)$

Table 3.1: Second-order Rosenbrock-Exponential schemes

$$Y_1 = y_n + \frac{h}{2}\left(I - \frac{h}{2} J_{1,n}\right)^{-1} f(y_n) \tag{3.5a}$$

$$y_{n+1} = y_n + h\left(I - \frac{h}{2} J_{1,n}\right)^{-1} f(y_n) + 2h\varphi_2(hJ_{2,n})(f_2(Y_1) - f_2(y_n)) \tag{3.5b}$$

Other ideas of partitioned nonlinear-nonlinear schemes were also explored in [8, 2], but both of the methods derived in these publications are limited to first-order accuracy. We will include the SIERE and SBDF2ERE schemes derived in these papers in our comparisons:

- *SIERE* [8]:

$$y_{n+1} = y_n + h\left(I - hJ_{1,n}\right)^{-1}\left(f_1(y_n) + \varphi_1(hJ_{2,n}) f_2(y_n)\right)$$

- *SBDF2ERE* [2]:

$$y_{n+1} = y_n + \frac{1}{3}\left(I - \frac{2h}{3} J_{1,n}\right)^{-1}\left(y_n - y_{n-1} + 2hf_1(y_n) + 2h\varphi_1(hJ_{2,n}) f_2(y_n)\right)$$

Note that all the schemes are written so that the f_1 partition is treated using the rational function, while the f_2 partition is treated exponentially.

4. Linear stability. As mentioned in the previous section, our work focuses on nonlinear-nonlinear partitioning where both f_1 and f_2 are stiff. In this context, it is important to have good stability properties. In order to study the linear stability of partitioned integrators we assume that the partitioned Jacobian terms $J_{1,n}$ and $J_{2,n}$ are simultaneously diagonalizable and consider the following problem:

$$y' = \lambda_1 y + \lambda_2 y \quad \text{where} \quad \lambda_1, \lambda_2 \in \mathbb{C}. \tag{4.1}$$

	RosExp2 ExpRos2 HImExp2N	*PartRosExp2 PartExpRos2*	*SIERE*
$R(z_1, z_2)$	$1 + \frac{2\varphi_1(z_2)}{2-z_1}(z_1 + z_2)$	$\frac{2+z_1}{2-z_1}e^{z_2}$	$\frac{e^{z_2}}{1-z_1}$

Table 4.1: Stability functions for one-step methods.

Note that this problem is not able to give a full picture of the linear stability of additively partitioned problems. For example, it does not take into account the potential coupling between the two partitions. However, based on our experiments, this problem allows us to give a good description of the stability of the different methods.

Any one-step method applied to (4.1) reduces to the recurrence $y_{n+1} = R(z_1, z_2)y_n$ where $z_1 = h\lambda_1$, $z_2 = h\lambda_2$ and $R(z_1, z_2)$ is the stability function of the scheme. The scheme is then stable if $|R(z_1, z_2)| \leq 1$. The stability functions for all one-step methods considered in this work are listed in Table 4.1.

Because the *SBDF2ERE* scheme is a multi-step method, we determine stability differently. After applying the method to the linear problem (4.1) we obtain the recurrence

$$y_{n+1} + R_1(z_1, z_2)y_n + R_0(z_1, z_2)y_{n-1} = 0$$

where $R_1(z_1, z_2) = -\frac{2}{3-2z_1}(1 + e^{z_2})$ and $R_0(z_1, z_2) = \frac{1}{3-2z_2}$. The method will be stable when w_1 and w_2, the roots of the polynomial $w^2 + R_1(z_1, z_2)w + R_0(z_1, z_2)$, satisfy $|w_i| \leq 1$.

