SIAM J. SCI. COMPUT.
Vol. 0, No. 0, pp. 000–000

MULTIRATE EXPONENTIAL ROSENBROCK METHODS*

VU THAI LUAN†, RUJEKO CHINOMONA‡, AND DANIEL R. REYNOLDS‡

Abstract. In this paper we propose a novel class of methods for high-order accurate integration of multirate systems of ordinary differential equation initial-value problems. The proposed methods construct multirate schemes by approximating the action of matrix φ functions within explicit exponential Rosenbrock (ExpRB) methods, thereby called *multirate ExpRB* (MERB) methods. They consist of the solution to a sequence of modified "fast" initial-value problems, which may themselves be approximated through subcycling any desired initial-value problem solver. In addition to proving how to construct MERB methods from certain classes of ExpRB methods, we provide rigorous convergence analysis of these methods and derive efficient MERB schemes of orders 2 through 6 (the highest-order infinitesimal multirate methods to date). We then present numerical simulations to confirm these theoretical convergence rates and to compare the efficiency of MERB methods against other recently introduced high-order multirate methods.

Key words. multirate time integration, exponential Rosenbrock methods, convergence analysis

MSC codes. 65L05, 65L06, 65M20, 65L20

DOI. 10.1137/21M1439481

1. Introduction.

In this paper, we consider numerical methods to perform highly accurate time integration for multirate systems of ordinary differential equation (ODE) initial-value problems (IVPs). The primary characteristic of these problems is that they are comprised of two or more components that on their own would evolve on significantly different time scales. Such problems may be written in the general additive form

$$u'(t) = F(t,u(t)) := F_f(t,u) + F_s(t,u), \quad t \in [t_0,T], \quad u(t_0) = u_0, \tag{1.1}$$

where F_f and F_s contain the "fast" and "slow" operators or variables, respectively. Typically, due to either stability or accuracy limitations the fast processes must be evolved with small step sizes; however the slow processes could allow much larger time steps. Such problems frequently arise in the simulation of "multiphysics" systems, wherein separate models are combined together to simulate complex physical phenomena [7]. While such problems may be treated using explicit, implicit, or mixed implicit-explicit time integration methods that evolve the full problem using a shared time step size, this treatment may prove inefficient, inaccurate, or unstable, depending on which time scale is used to dictate this shared step size. Historically, scientific simulations have treated such problems using ad hoc operator splitting schemes where faster components are "subcycled" using smaller time steps than slower components.

*Submitted to the journal's Methods and Algorithms for Scientific Computing section August 9, 2021; accepted for publication (in revised form) July 11, 2022; published electronically DATE.
https://doi.org/10.1137/21M1439481

Funding: The first author is supported by NSF grant DMS-2012022. The second and third authors were supported in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, and Scientific Discovery through Advanced Computing (SciDAC) Program through the FASTMath Institute, under Lawrence Livermore National Laboratory subcontract B626484 and DOE award DE-SC0021354.

†Department of Mathematics and Statistics, Mississippi State University, Mississippi State, MS 39762 USA (luan@math.msstate.edu).

‡Department of Mathematics, Southern Methodist University, Dallas, TX 75275 USA (rchinomona@smu.edu, reynolds@smu.edu).

Schemes in this category include Lie–Trotter [22] and Strang–Marchuk [21, 28] techniques that are first- and second-order accurate, respectively. In recent years, however, methods with increasingly high orders of accuracy have been introduced. Our particular interest lies in methods allowing so-called infinitesimal formulations, wherein the fast time scale is assumed to be solved exactly, typically through evolution of a sequence of modified fast IVPs,

$$v'(\tau) = F_f(\tau, v) + g(\tau), \quad \tau \in [\tau_0, \tau_f], \quad v(\tau_0) = v_0,$$

and where the forcing function $g(\tau)$, time interval $[\tau_0, \tau_f]$, and initial condition v_0 are determined by the multirate method to incorporate information from the slow time scale. In practice, however, these fast IVPs are solved using any viable numerical method, typically with smaller step size than is used for the slow dynamics. While both the legacy Lie–Trotter and Strang–Marchuk schemes satisfy this description, each uses $g(\tau) = 0$ and only couple the time scales through the initial condition v_0. The first higher-order infinitesimal multirate methods were the *multirate infinitesimal step* (MIS) methods [26, 30], which allowed up to third-order accuracy. These have been extended by numerous authors in recent years to support fourth and fifth orders of accuracy, as well as implicit or even mixed implicit-explicit treatment of the slow time scale [1, 3, 16, 25, 27].

Most higher-order (≥ 3) infinitesimal methods, including MIS, relaxed MIS [27], extended MIS [1], multirate infinitesimal general structure additive Runge–Kutta (GARK) [24, 25], and implicit-explicit multirate infinitesimal (MRI) GARK [3], place no restrictions on the operators F_f and F_s. The corresponding order conditions for these methods are rooted in partitioned Runge–Kutta theory, to the end that the number of order conditions grows exponentially with the desired order of accuracy, to the effect that none of these methods have been proposed with order of accuracy greater than four.

In previous work, we presented an alternate approach for deriving infinitesimal multirate methods that was based on exponential Runge–Kutta (ExpRK) theory, named *multirate ExpRK* (MERK) methods [16]. A particular benefit of this theory is that ExpRK methods require fewer order conditions than partitioned Runge–Kutta methods; however, to leverage this theory, MERK methods require that the fast time scale operator is autonomous and that it depends *linearly* on the solution u; i.e., these consider the IVP

$$u'(t) = F(t, u(t)) := \mathcal{L}u + \mathcal{N}(t, u), \quad t \in [t_0, T], \quad u(t_0) = u_0, \tag{1.2}$$

where the "fast" and "slow" components are $F_f(t, u) = \mathcal{L}u$ and $F_s(t, u) = \mathcal{N}(t, u)$, respectively. With this restriction in place, however, MERK methods have been proposed with orders of accuracy up to five.

In this work, we address the case of a nonautonomous and nonlinear fast time scale operator $F_f(t, u)$ by proposing to use a dynamic linearization approach that updates the operators $\mathcal{L}$ and $\mathcal{N}$ within each time step. Nonlinear dynamical systems often operate on multiple time scales away from equilibrium; hence linearization techniques can offer important information on how such systems behave in the neighborhood of equilibrium points. Near an equilibrium point the eigenvalues of the linearized system often provide the necessary information on the time scale structure. Therefore, we expect that the dynamic linearization approach that updates $\mathcal{L}u$ (within each time step) at the fast time scale will be applicable for any dynamical system wherein its linearization captures the majority of the dynamics. In addition, as mentioned in

[11], a bad choice of fixed linearization (1.2) can lead to stability issues, for example, if the numerical solution stays near an equilibrium point of the problem for a long time. In such a case, MERK methods may require taking smaller time steps, thereby causing computational inefficiency. This further motivates us to consider the idea of linearizing (1.2) in each integration step in order to overcome these issues. We leverage this dynamic linearization approach through building multirate schemes from exponential Rosenbrock (ExpRB) methods. This new class of multirate schemes, called *multirate ExpRB* (MERB) methods, approximates the action of matrix φ functions within explicit ExpRB methods and consists of solving a sequence of modified linear ODE-IVPs, which can be integrated using any desired ODE solvers. Moreover, we establish an elegant convergence theory for MERB methods, allowing us to determine a minimum order of accuracy for the numerical methods needed for solving the corresponding fast time scale IVPs. In addition to this theory, we generalize the coefficients for a number of high-order ExpRB methods and exploit their parallel stage structure to derive efficient multirate methods of very high-order (including the first-ever infinitesimal multirate method of order 6), with optimized numbers of modified fast IVPs. Our numerical experiments show that these new proposed MERB schemes are uniformly the most efficient when considering slow function calls (of particular interest for multirate systems where the fast component is much less costly to compute than the slow component) and thus are very competitive in comparison with recently developed high-order multirate methods such as MERK and MRI-GARK.

The remainder of this paper is organized as follows. We first present the structure of ExpRB methods (section 2.1). Then in section 2.2 we interpret the corresponding ExpRB internal stages and time step approximations as exact solutions to modified "fast" IVPs, thereby deriving MERB methods. In section 2.3 we present rigorous convergence analysis for this family of newly proposed methods. Then in section 2.4 we construct specific multirate methods from this family for practical use and discuss techniques for their numerical implementation in section 2.5. In section 3 we provide detailed numerical results to compare the performance of the proposed methods with the recent MERK methods of orders 3 through 5, as well as with third- and fourth-order explicit MRI-GARK methods. Finally, we provide concluding remarks and discuss avenues for future research in section 4.

2. MERB methods.

2.1. ExpRB schemes.

ExpRB methods are constructed by linearizing the vector field $F(t,u)$ at each step along the numerical solution (t_n, u_n),

$$u'(t) = F(t,u(t)) = J_n u(t) + V_n t + N_n(t,u(t)) \tag{2.1}$$

with

$$J_n = \frac{\partial F}{\partial u}(t_n,u_n), \quad V_n = \frac{\partial F}{\partial t}(t_n,u_n), \quad N_n(t,u) = F(t,u) - J_n u - V_n t. \tag{2.2}$$

We note that if (1.1) is in fact autonomous, i.e., $u'(t) = F(u(t))$, then this linearization simplifies since $V_n = 0$ and $N_n(t,u) = N_n(u) = F(u) - J_n u$.

One can represent the exact solution to (2.1) at time $t_{n+1} = t_n + H$ as in [14] by applying the variation-of-constants formula (also known as, Duhamel's principle),

$$u(t_{n+1}) = \mathrm{e}^{HJ_n}u(t_n) + \int_0^H \mathrm{e}^{(H-\tau)J_n}\Big(V_n(t_n+\tau) + N_n(t_n+\tau, u(t_n+\tau))\Big)\mathrm{d}\tau$$

$$= \mathrm{e}^{HJ_n}u(t_n) + H\varphi_1(HJ_n)V_n t_n + H^2\varphi_2(HJ_n)V_n + \int_0^H \mathrm{e}^{(H-\tau)J_n}N_n(t_n+\tau, u(t_n+\tau))\mathrm{d}\tau, \tag{2.3}$$

where $\varphi_k(Z)$ $(Z = HJ_n)$ belong to the family of φ functions given by

$$\varphi_k(Z) = \frac{1}{H^k}\int_0^H \mathrm{e}^{(H-\tau)\frac{Z}{H}}\frac{\tau^{k-1}}{(k-1)!}\mathrm{d}\tau, \quad k \geq 1. \tag{2.4}$$

Explicit ExpRB methods approximate the integral in (2.3) by using a quadrature rule with nodes c_i in $[0,1]$ $(i = 1,\ldots,s)$ $(c_1 = 0)$. Denoting the resulting approximations $u_n \approx u(t_n)$ and $U_{ni} \approx u(t_n + c_i H)$, ExpRB methods may be written as

(2.5)

$$U_{ni} = u_n + c_i H\varphi_1(c_i HJ_n)F(t_n,u_n) + c_i^2 H^2\varphi_2(c_i HJ_n)V_n + H\sum_{j=2}^{i-1} a_{ij}(HJ_n)D_{nj},$$

$$u_{n+1} = u_n + H\varphi_1(HJ_n)F(t_n,u_n) + H^2\varphi_2(HJ_n)V_n + H\sum_{i=2}^{s} b_i(HJ_n)D_{ni},$$

where

$$D_{ni} = N_n(t_n + c_i H, U_{ni}) - N_n(t_n, u_n), \tag{2.6}$$

$(i = 2,\ldots,s)$ and where $D_{n1} = 0$ [11, 14]. Here, the weights $a_{ij}(HJ_n)$ and $b_i(HJ_n)$ are usually chosen (by construction) as linear combinations of the $\varphi_k(c_i HJ_n)$ and $\varphi_k(HJ_n)$ functions given in (2.4), respectively. These unknown functions can be determined by solving order conditions, depending on the required order of accuracy.

Remark 2.1 (order conditions). For later use, in Table 1 we recall the stiff order conditions for ExpRB methods up to order 6 from [18]. We note that an ExpRB method of order 6 only requires 7 conditions, which is much less than the 36 conditions needed for explicit Runge–Kutta or ExpRK methods of the same order. This is the advantage of the dynamic linearization approach (2.1) and can be understood by observing from (2.2) that

$$\frac{\partial N_n}{\partial u}(t_n,u_n) = 0 \quad \text{and} \quad \frac{\partial N_n}{\partial t}(t_n,u_n) = 0. \tag{2.7}$$

This property significantly simplifies the number of order conditions, particularly for higher-order schemes. A further consequence of (2.7) is that from (2.6) we have $D_{ni} = \mathcal{O}(H^2)$, meaning that ExpRB methods are at least of order 2.

2.2. A multirate procedure for ExpRB methods. Inspired by [16], we now show how ExpRB schemes can be interpreted as a class of MIS-type methods. Namely, we construct modified ODEs whose exact solutions correspond to the ExpRB internal stages U_{ni} $(i = 2,\ldots,s)$ and the final stage u_{n+1}.

LEMMA 2.2. *Consider an explicit ExpRB scheme* (2.5)*, where the weights* $a_{ij}(HJ_n)$ *and* $b_i(HJ_n)$ *can be written as linear combinations of* φ_k *functions,*

$$a_{ij}(HJ_n) = \sum_{k=1}^{\ell_{ij}} \alpha_{ij}^{(k)}\varphi_k(c_i HJ_n), \quad b_i(HJ_n) = \sum_{k=1}^{m_i}\beta_i^{(k)}\varphi_k(HJ_n), \tag{2.8}$$

TABLE 1
Stiff order conditions for ExpRB methods up to order 6 (from [18]). Here Z, K, and M denote arbitrary square matrices.