Because both z_1 and z_2 are complex-valued, the stability regions for both one-step and multi-step methods are challenging to visualize. To simplify our presentation of stability, we will use $A(\boldsymbol{\alpha})$-stability. For a stability function $R(z)$ of a single complex variable, a method is said to be $A(\boldsymbol{\alpha})$-stable if it includes a sector of an angle $\boldsymbol{\alpha}$ in its stability region with $\boldsymbol{\alpha}$ defined as:

$$\text{(4.2)} \qquad \boldsymbol{\alpha} = \max\{\alpha : \forall z \quad (z \in \mathbb{C}^- \wedge |\arg(z) - \pi| \leq \alpha) \Rightarrow |R(z)| \leq 1\}.$$

For non-partitioned schemes, $\boldsymbol{\alpha}$ is the maximum value of the angle such that the method is stable for all complex z values in the sector delimited by the lines with an angle $-\boldsymbol{\alpha}$ and $+\boldsymbol{\alpha}$ with respect to the negative real axis. This value ranges from 0° if the method is only stable on the negative real axis to 90° for a method that is stable in the entire left half-plane. When the angle of the $\boldsymbol{\alpha}$-stability of a scheme is equal to 90°, we say that the method is A-stable.

By fixing either z_1 or z_2, we can reduce the stability function of a partitioned scheme to a function of a single complex variable. We can then compute the stability angle $\boldsymbol{\alpha}$ in the remaining free variable. This can be expressed mathematically as

(4.3a)
$$\text{fixing } z_1\text{:} \quad \boldsymbol{\alpha}(z_1) = \max\{\alpha : \forall z_2 \quad (z_2 \in \mathbb{C}^- \wedge |\arg(z_2) - \pi| \leq \alpha) \Rightarrow |R(z_1, z_2)| \leq 1\},$$

(4.3b)
$$\text{fixing } z_2\text{:} \quad \boldsymbol{\alpha}(z_2) = \max\{\alpha : \forall z_1 \quad (z_1 \in \mathbb{C}^- \wedge |\arg(z_1) - \pi| \leq \alpha) \Rightarrow |R(z_1, z_2)| \leq 1\},$$

Fixing z_1 or z_2 over a grid of values and using color to represent the stability angle makes it possible to easily visualize the stability of each method. Figure 4.1 shows the $\boldsymbol{\alpha}$-stability for the schemes presented in the previous section. Note that ordinarily, due to the high dimensionality of the stability function $R(z_1, z_2)$, it is difficult to assess the properties of the stability regions. Using the approach described above, it is easier to visualize the stability regions. To our knowledge, this approach to visualizing the linear stability of a method has not been used before. Plots like Figure 4.1 provide a visual guide to the overall shape of the stability regions. Additional visualization can be done if one is interested in the geometric details of a subregion. For example, different types of plots, such as graph of the angle value along the real axes, can be created to better assess stability for eigenvalues on the real axes.

In Figure 4.1 (a), we see that $\boldsymbol{\alpha} = 90^\circ$ for all values of z_1 and z_2. This implies that the schemes *PartRosExp2*, *PartExpRos2*, and *SIERE* are all A-stable. This can be formally proven by observing that the stability function for each of these methods is a product of two A-stable functions in z_1 and z_2, respectively (e.g. the stability function of *PartRosExp2* and *PartExpRos2* are the products of the functions $R_1(z) = e^z$ and $R_2(z) = \frac{2+z}{2-z}$). Since both of these functions are A-stable, the product must also be A-stable.

Figure 4.1 (b), shows that the stability of the schemes *RosExp2, ExpRos2* and *HImExp2N* is more restricted. Specifically, there are restrictions on stability in z_1 if values of z_2 are close to the imaginary axes. However, for problems with spectrum lying sufficiently away from the imaginary axes stability is retained. Finally, the stability of the scheme *SBDF2ERE*, presented in Figure 4.1 (c), is good overall, with some limitations close to the origin.

5. Numerical experiments.

The stability properties of the new schemes for linear equations with constant coefficients provide necessary, but not sufficient conditions for the stability of variable coefficients and nonlinear problems.

In this section, we summarize numerical experiments that confirm that the conclusions of our analysis also apply to more complicated problems.