No.	Order condition	Order
1	$\sum_{i=2}^{s} b_i(Z)c_i^2 = 2\varphi_3(Z)$	3
2	$\sum_{i=2}^{s} b_i(Z)c_i^3 = 6\varphi_4(Z)$	4
3	$\sum_{i=2}^{s} b_i(Z)c_i^4 = 24\varphi_5(Z)$	5
4	$\sum_{i=2}^{s} b_i(Z)c_iK\big(\sum_{k=2}^{i-1} a_{ik}(Z)\frac{c_k^2}{2!} - c_i^3\varphi_3(c_iZ)\big) = 0$	5
5	$\sum_{i=2}^{s} b_i(Z)c_i^5 = 120\varphi_6(Z)$	6
6	$\sum_{i=2}^{s} b_i(Z)c_i^2M\big(\sum_{k=2}^{i-1} a_{ik}(Z)\frac{c_k^2}{2!} - c_i^3\varphi_3(c_iZ)\big) = 0$	6
7	$\sum_{i=2}^{s} b_i(Z)c_iK\big(\sum_{k=2}^{i-1} a_{ik}(Z)\frac{c_k^3}{3!} - c_i^4\varphi_4(c_iZ)\big) = 0$	6

and where ℓ_{ij} and m_i are some positive integers. Then, U_{ni} and u_{n+1} are the exact solutions of the (linear) modified differential equations

$$v'_{ni}(\tau) = J_n v_{ni}(\tau) + p_{ni}(\tau), \qquad v_{ni}(0) = u_n, \qquad i = 2, \dots, s, \tag{2.9a}$$

$$v'_{n+1}(\tau) = J_n v_{n+1}(\tau) + q_n(\tau), \qquad v_{n+1}(0) = u_n \qquad , \tag{2.9b}$$

at the times $\tau = c_iH$ and $\tau = H$, respectively. Here, $p_{ni}(\tau)$ and $q_n(\tau)$ are polynomials in τ given by

$$p_{ni}(\tau) = N_n(t_n, u_n) + (t_n + \tau)V_n + \sum_{j=2}^{i-1}\left(\sum_{k=1}^{\ell_{ij}} \frac{\alpha_{ij}^{(k)}}{c_i^k H^{k-1}(k-1)!}\tau^{k-1}\right) D_{nj}, \tag{2.10a}$$

$$q_n(\tau) = N_n(t_n, u_n) + (t_n + \tau)V_n + \sum_{i=2}^{s}\left(\sum_{k=1}^{m_i} \frac{\beta_i^{(k)}}{H^{k-1}(k-1)!}\tau^{k-1}\right) D_{ni}. \tag{2.10b}$$

Proof. The proof can be carried out in a very similar manner as in [16, Theorem 3.1]. Here, we only sketch the main idea. First, we insert the φ_k functions from (2.4) into (2.8) to get the integral representations of $a_{ij}(HJ_n)$ and $b_i(HJ_n)$:

$$a_{ij}(HJ_n) = \int_0^{c_iH} \mathrm{e}^{(c_iH-\tau)J_n} \sum_{k=1}^{\ell_{ij}} \frac{\alpha_{ij}^{(k)}}{(c_iH)^k(k-1)!}\tau^{k-1}\mathrm{d}\tau, \tag{2.11a}$$

$$b_i(HJ_n) = \int_0^{H} \mathrm{e}^{(H-\tau)J_n} \sum_{k=1}^{m_i} \frac{\beta_i^{(k)}}{H^k(k-1)!}\tau^{k-1}\mathrm{d}\tau. \tag{2.11b}$$

Inserting these into (2.5) shows that

$$U_{ni} = \mathrm{e}^{c_iHJ_n}u_n + \int_0^{c_iH} \mathrm{e}^{(c_iH-\tau)J_n}p_{ni}(\tau)\mathrm{d}\tau, \quad i = 2, \dots, s, \tag{2.12a}$$

$$u_{n+1} = \mathrm{e}^{HJ_n}u_n + \int_0^{H} \mathrm{e}^{(H-\tau)J_n}q_n(\tau)\mathrm{d}\tau, \tag{2.12b}$$

which clearly show that $U_{ni} = v_{ni}(c_iH)$ and $u_{n+1} = v_{n+1}(H)$ by means of the variation-of-constants formula applied to (2.9a) and (2.9b), respectively. ∎

MERB methods. Starting from the initial value $u_0 = u(t_0)$, Lemma 2.2 suggests a multirate procedure to approximate the numerical solutions u_{n+1} $(n = 0, 1, 2, \ldots)$ obtained by ExpRB methods. Specifically, one may integrate the slow process $(V_n t + N_n(t, u))$ using a macro time step H and integrate the fast process $(J_n u)$ using a micro time step $h = H/m$ (where $m > 1$ is an integer representing the time scale separation factor) via solving the "fast" ODEs (2.9a) on $[0, c_i H]$ and (2.9b) on $[0, H]$. Let us denote the corresponding numerical solutions of these ODEs as $\widehat{U}_{ni}$ $(\approx v_{ni}(c_i H) = U_{ni})$ and $\hat{u}_{n+1}$ $(\approx v_{n+1}(H) = u_{n+1})$. Then this multirate procedure consists in each step of solving (2.9)–(2.10) with the initial value $\hat{u}_n$ $(\hat{u}_0 = u_0)$. Since we must linearize each step around the approximate solution $\hat{u}_n$ instead of the true value u_n, we denote the approximations of $J_n, V_n, N_n(t, u)$, and D_{nj} appearing in polynomials (2.10) as

$$\hat{J}_n = \frac{\partial F}{\partial u}(t_n, \hat{u}_n),\ \hat{V}_n = \frac{\partial F}{\partial t}(t_n, \hat{u}_n),\ \hat{N}_n(t, u) = F(t, u) - \hat{J}_n u - \hat{V}_n t, \tag{2.13a}$$

$$\widehat{D}_{nj} = \hat{N}_n(t_n + c_j H, \widehat{U}_{nj}) - \hat{N}_n(t_n, \hat{u}_n). \tag{2.13b}$$

Thus, starting with $\hat{u}_0 = u_0$, for each time step $t_n \to t_{n+1}$ we solve perturbed linear ODEs for $i = 2, \ldots, s$,

$$y'_{ni}(\tau) = \hat{J}_n y_{ni}(\tau) + \hat{p}_{ni}(\tau), \quad \tau \in [0, c_i H], \quad y_{ni}(0) = \hat{u}_n, \tag{2.14}$$

with

$$\hat{p}_{ni}(\tau) = \hat{N}_n(t_n, \hat{u}_n) + (t_n + \tau)\hat{V}_n + \sum_{j=2}^{i-1} \left(\sum_{k=1}^{\ell_{ij}} \frac{\alpha_{ij}^{(k)}}{c_i^k H^{k-1}(k-1)!} \tau^{k-1} \right) \widehat{D}_{nj}, \tag{2.15}$$

to obtain

$$\widehat{U}_{ni} \approx y_{ni}(c_i H) \approx v_{ni}(c_i H) = U_{ni}.$$

Then, using these approximations, we find

$$\hat{q}_n(\tau) = \hat{N}_n(t_n, \hat{u}_n) + (t_n + \tau)\hat{V}_n + \sum_{i=2}^{s} \left(\sum_{k=1}^{m_i} \frac{\beta_i^{(k)}}{H^{k-1}(k-1)!} \tau^{k-1} \right) \widehat{D}_{ni} \tag{2.16}$$

and solve one additional linear ODE

$$y'_{n+1}(\tau) = \hat{J}_n y_{n+1}(\tau) + \hat{q}_n(\tau), \quad \tau \in [0, H], \quad y_{n+1}(0) = \hat{u}_n, \tag{2.17}$$

to obtain the update

$$\hat{u}_{n+1} \approx y_{n+1}(H) \approx v_{n+1}(H) = u_{n+1}.$$

Since this process can be derived from ExpRB schemes satisfying (2.8), we call the resulting methods (2.14)–(2.17) *MERB* methods. Note that, since $\hat{U}_{n1}$ and $y_{n1}(0)$ do not enter the MERB scheme, for the sake of completeness, one can define $\hat{U}_{n1} = y_{n1}(0) = \hat{u}_n$.

Remark 2.3. Based on MERB's formulation in (2.14)–(2.17), they have similar structure to MERK methods. Hence, they can retain MERK's interesting features, including very few evaluations of the costly slow components, and they do not require computing matrix functions as ExpRB methods do. The main difference is that at each integration step MERB methods must update the linearization components $\hat{J}_n$, $\hat{V}_n$, $\hat{N}_n$, and $\hat{D}_{nj}$. However, this increased cost may be balanced by the fact that, due to the property (2.7), high-order MERB methods should require considerably fewer modified ODEs than MERK methods of the same order (see section 2.4).

2.3. Convergence analysis of MERB methods.

2.3.1. Analytical framework. To analyze the convergence of MERB methods, we employ the abstract framework of strongly continuous semigroups (see, e.g., [5, 23]) on a Banach space X. Throughout this paper, we denote the norm in X by $\|\cdot\|$. Let

$$J = \frac{\partial F}{\partial u}(t,u) \tag{2.18}$$

be the Fréchet partial derivative of F. We make use of the following assumptions.

Assumption 1. *The Jacobian* (2.18) *is the generator of a strongly continuous semigroup* e^{tJ} *in* X. This implies that there exist constants C and ω such that

$$\left\|\mathrm{e}^{tJ}\right\| \le C\mathrm{e}^{\omega t}, \quad t \ge 0, \tag{2.19}$$

and consequently $\varphi_k(HJ)$, $a_{ij}(HJ)$, and $b_i(HJ)$ are bounded operators.

Assumption 2. *The solution* $u : [t_0, T] \to X$ *of* (1.1) *is sufficiently smooth with derivatives in* X, *and* $F : [t_0, T] \times X \to X$ *is sufficiently Fréchet differentiable in a strip along the exact solution to* (1.1). *All derivatives occurring are assumed to be uniformly bounded.*

Stability bound. Since $\hat{J}_n = \frac{\partial F}{\partial u}(t_n, \hat{u}_n)$ arising in MERB methods changes at every step and $\hat{J}_n \approx J_n$, we also employ the following stability bound (for the discrete evolution operators on X) of ExpRB methods (see [11, section 3.3]) to have

$$\left\| \prod_{j=0}^{n-k} \mathrm{e}^{H\hat{J}_{n-j}} \right\| \le C_{\mathrm{S}}, \qquad t_0 \le t_k \le t_n \le T. \tag{2.20}$$

The importance of this bound is that the constant C_{S} is uniform in k and n, despite the fact that J_n varies from step to step.

2.3.2. A global error representation of MERB methods. Since MERB methods (2.14)–(2.17) result in a numerical solution $\hat{u}_{n+1}$ which approximates the numerical solution u_{n+1} of ExpRB methods (as denoted above) at time t_{n+1}, we will employ the local errors of ExpRB methods to analyze the global error of MERB methods. Throughout the paper the following error notations will be used.

• *Global error notation for MERB methods.* We denote the global error at time t_{n+1} of a MERB method as

$$\hat{e}_{n+1} = \hat{u}_{n+1} - u(t_{n+1}). \tag{2.21}$$

• *Local error notation for ExpRB methods.* We denote the local error at t_{n+1} of the base ExpRB method as

$$\tilde{e}_{n+1} = \tilde{u}_{n+1} - u(t_{n+1}). \tag{2.22}$$

Here, $\tilde{u}_{n+1}$ is the numerical solution of the base ExpRB method obtained after carrying out one step of (2.5) starting from the exact solution $u(t_n)$ as the initial value:

$$\tilde{u}_{n+1} = \mathrm{e}^{H\tilde{J}_n}u(t_n) + H\varphi_1(H\tilde{J}_n)\tilde{V}_n t_n + H^2\varphi_2(HJ_n)\tilde{V}_n + H\sum_{i=1}^{s} b_i(H\tilde{J}_n)\tilde{N}_n(t_n + c_iH, \tilde{U}_{ni}), \tag{2.23a}$$

$$\tilde{U}_{ni} = \mathrm{e}^{c_iH\tilde{J}_n}u(t_n) + c_iH\varphi_1(c_iH\tilde{J}_n)\tilde{V}_n t_n + c_i^2H^2\varphi_2(c_iH\tilde{J}_n)\tilde{V}_n + H\sum_{j=1}^{i-1} a_{ij}(H\tilde{J}_n)\tilde{N}_n(t_n + c_jH, \tilde{U}_{nj}), \tag{2.23b}$$

where

$$\tilde{J}_n = \frac{\partial F}{\partial u}(t_n, u(t_n)),\ \tilde{V}_n = \frac{\partial F}{\partial t}(t_n, u(t_n)),\ \tilde{N}_n(t,u) = F(t,u) - \tilde{J}_n u - \tilde{V}_n t. \tag{2.24}$$

Note that, from Lemma 2.2, (2.23) is equivalent to one step of the MERB scheme starting from the exact initial value $y_{n+1}(0) = u(t_n)$ (for which the solution of the IVP (2.17) on $[0, H]$ is "known" to be $y_{n+1}(H) = \tilde{u}_{n+1}$). Therefore, one can consider that $\tilde{e}_{n+1}$ is also the local error of MERB methods.

• *Global error notation for approximation of the IVP* (2.17). As $\hat{u}_{n+1} \approx y_{n+1}(H)$ (the true solution of the ODE (2.17)), we denote the global error of an ODE solver used for integrating (2.17) on $[0, H]$ as

$$\hat{\varepsilon}_{n+1} = \hat{u}_{n+1} - y_{n+1}(H). \tag{2.25}$$

• *Global error notation for approximation of the IVP* (2.14). Similarly, since $\hat{U}_{ni}$ is the numerical solution of (2.14) on $[0, c_iH]$ obtained by an ODE solver, let us denote the global error of this approximation as

$$\hat{\varepsilon}_{ni} = \hat{U}_{ni} - y_{ni}(c_iH). \tag{2.26}$$

Note that, by applying the variation-of-constants formula to (2.17) and using (2.11b), $y_{n+1}(H)$ can be represented as

$$y_{n+1}(H) = \mathrm{e}^{H\hat{J}_n}\hat{u}_n + H\varphi_1(H\hat{J}_n)\hat{V}_n t_n + H^2\varphi_2(HJ_n)\hat{V}_n + H\sum_{i=1}^{s} b_i(H\hat{J}_n)\hat{N}_n(t_n + c_iH, \hat{U}_{ni}).$$

In view of (2.21), (2.22), and (2.25), we can represent the global error of MERB methods as

$$\hat{e}_{n+1} = \hat{u}_{n+1} - \tilde{u}_{n+1} + \tilde{e}_{n+1} = \hat{\varepsilon}_{n+1} + (y_{n+1}(H) - \tilde{u}_{n+1}) + \tilde{e}_{n+1}. \tag{2.27}$$

To keep our presentation in a compact form, we introduce

$$t_{ni} = t_n + c_iH, \tag{2.28a}$$

$$\hat{B}_n = \varphi_1(H\hat{J}_n)\hat{V}_n t_n + H\varphi_2(H\hat{J}_n)\hat{V}_n + \sum_{i=1}^{s} b_i(H\hat{J}_n)\hat{N}_n(t_{ni}, \hat{U}_{ni}), \tag{2.28b}$$

$$\tilde{B}_n = \varphi_1(H\tilde{J}_n)\tilde{V}_n t_n + H\varphi_2(H\tilde{J}_n)\tilde{V}_n + \sum_{i=1}^{s} b_i(H\tilde{J}_n)\tilde{N}_n(t_{ni}, \tilde{U}_{ni}). \tag{2.28c}$$

Using (2.28), we now derive a full expansion of (2.27), which shows how the global error of MERB methods can be estimated by the sum of the propagated local errors of ExpRB methods and the global errors of the ODE solvers used for (2.14) and (2.17).