5.1. Advection-diffusion PDE (AdvDiff).

We consider the following 1D advection-diffusion PDE:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\alpha_0 u + \alpha_1 u^2\right) = \frac{\partial}{\partial x}\left[(\beta_0 + \beta_1 u)\frac{\partial u}{\partial x}\right], \quad x \in [0,1], \quad t \in [0, 0.1], \tag{5.1}$$

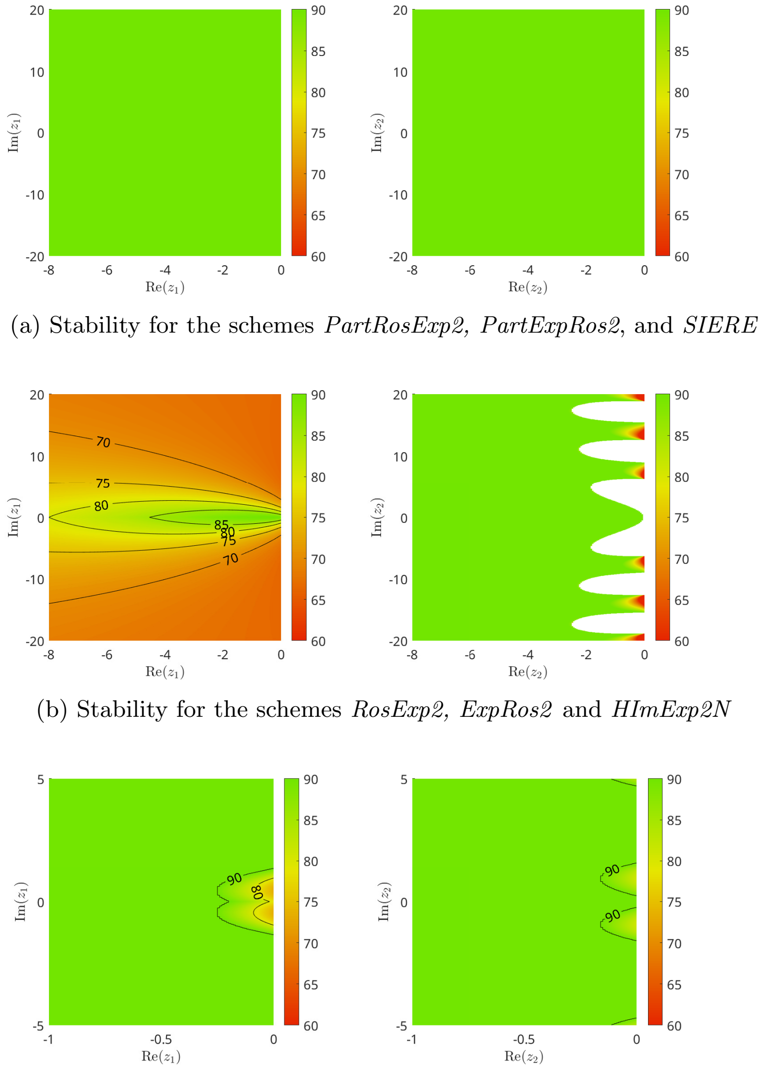
We use a Gaussian function as the initial condition $u(x,0) = e^{-5000(x-0.2)^2}$ and homogeneous Dirichlet boundary conditions $u(0,t) = u(1,t) = 0$. We also consider two sets of parameters: the first correspond to a linear problem with $\alpha_0 = 5$, $\alpha_1 = 0$, $\beta_0 = 10^{-2}$, and $\beta_1 = 0$, and the second representing a nonlinear problem with $\alpha_0 = 5$, $\alpha_1 = 5$, $\beta_0 = 5 \times 10^{-4}$, and $\beta_1 = 10^{-1}$. Figure 5.1 shows the solution u of this PDE at the initial and final time. Equation (5.1) is discretized in space using standard second-order centered finite differences with 1000 grid points. This discretization leads to a system of N ordinary differential equations that can be written as:

$$u' = f_{\text{adv}}(u) + f_{\text{diff}}(u)$$

where $f_{\text{adv}}(u)$ correspond to the discretized advection term $\frac{\partial}{\partial x}\left(\alpha_0 u + \alpha_1 u^2\right)$ and $f_{\text{diff}}(u)$ correspond to the discretized diffusion term $\frac{\partial}{\partial x}\left[(\beta_0 + \beta_1 u)\frac{\partial u}{\partial x}\right]$. In the next section, we will explore the cases where $f_1 = f_{\text{adv}}, f_2 = f_{\text{diff}}$ (rational advection / exponential diffusion) as well as $f_1 = f_{\text{diff}}, f_2 = f_{\text{adv}}$ (rational diffusion / exponential advection).