THEOREM 2.4. *The global error $\hat{e}_{n+1}$ of MERB methods* (2.14)–(2.17) *at time t_{n+1} can be expressed as*

(2.29)
$$\hat{e}_{n+1} = \underbrace{\left(\prod_{j=0}^{n} \mathrm{e}^{H\hat{J}_{n-j}} - \prod_{j=0}^{n} \mathrm{e}^{H\tilde{J}_{n-j}}\right) u_0}_{Error1} + \underbrace{\sum_{k=0}^{n} \left(\prod_{j=0}^{n-k-1} \mathrm{e}^{H\tilde{J}_{n-j}}\right) \tilde{e}_{k+1}}_{Error2}$$
$$+ \underbrace{\sum_{k=0}^{n} \left(\prod_{j=0}^{n-k-1} \mathrm{e}^{H\hat{J}_{n-j}}\right) \hat{\varepsilon}_{k+1}}_{Error3} + \underbrace{H \sum_{k=0}^{n} \left[\left(\prod_{j=0}^{n-k-1} \mathrm{e}^{H\hat{J}_{n-j}}\right) \hat{B}_k - \left(\prod_{j=0}^{n-k-1} \mathrm{e}^{H\tilde{J}_{n-j}}\right) \tilde{B}_k\right]}_{Error4}.$$

Proof. In view of (2.27), we first study the difference $(y_{n+1}(H) - \tilde{u}_{n+1})$. Using (2.28b) and (2.25) (which implies $\hat{u}_n = y_n(H) + \hat{\varepsilon}_n$), we have

$$y_{n+1}(H) = \mathrm{e}^{H\hat{J}_n} \hat{u}_n + H\hat{B}_n = \mathrm{e}^{H\hat{J}_n} y_n(H) + \mathrm{e}^{H\hat{J}_n} \hat{\varepsilon}_n + H\hat{B}_n. \tag{2.30}$$

Solving this recurrence relation (with $y_0(H) = u(t_0) = u_0$) gives

(2.31)
$$y_{n+1}(H) = \left(\prod_{j=0}^{n} \mathrm{e}^{H\hat{J}_{n-j}}\right) u_0 + \sum_{k=0}^{n-1} \left(\prod_{j=0}^{n-k-1} \mathrm{e}^{H\hat{J}_{n-j}}\right) \hat{\varepsilon}_{k+1} + H \sum_{k=0}^{n} \left(\prod_{j=0}^{n-k-1} \mathrm{e}^{H\hat{J}_{n-j}}\right) \hat{B}_k.$$

Similarly, using (2.28c) and (2.22), we can write $\tilde{u}_{n+1}$ in (2.23a) as

$$\tilde{u}_{n+1} = \mathrm{e}^{H\tilde{J}_n} u(t_n) + H\tilde{B}_n = \mathrm{e}^{H\tilde{J}_n} \tilde{u}_n - \mathrm{e}^{H\tilde{J}_n} \tilde{e}_n + H\tilde{B}_n. \tag{2.32}$$

After solving this recurrence, we end up with

(2.33)
$$\tilde{u}_{n+1} = \left(\prod_{j=0}^{n} \mathrm{e}^{H\tilde{J}_{n-j}}\right) u_0 - \sum_{k=0}^{n-1} \left(\prod_{j=0}^{n-k-1} \mathrm{e}^{H\tilde{J}_{n-j}}\right) \tilde{e}_{k+1} + H \sum_{k=0}^{n} \left(\prod_{j=0}^{n-k-1} \mathrm{e}^{H\tilde{J}_{n-j}}\right) \tilde{B}_k.$$

Subtracting (2.33) from (2.31) and inserting the result into (2.27) prove (2.29). ∎

For the sake of completeness, we have set $\prod_{j=0}^{-1}(\cdot)$ (empty products of operators) to equal the identity in the proof above. This will be used throughout section 2.3.

Next, we prove some preliminary results before estimating the global error $\hat{e}_{n+1}$.

2.3.3. Preliminary results and error bounds.

LEMMA 2.5. *The term Error4 in* (2.29) *can be further expressed as*

(2.34)
$$Error4 = H \sum_{k=0}^{n} \left[\left(\prod_{j=0}^{n-k-1} \mathrm{e}^{H\hat{J}_{n-j}} - \prod_{j=0}^{n-k-1} \mathrm{e}^{H\tilde{J}_{n-j}}\right) \hat{B}_k + \left(\prod_{j=0}^{n-k-1} \mathrm{e}^{H\tilde{J}_{n-j}}\right) (\hat{B}_k - \tilde{B}_k)\right],$$

where

$$\begin{aligned} \hat{B}_k - \tilde{B}_k = &\sum_{j=1}^{2} \left[\left(\varphi_j(H\hat{J}_k) - \varphi_j(H\tilde{J}_k)\right)\tilde{V}_k + \varphi_j(H\hat{J}_k)(\hat{V}_k - \tilde{V}_k)\right] t_k^{2-j} H^{j-1} \\ &+ \sum_{i=1}^{s} \left(b_i(H\hat{J}_k) - b_i(H\tilde{J}_k)\right)\tilde{N}_k(t_{ki}, \tilde{U}_{ki}) \\ &+ \sum_{i=1}^{s} b_i(H\hat{J}_k)\left(\hat{N}_k(t_{ki}, \hat{U}_{ki}) - \tilde{N}_k(t_{ki}, \tilde{U}_{ki})\right). \end{aligned} \tag{2.35}$$

Proof. The derivation of (2.34) is straightforward by subtracting and adding the same term $\prod_{j=0}^{n-k-1} \mathrm{e}^{H\tilde{J}_{n-j}}\hat{B}_k$ within the sum $\sum_{k=0}^{n}[\cdot]$ in *Error*4. Also, by subtracting (2.28c) from (2.28b), one can easily obtain (2.35). ∎

To estimate the difference in the nonlinear terms at each internal MERB and ExpRB stage, $\big(\hat{N}_k(t_{ki},\hat{U}_{ki}) - \tilde{N}_k(t_{ki},\tilde{U}_{ki})\big)$ in (2.35), we first study the difference

$$\hat{E}_{ni} = \hat{U}_{ni} - \tilde{U}_{ni}. \tag{2.36}$$

Denoting

$$\hat{A}_{ni} = c_i\varphi_1(c_iH\hat{J}_n)\hat{V}_n t_n + c_i^2 H\varphi_2(c_iH\hat{J}_n)\hat{V}_n + \sum_{j=1}^{i-1} a_{ij}(H\hat{J}_n)\hat{N}_n(t_{nj},\hat{U}_{nj}), \tag{2.37a}$$

$$\tilde{A}_{ni} = c_i\varphi_1(c_iH\tilde{J}_n)\tilde{V}_n t_n + c_i^2 H\varphi_2(c_iH\tilde{J}_n)\tilde{V}_n + \sum_{j=1}^{i-1} a_{ij}(H\tilde{J}_n)\tilde{N}_n(t_{nj},\tilde{U}_{nj}), \tag{2.37b}$$

we obtain the following result.