5.2. Schnakenberg with non-linear diffusion (Schnakenberg_NL).

The following equations describe two reacting and diffusing chemical species (u and v)



(a) Stability for the schemes *PartRosExp2, PartExpRos2*, and *SIERE*

(b) Stability for the schemes *RosExp2, ExpRos2* and *HImExp2N*

(c) Stability for the scheme *SBDF2ERE*

Fig. 4.1: $\boldsymbol{\alpha}$-stability angles for the partitioned schemes when z_1 is fixed (left-column) or z_2 is fixed (right-column). The x and y axis of the plots in the left and right columns, respectively, correspond to the real and imaginary parts of z_1 and z_2. The color represents the stability angle $\boldsymbol{\alpha}$ defined in (4.3a) and (4.3b). The white regions correspond to parameter values where the stability region is bounded and therefore not $\boldsymbol{\alpha}$-stable even for $\boldsymbol{\alpha} = 0$.

evolving in two-dimensional space:

$$\frac{\partial u}{\partial t} = \gamma(a - u + u^2 v) + \nabla.(u^{\beta_1}\nabla u), \tag{5.2a}$$

$$\frac{\partial v}{\partial t} = \gamma(b - u^2 v) + d\ \nabla.(v^{\beta_2}\nabla v), \qquad (x,y) \in [0,1]^2 \tag{5.2b}$$

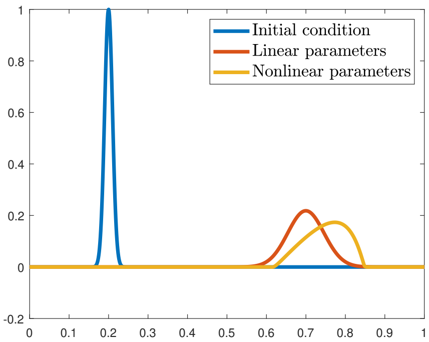


Fig. 5.1: Solution at initial and final time for the PDE (5.1) with both sets of parameters

where $a = 0.1, b = 0.9, d = 10, \beta_1 = \beta_2 = 10, t_{end} = 10^{-2}$, and $\gamma = 1000$. As in [17, Section 4.2] the initial condition is a perturbation of the stable equilibrium and the boundary conditions are periodic in both directions. The diffusion terms are discretized using the standard second-order finite differences on a uniform grid with $N_x = N_y = 128$. The reaction terms are treated exponentially, while the diffusion terms are treated using the rational function.

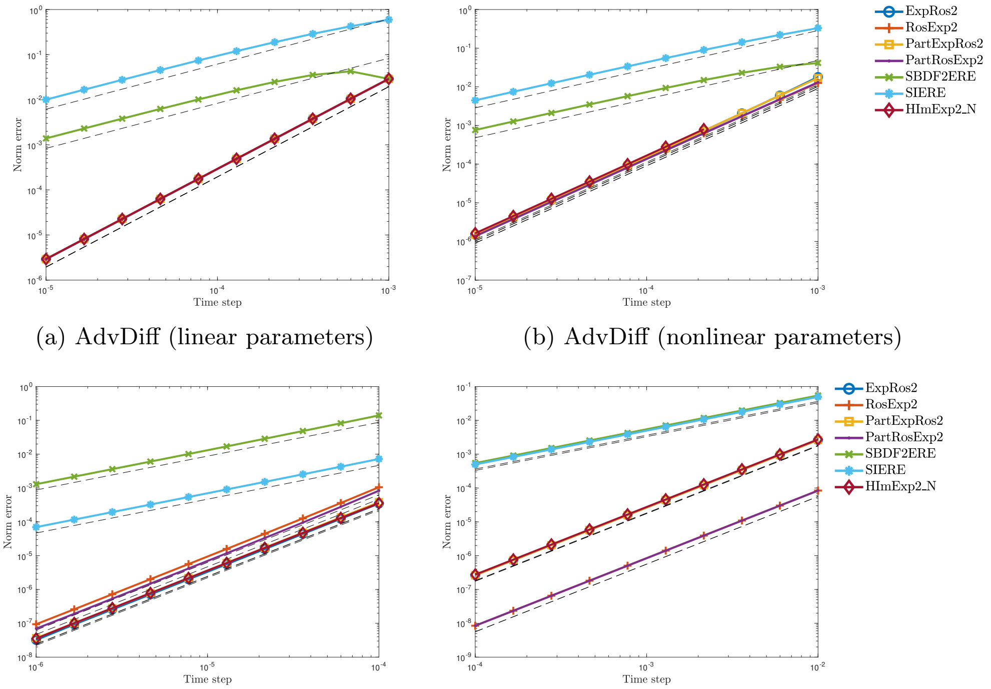
5.3. 1D Semilinear parabolic problem (Semilinear_para). Finally, we use the following one-dimensional semilinear parabolic problem described in [10] (note we use the term ”semilinear parabolic problem” as it was named in [10]):