LEMMA 2.6. *The difference between $\hat{U}_{ni}$ and $\tilde{U}_{ni}$ can be expressed as*

$$\hat{E}_{ni} = \hat{\varepsilon}_{ni} + \mathrm{e}^{c_iH\hat{J}_n}\hat{e}_n + \big(\mathrm{e}^{c_iH\hat{J}_n} - \mathrm{e}^{c_iH\tilde{J}_n}\big)u(t_n) + H(\hat{A}_{ni} - \tilde{A}_{ni}) \tag{2.38}$$

with

(2.39)

$$\begin{aligned}\hat{A}_{ni} - \tilde{A}_{ni} = &\sum_{\ell=1}^{2}\big[\big(\varphi_\ell(c_iH\hat{J}_n) - \varphi_\ell(c_iH\tilde{J}_n)\big)\tilde{V}_n + \varphi_\ell(c_iH\hat{J}_n)(\hat{V}_n - \tilde{V}_n)\big]c_i^\ell t_n^{2-\ell}H^{\ell-1}\\ &+ \sum_{j=1}^{i-1}\big(a_{ij}(H\hat{J}_n) - a_{ij}(H\tilde{J}_n)\big)\tilde{N}_n(t_{nj},\tilde{U}_{nj})\\ &+ \sum_{j=1}^{i-1} a_{ij}(H\hat{J}_n)\big(\hat{N}_n(t_{nj},\hat{U}_{nj}) - \tilde{N}_n(t_{nj},\tilde{U}_{nj})\big).\end{aligned}$$

Here, $\hat{\varepsilon}_{n1} = \hat{U}_{n1} - y_{n1}(c_1H) = \hat{u}_n - y_{n1}(0) = 0$ (due to $c_1 = 0$), and thus $\hat{E}_{n1} = \hat{e}_n$.

Proof. From (2.36) and (2.26), we have

$$\hat{E}_{ni} = \hat{\varepsilon}_{ni} + y_{ni}(c_iH) - \tilde{U}_{ni}. \tag{2.40}$$

Using (2.37b), one can write $\tilde{U}_{ni}$ given in (2.23b) as

$$\tilde{U}_{ni} = \mathrm{e}^{c_iH\tilde{J}_n}u(t_n) + H\tilde{A}_{ni}. \tag{2.41}$$

By applying the variation-of-constants formula to (2.14) and using (2.11a),

$$y_{ni}(c_iH) = \mathrm{e}^{c_iH\hat{J}_n}\hat{u}_n + H\hat{A}_{ni} = \mathrm{e}^{c_iH\hat{J}_n}(\hat{e}_n + u(t_n)) + H\hat{A}_{ni}, \tag{2.42}$$

where $\hat{A}_{ni}$ is given in (2.37a). Inserting (2.41) and (2.42) into (2.40) gives (2.38). Similarly to (2.35), the expression (2.39) can be verified by subtracting (2.37b) from (2.37a) first and then adding and subtracting to the result the same terms $c_i\varphi_1(c_iH\hat{J}_n)\tilde{V}_n t_n$, $c_i^2H\varphi_2(c_iH\hat{J}_n)\tilde{V}_n$, and $\sum_{j=1}^{i-1} a_{ij}(H\hat{J}_n)\tilde{N}_n(t_n + c_jH,\tilde{U}_{nj})$. ∎

Remark 2.7 (reasonable assumptions on $\hat{e}_n$ and $\hat{E}_{ni}$). In view of the expressions (2.29) and (2.38), one can investigate how the global error $\hat{e}_{n+1} = \hat{e}_{n+1}(H)$ and the difference $\hat{E}_{ni} = \hat{E}_{ni}(H)$ behave as the step size H approaches 0. First, it is clear that $\lim_{H\to 0} Error1 = 0$ and $\lim_{H\to 0} Error4 = 0$. Second, we note that, if the linear ODEs (2.17) and (2.14) of the MERB scheme are solved with ODE solvers which have convergence orders r and q (using a micro time step $h = H/m$, $m > 1$), their global errors behave as $\hat{\varepsilon}_{k+1} = \mathcal{O}(h^r) = \frac{1}{m^r}\mathcal{O}(H^r)$, $\hat{\varepsilon}_{ni} = \mathcal{O}(h^q) = \frac{1}{m^q}\mathcal{O}(H^q)$, respectively. Next, as for the local errors $\tilde{e}_{k+1}$ of the ExpRB methods, in [19, section 3.3] it was shown that these errors are at least proportional to H^3 (for free—without any order conditions). In fact, the stiff order conditions for ExpRB methods of orders up to 6 have been derived so far [18, 20], meaning that $\tilde{e}_{k+1} = \mathcal{O}(H^{p+1})$ $(2 \le p \le 6)$. Putting these together, we observe from (2.29) that

$$\lim_{H\to 0} \hat{e}_{n+1} = \sum_{k=0}^{n} \Big(\lim_{H\to 0} \tilde{e}_{k+1} + \lim_{H\to 0} \hat{\varepsilon}_{k+1} \Big) = \sum_{k=0}^{n} \Big(\lim_{H\to 0} \mathcal{O}(H^{p+1}) + \lim_{H\to 0} \frac{1}{m^r}\mathcal{O}(H^r) \Big) = 0.$$

Using this, we deduce from (2.38) that

$$\lim_{H\to 0} \hat{E}_{ni} = \lim_{H\to 0} \hat{\varepsilon}_{ni} + \lim_{H\to 0} \hat{e}_n = \lim_{H\to 0} \frac{1}{m^q}\mathcal{O}(H^q) + \lim_{H\to 0} \hat{e}_n = 0.$$

Therefore, henceforth we will reasonably assume that $\hat{e}_{n+1}$ and $\hat{E}_{ni}$ remain in a sufficiently small neighborhood of 0 for small step sizes.

Next, we prove several bounds needed to estimate the terms in (2.35) and (2.39). To simplify our presentation within both this and the following subsections, we will use C as a generic constant that may have different values at each occurrence.

LEMMA 2.8. *Under Assumption* 2*, the bound*

$$\|\hat{N}_n(t_{ni}, \hat{U}_{ni}) - \tilde{N}_n(t_{ni}, \tilde{U}_{ni})\| \leqslant C\|\hat{E}_{ni}\| + C\|\hat{J}_n - \tilde{J}_n\| + C\|\hat{V}_n - \tilde{V}_n\| \tag{2.43}$$

holds for all n and i as long as $\hat{E}_{ni}$ remains in a sufficiently small neighborhood of 0.

Proof. First, we split

$$\hat{N}_n(t_{ni}, \hat{U}_{ni}) - \tilde{N}_n(t_{ni}, \tilde{U}_{ni}) = \underbrace{\hat{N}_n(t_{ni}, \hat{U}_{ni}) - \hat{N}_n(t_{ni}, \tilde{U}_{ni})}_{Nsplit1} + \underbrace{\hat{N}_n(t_{ni}, \tilde{U}_{ni}) - \tilde{N}_n(t_{ni}, \tilde{U}_{ni})}_{Nsplit2}.$$

Using (2.13a) and (2.24), we write the term $Nsplit2$ as

$$\begin{aligned} Nsplit2 &= \big(F(t_{ni}, \tilde{U}_{ni}) - \hat{J}_n\tilde{U}_{ni} - \hat{V}_n t_{ni}\big) - \big(F(t_{ni}, \tilde{U}_{ni}) - \tilde{J}_n\tilde{U}_{ni} - \tilde{V}_n t_{ni}\big) \\ &= (\tilde{J}_n - \hat{J}_n)\tilde{U}_{ni} + (\tilde{V}_n - \hat{V}_n)t_{ni}. \end{aligned} \tag{2.44}$$

Expanding $\hat{N}_n(t, U)$ into a Taylor series expansion around $(t_{ni}, \tilde{U}_{ni})$ gives

$$Nsplit1 = \int_0^1 \frac{\partial \hat{N}_n}{\partial u}(t_{ni}, \tilde{U}_{ni} + \theta\hat{E}_{ni})\hat{E}_{ni}\,\mathrm{d}\theta. \tag{2.45}$$

Under Assumption 2, (2.43) follows by bounding $\|Nsplit1\| + \|Nsplit2\|$. ∎

LEMMA 2.9. *Under Assumptions* 1 *and* 2*, the bounds*

$$\|\hat{J}_n - \tilde{J}_n\| \leqslant C\|\hat{e}_n\|, \quad \|\hat{V}_n - \tilde{V}_n\| \leqslant C\|\hat{e}_n\|, \tag{2.46a}$$

$$\|\mathrm{e}^{t\hat{J}_n} - \mathrm{e}^{t\tilde{J}_n}\| \leqslant Ct\|\hat{e}_n\|, \quad t \geq 0, \tag{2.46b}$$

$$\|\varphi_\ell(t\hat{J}_n) - \varphi_\ell(t\tilde{J}_n)\| \leqslant Ct\|\hat{e}_n\|, \quad t \geq 0, \tag{2.46c}$$

$$\|b_i(H\hat{J}_n) - b_i(H\tilde{J}_n)\| \leqslant CH\|\hat{e}_n\|, \tag{2.46d}$$

$$\|a_{ij}(H\hat{J}_n) - a_{ij}(H\tilde{J}_n)\| \leqslant CH\|\hat{e}_n\| \tag{2.46e}$$

hold for all n*,* ℓ*,* i*, and* j*, as long as the global errors* $\hat{e}_n$ *remain in a sufficiently small neighborhood of* 0*.*

Proof. It is straightforward to verify (2.46a) by first noting that $\hat{J}_n - \tilde{J}_n = \frac{\partial F}{\partial u}(t_n, \hat{u}_n) - \frac{\partial F}{\partial u}(t_n, u(t_n))$, $\hat{V}_n - \tilde{V}_n = \frac{\partial F}{\partial t}(t_n, \hat{u}_n) - \frac{\partial F}{\partial t}(t_n, u(t_n))$ and then expanding $\frac{\partial F}{\partial u}(t, u)$, $\frac{\partial F}{\partial t}(t, u)$ in a Taylor series around $(t_n, u(t_n))$ (using with the integral remainder terms of $\mathcal{O}(\|\hat{e}_n\|)$). Next, we estimate the difference between the two semigroups $\mathrm{e}^{t\hat{J}_n}$ and $\mathrm{e}^{t\tilde{J}_n}$ in a similar manner as in [19, Lemma 4.2]. Namely, it is observed that $\mathrm{e}^{t\hat{J}_n}$ is the solution of the IVP

$$w'(t) = \hat{J}_n w(t) = \tilde{J}_n w(t) + (\hat{J}_n - \tilde{J}_n)w(t), \quad w(0) = I.$$

Applying the variation-of-constants formula to this IVP gives

$$\mathrm{e}^{t\hat{J}_n} - \mathrm{e}^{t\tilde{J}_n} = t\int_0^1 \mathrm{e}^{(1-\theta)t\tilde{J}_n}(\hat{J}_n - \tilde{J}_n)\mathrm{e}^{\theta t\hat{J}_n}\,\mathrm{d}\theta.$$

Therefore, (2.46b) follows directly from (2.19) and (2.46a) (the first bound). Using this, (2.46c)–(2.46e) follow from using (2.4) and (2.8) (see also [19, Lemma 4.3]). ∎

Using the results from Lemmas 2.6, 2.8, and 2.9, we obtain the following result.

COROLLARY 2.10. *Under Assumptions* 1 *and* 2*, the estimate*

$$\|\hat{B}_k - \tilde{B}_k\| \leqslant \sum_{j=1}^{s} C\|\hat{\varepsilon}_{kj}\| + C\|\hat{e}_k\| \tag{2.47}$$

holds for all k*, as long as* $\hat{E}_{ki}$ *and the global errors* $\hat{e}_k$ *remain in a sufficiently small neighborhood of* 0*.*

Proof. Using Lemmas 2.9 and 2.8, one can bound (2.35) as

$$\|\hat{B}_k - \tilde{B}_k\| \leqslant C\|\hat{e}_k\| + C\sum_{i=1}^{s}\|\hat{E}_{ki}\|. \tag{2.48}$$

Next, we apply Lemma 2.6 (with $n = k$) to get $\hat{E}_{ki}$ and then estimate it by using (2.19) and Lemma 2.9 (the bound (2.46b)):

$$\|\hat{E}_{ki}\| \leqslant \|\hat{\varepsilon}_{ki}\| + C\|\hat{e}_k\| + H\|\hat{A}_{ki} - \tilde{A}_{ki}\|. \tag{2.49}$$

Again using Lemmas 2.9 and 2.8, the bound on $\|\hat{A}_{ki} - \tilde{A}_{ki}\|$ (see (2.39)) is similar to (2.48). Inserting this into (2.49) and using $\hat{E}_{k1} = \hat{e}_k$ finally show that

$$\|\hat{E}_{ki}\| \leqslant \|\hat{\varepsilon}_{ki}\| + C\|\hat{e}_k\| + \sum_{j=1}^{i-1} C\|\hat{\varepsilon}_{kj}\|. \tag{2.50}$$

It is clear now that (2.47) follows from (2.48) and (2.50). ∎

Finally, we give a technical lemma, which can be later used to estimate the term *Error*1 appearing in (2.29).

LEMMA 2.11. *Let* $\{Z_j\}_{j=0}^n$ *and* $\{Y_j\}_{j=0}^n$ *be two sequences of operators on* X *(the state space). We have*

$$\prod_{j=0}^{n} Z_{n-j} - \prod_{j=0}^{n} Y_{n-j} = \sum_{k=0}^{n} \left(\prod_{j=0}^{n-k-1} Z_{n-j} \right) (Z_k - Y_k) \left(\prod_{j=n-k+1}^{n} Y_{n-j} \right). \tag{2.51}$$

Proof. By adding and subtracting $\prod_{j=0}^{n-1} Z_{n-j} Y_0$ and then $\prod_{j=0}^{n-2} Z_{n-j} Y_0 Y_1$, the left-hand side of (2.51) can be written as

$$\left(\prod_{j=0}^{n-1} Z_{n-j} \right) (Z_0 - Y_0) + \left(\prod_{j=0}^{n-2} Z_{n-j} \right) (Z_1 - Y_1) Y_0 + (Z_n Z_{n-1} \dots Z_2 - Y_n Y_{n-1} \dots Y_2) Y_1 Y_0.$$

We continue adding and subtracting $(\prod_{j=0}^{n-k-1} Z_{n-j})(\prod_{j=n-k+1}^{n} Y_{n-j})$ in this manner until $k = n$ to obtain the right-hand side (2.51). ∎

COROLLARY 2.12. *Under Assumptions* 1 *and* 2*, the estimate*

$$\left\| \prod_{j=0}^{n} \mathrm{e}^{H \hat{J}_{n-j}} - \prod_{j=0}^{n} \mathrm{e}^{H \tilde{J}_{n-j}} \right\| \leqslant H \sum_{k=0}^{n} C \|\hat{e}_k\|. \tag{2.52}$$

holds for all n *as long as the global errors* $\hat{e}_k$ *remain sufficiently small.*