$$(5.3) \qquad \frac{\partial u}{\partial t}(x,t) - \frac{\partial^2 u}{\partial x^2}(x,t) = \int_0^1 u(x,t)dx + \phi(x,t) \qquad x \in [0,1], t \in [0,1],$$

with the homogeneous Dirichlet boundary conditions. The source function ϕ is chosen so that $u(x,t) = x(1-x)e^t$ is the exact solution. This problem was originally designed to demonstrate the order reduction that some exponential integrators can suffer when applied to stiff problems. It is therefore used here to validate that no such order reduction is exhibited by our schemes. The diffusion term is discretized using the standard second-order finite differences on a uniform grid with $N_x = 400$. The nonlinear terms on the right-hand side are treated exponentially, while the diffusion term is treated using the rational function.

5.4. Numerical results. Numerical examples presented below verify the order of convergence of the newly derived methods and compare their performance with the existing methods described above. The implementation of the integrators was done in MATLAB 2020b. For all the schemes, we use the KIOPS method introduced in [9] to approximate the products of exponential and $\varphi-$functions with vectors. This method allows us to approximate both exponential functions in the schemes *PartExpRos2* and *PartRosExp2* at once as a single computation. The rational functions are approximated using the GMRES method [21] with an incomplete LU factorization with no fill preconditioner (ILU(0)). Because the scheme *BDF2ERE* is a multi-step integrator where the solution at the current and previous time step must be known, the initial step must be treated differently. In this work, the initial time step is computed using the 2^{nd} order *EPI2* method. The error is computed at the final time as the discrete $2-$norm between the approximate solution and a reference solution computed using MATLAB's ode15s integrator with absolute and relative tolerances set to 10^{-14}.

In the first set of tests, we verify the order of convergence of all the methods on the problems presented above. Figure 5.2 shows the convergence plot (error vs. time-step in log-log scale) on the linear and nonlinear advection-diffusion PDE, Schnakenberg PDE, and the semilinear parabolic problems. Note that for the advection diffusion PDE, we used $f_1 = f_{\text{adv}}$ and $f_2 = f_{\text{diff}}$. We can see that, as expected, the methods *SBDF2ERE* and *SIERE* both converge at first-order, while the methods introduced in Table 3.1 and the *HImExp2N* scheme converge at second order. We can also see that for the advection-diffusion and the Schnakenberg PDE, the order of multiplication of the functions $Q_{i,j}$ does not influence the accuracy of the solution (ansatz (3.3) vs. (3.4)). However, for the semilinear parabolic problem, the order does affect the accuracy. For this problem and this partitioning, applying the function of $J_{2,n}$ first leads to better accuracy. This case illustrates that the accuracy of the method does depend on the problem and the chosen partitioning.



(a) AdvDiff (linear parameters)

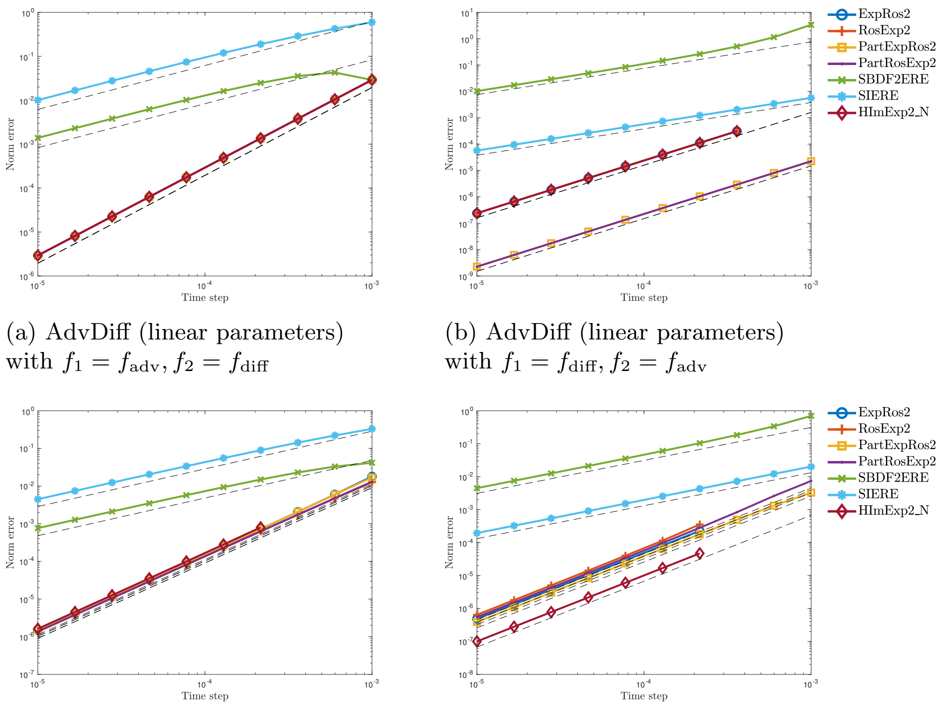
(b) AdvDiff (nonlinear parameters)

(c) Schnakenberg_NL ($\gamma = 10^3$)

(d) Semilinear_para

Fig. 5.2: Convergence plots (error vs. time step) for the linear and nonlinear AdvDiff, Schnakenberg_NL, and Semilinear_para problems

Next, we want to validate the stability advantages of the new schemes that our analysis of section 4 predicted. We showed that for the schemes *ExpRos2*, *RosExp2* and *HImExp2N*, if z_2 is close to the imaginary axis, then stability for z_1 is either bounded or restricted. Figure 5.3 shows the convergence diagram for the advection-diffusion PDE problem. Figures 5.3a and 5.3b correspond to the problem with linear parameters while Figures 5.3c and 5.3d correspond to the nonlinear parameters. The plots on the left (Figures 5.3a and 5.3c) are obtained using the partitioning $f_1 = f_{\text{adv}}, f_2 = f_{\text{diff}}$ and the plots on the right (Figures 5.3b and 5.3d) are obtained using the partitioning $f_1 = f_{\text{diff}}, f_2 = f_{\text{adv}}$. The eigenvalues corresponding to the advection term f_{adv} are expected to be close to the imaginary axis, while the eigenvalues of the diffusion term are expected to be along the negative real axis. Therefore, based on the stability analysis, we are expecting the schemes *ExpRos2*, *RosExp2* and *HImExp2N* to have worse stability for $f_2 = f_{\text{adv}}$ (right plots). For both the linear and nonlinear parameters, we see that this is indeed the case, and these methods are stable only for a more restrictive range of time step sizes.



(a) AdvDiff (linear parameters) with $f_1 = f_{\text{adv}}, f_2 = f_{\text{diff}}$

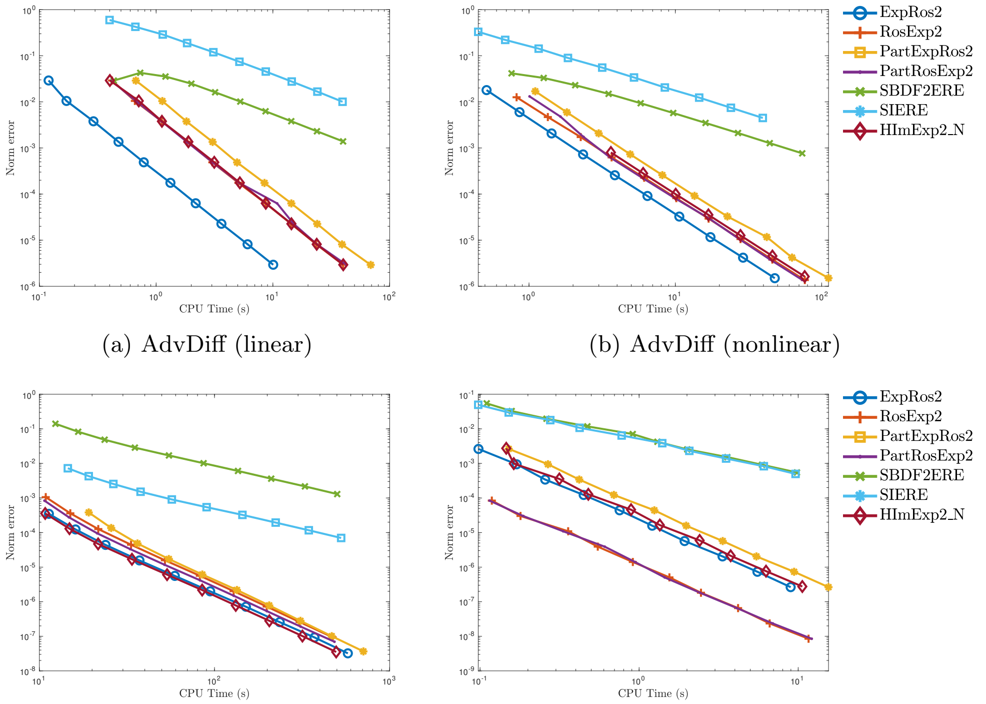
(b) AdvDiff (linear parameters) with $f_1 = f_{\text{diff}}, f_2 = f_{\text{adv}}$

(c) AdvDiff (nonlinear parameters) with $f_1 = f_{\text{adv}}, f_2 = f_{\text{diff}}$

(d) AdvDiff (nonlinear parameters) with $f_1 = f_{\text{diff}}, f_2 = f_{\text{adv}}$

Fig. 5.3: Stability comparison for the AdvDiff problem with different partitioning

We now compare the performance of the methods on the different test problems. Figure 5.4 shows the precision diagrams (error vs. CPU time) for the linear and nonlinear advection-diffusion PDE, Schnakenberg PDE, and the semilinear parabolic problems. As expected, the precision diagrams clearly demonstrate that the first order methods *SBDF2ERE* and *SIERE* are less efficient than all of the second order schemes. Also for cases where the linear solve is sufficiently more costly than the exponential functions estimation, such as systems solved in Figure 5.4(a), (c), method *PartExpRos2* is less efficient since unlike all other second order schemes it requires two linear systems to be solved per iteration. Among the second orders methods there is no a clear winner in terms of efficiency and the choice of the best methods should depend on the particulars of the operators f_1 and f_2 and the costs of evaluating these functions, their respective Jacobian contributions and the corresponding costs of linear solves and exponential functions evaluations.



(a) AdvDiff (linear)

(b) AdvDiff (nonlinear)

(c) Schnakenberg_NL ($\gamma = 10^3$)

(d) Semilinear_para

Fig. 5.4: Precision diagram (error vs. CPU time) for the linear and nonlinear AdvDiff, Schnakenberg_NL, and Semilinear_parabolic problems.

6. Conclusion. In this paper, we presented a new framework for deriving partitioned integrators for stiff systems of ODEs with nonlinear-nonlinear additive forcing terms. The new time integrators constructed using this framework are particularly efficient for problems where both nonlinear forcing terms are stiff, but one of them can be solved efficiently using an implicit approach, and another can be integrated exponentially. The new ansatz that allowed us to derive specific second-order schemes can potentially be extended to construct higher-order methods, where the choice of the sequential order for the operators is also very important. We intend to pursue this line of research in our future work. We have used linear stability analysis and a novel way to visualize the properties of a stability function to demonstrate that several of the new methods are A-stable and thus offer superior stability compared to existing schemes for similar problems. Convergence and efficient performance of the new methods have been demonstrated using several numerical examples. A thorough comparison of these schemes with integrators proposed for such problems in previous publications has been performed. We showed that the novel exponential-Rosenbrock-type methods are both more accurate and more stable than previously published methods and can be effectively used for a variety of applications.

REFERENCES

[1] A. H. Al-Mohy and N. J. Higham, *Computing the action of the matrix exponential, with an application to exponential integrators*, SIAM journal on scientific computing, 33 (2011), pp. 488–511.

[2] U. M. Ascher, E. Larionov, S. H. Sheen, and D. K. Pai, *Simulating deformable objects for computer animation: A numerical perspective*, Journal of Computational Dynamics, 9 (2022), pp. 47–68.

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

[4] U. M. Ascher, S. J. Ruuth, and B. T. Wetton, *Implicit-explicit methods for time-dependent partial differential equations*, SIAM Journal on Numerical Analysis, 32 (1995), pp. 797–823.

[5] T. Belytschko, H.-J. Yen, and R. Mullen, *Mixed methods for time integration*, Computer Methods in Applied Mechanics and Engineering, 17 (1979), pp. 259–275.

[6] M. Caliari, M. Vianello, and L. Bergamaschi, *Interpolating discrete advection–diffusion propagators at Leja sequences*, Journal of Computational and Applied Mathematics, 172 (2004), pp. 79–99.

[7] J. Cervi and R. J. Spiteri, *High-order operator splitting for the bidomain and monodomain models*, SIAM Journal on Scientific Computing, 40 (2018), pp. A769–A786.

[8] Y. J. Chen, S. H. Sheen, U. M. Ascher, and D. K. Pai, *SIERE: A hybrid semi-implicit exponential integrator for efficiently simulating stiff deformable objects*, ACM Transactions on Graphics (TOG), 40 (2020), pp. 1–12.

[9] S. Gaudreault, G. Rainwater, and M. Tokman, *KIOPS: A fast adaptive Krylov subspace solver for exponential integrators*, Journal of Computational Physics, 372 (2018), pp. 236–255.

[10] M. Hochbruck and A. Ostermann, *Explicit exponential Runge–Kutta methods for semilinear parabolic problems*, SIAM Journal on Numerical Analysis, 43 (2005), pp. 1069–1090.

[11] W. H. Hundsdorfer, J. G. Verwer, and W. Hundsdorfer, *Numerical solution of time-dependent advection-diffusion-reaction equations*, vol. 33, Springer, 2003.

[12] C. A. Kennedy and M. H. Carpenter, *Additive Runge–Kutta schemes for convection–diffusion–reaction equations*, Applied numerical mathematics, 44 (2003), pp. 139–181.

[13] D. E. Keyes, L. C. McInnes, C. Woodward, W. Gropp, E. Myra, M. Pernice, J. Bell, J. Brown, A. Clo, J. Connors, et al., *Multiphysics simulations: Challenges and opportunities*, The International Journal of High Performance Computing Applications, 27 (2013), pp. 4–83.

[14] V. T. Luan, R. Chinomona, and D. R. Reynolds, *A new class of high-order methods for multirate differential equations*, SIAM Journal on Scientific Computing, 42 (2020), pp. A1245–A1268.

[15] V. T. Luan, M. Tokman, and G. Rainwater, *Preconditioned implicit-exponential integrators (IMEXP) for stiff PDEs*, Journal of Computational Physics, 335 (2017), pp. 846–864.

[16] S. MacNamara and G. Strang, *Operator splitting*, in Splitting methods in communication, imaging, science, and engineering, Springer, 2016, pp. 95–114.

[17] A. Madzvamuse and P. K. Maini, *Velocity-induced numerical solutions of reaction-diffusion systems on continuously growing domains*, Journal of Computational Physics, 225 (2007), pp. 100–119.

[18] B. Minchev and W. Wright, *A review of exponential integrators for first order semi-linear problems*, technical report 2, Norwegian University of Science and Technology, 2005.

[19] L. Pareschi and G. Russo, *Implicit–explicit Runge–Kutta schemes and applications to hyperbolic systems with relaxation*, Journal of Scientific computing, 25 (2005), pp. 129–155.

[20] H. H. Rosenbrock, *Some general implicit processes for the numerical solution of differential equations*, The Computer Journal, 5 (1963), pp. 329–330.

[21] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869.

[22] A. Sandu and M. Günther, *A generalized-structure approach to additive Runge–Kutta methods*, SIAM Journal on Numerical Analysis, 53 (2015), pp. 17–42.

[23] M. Tokman, *A new class of exponential propagation iterative methods of Runge–Kutta type (EPIRK)*, Journal of Computational Physics, 230 (2011), pp. 8762–8778.