Proof. This follows by applying Lemma 2.11 to $Z_{n-j} = \mathrm{e}^{H \hat{J}_{n-j}}$ and $Y_{n-j} = \mathrm{e}^{H \tilde{J}_{n-j}}$ and by using the stability bound (2.19) and the bound (2.46b) from Lemma 2.9. ∎

2.3.4. MERB convergence. With the above preparation in hand, we are now ready to prove convergence of our MERB methods.

THEOREM 2.13. *Let the IVP* (1.1) *satisfy Assumptions* 1–2*. Consider for its numerical solution a MERB method* (2.14)–(2.17) *that is constructed from an ExpRB method of global order* p *using with macro time step* H*. Let* m *denote the number of fast steps per slow step. If the fast ODEs* (2.14) *and* (2.17) *associated with the MERB method are integrated with micro time step* $h = H/m$ *by using ODE solvers that have global order of convergence* q *and* r*, respectively, then the MERB method is convergent with the error bound*

$$\|\hat{u}_n - u(t_n)\| \leqslant CH^p + Ch^q + Ch^{r-1} = CH^p + \frac{C}{m^q} H^q + \frac{C}{m^r} H^{r-1} \tag{2.53}$$

on compact time intervals $t_0 \leq t_n = t_0 + nH \leq T$*. Here, while the first error constant depends on* $T - t_0$ *(but is independent of* n *and* H*), the second and third error constants also depend on the error constants of the chosen ODE solvers.*

Proof. Using Corollary 2.12, the stability bound (2.19), and Corollary 2.10, one can estimate the four error terms $Error1, Error2, Error3$, and $Error4$ in (2.29), which together show that

$$\|\hat{e}_{n+1}\| \leqslant H \sum_{k=0}^{n} C \|\hat{e}_k\| + \sum_{k=0}^{n} C \|\tilde{e}_{k+1}\| + \sum_{k=0}^{n} C \|\hat{\varepsilon}_{k+1}\| + H \sum_{k=0}^{n} \left(\sum_{j=1}^{s} C \|\hat{\varepsilon}_{kj}\| \right). \tag{2.54}$$

From our assumption that the base ExpRB method has global order p, its local error satisfies $\|\tilde{e}_{k+1}\| \leqslant CH^{p+1}$. Since the fast ODEs (2.14) and (2.17) are integrated by

solvers with global orders of convergence q and r (using a micro time step h), we have $\hat{\varepsilon}_{kj} = \mathcal{O}(h^q)$ and $\hat{\varepsilon}_{k+1} = \mathcal{O}(h^r)$. Inserting these into (2.54) gives

$$\|\hat{e}_n\| \leqslant H \sum_{k=0}^{n-1} C\|\hat{e}_k\| + \sum_{k=0}^{n-1} \big(CH^{p+1} + Ch^r + CHh^q\big). \tag{2.55}$$

The error bound (2.53) results from applying a discrete Gronwall lemma to (2.55). ∎

We next distinguish two cases for (1.1), corresponding to whether the problem is stiff or nonstiff/mildly stiff, in order to further comment on the error bound (2.53).

Remark 2.14 (stiff problems). For stiff problems where the stiffness is dominated by the linear part, $u' = F(t,u) = Lu + g(t,u)$ (L has a large norm or is potentially unbounded), our convergence theory presented above is still valid provided that the linear ODEs (2.14) and (2.17) are solved with stiff solvers. This is because, for such stiff problems, one can prove that the first error constant C in the error bound (2.53) is uniformly bounded independent of the stiffness. First, we can assume that L is the generator of a strongly continuous semigroup e^{tL} in X and $g : [t_0, T] \times X \to X$ is sufficiently Fréchet differentiable (with uniformly bounded derivatives) in a strip along the exact solution. Note that these still imply our Assumption 1 (by using a standard perturbation result of semigroup as noted in [19, section 2.2]) and Assumption 2 (since $F(t,u) = Lu + g(t,u)$). Then, we only need to modify our previous proof for the important bound (2.46a) in Lemma 2.9 such that it now holds with a constant C that is bounded independent of $\|L\|$. In fact, using $J(u) = L + \frac{\partial g}{\partial u}(t,u)$, one simply sees that

$$\|\hat{J}_n - \tilde{J}_n\| = \left\| \frac{\partial g}{\partial u}(t_n, \hat{u}_n) - \frac{\partial g}{\partial u}(t_n, u(t_n)) \right\| \leq C\|\hat{e}_n\|,$$

where the constant C depends only on value uniformly bounded by the assumption on g. Similarly, one has $\|\hat{V}_n - \tilde{V}_n\| = \|\frac{\partial g}{\partial t}(t_n, \hat{u}_n) - \frac{\partial g}{\partial t}(t_n, u(t_n))\| \leq C\|\hat{e}_n\|$. Using these, all of the above proofs still hold, and all bounds associated with terms involving $C\|\hat{e}_n\|$ are still valid with constants C uniformly bounded and independent of the stiffness. The use of stiff solvers for the linear ODEs (2.14) and (2.17) thus guarantees that the second and third error constants of the error bound (2.53) could be also independent of the stiffness. Therefore, for stiff problems, it is suggested from (2.53) to use stiffly accurate solvers of orders $q \geq p$ and $r \geq p+1$ for (2.14) and (2.17), respectively (for a fixed m), to have a stiffly accurate MERB method (2.14)–(2.17) that converges with order p overall. We note, however, that m may need to be larger for stiff problems, and thus if a high-order ODE solver is used for (2.17) (say $r = 4$ or 5), the requirement $r \geq p+1$ may be relaxed to $r \geq p$ as the constant $\frac{1}{m^r}$ in the third error term becomes much smaller.

Remark 2.15 (nonstiff/mildly stiff problems). For nonstiff/mildly stiff problems, one can improve the second and third error terms in the global error bound (2.53). Specifically, since the linear ODEs (2.17) and (2.14) are solved on small intervals $[0, H]$ and $[0, c_i H]$, respectively (using micro time step h), and they share the same Jacobian $\hat{J}_k$ which can be assumed to satisfy $\|\hat{J}_k\| \leqslant M$ (a moderate value), we employ the global error analysis in [9, Theorem 3.4] to derive that

$$\|\hat{\varepsilon}_{k+1}\| \leqslant h^r \tfrac{C}{M}(e^{MH} - 1) = Ch^r H\varphi_1(MH) \leqslant Ch^r H, \tag{2.56a}$$

$$\|\hat{\varepsilon}_{kj}\| \leqslant h^q \tfrac{C}{M}(e^{Mc_iH} - 1) \leqslant Ch^q H\varphi_1(Mc_iH) \leqslant Ch^q H. \tag{2.56b}$$

Using these for (2.54), we get

$$\|\hat{e}_n\| \leqslant H \sum_{k=0}^{n-1} C\|\hat{e}_k\| + \sum_{k=0}^{n-1} \left(CH^{p+1} + Ch^r H + Ch^q H^2\right). \tag{2.57}$$

Applying a discrete Gronwall lemma shows the new error bound

$$\|\hat{e}_n\| \leqslant CH^p + CHh^q + Ch^r = CH^p + \frac{C}{m^q} H^{q+1} + \frac{C}{m^r} H^r,$$

in which we gain an additional factor of H for the second and third error terms when compared to the original error bound (2.53). Thus for a fixed m, a MERB method (2.14)–(2.17) will converge with order p provided that the inner ODE solvers for (2.14) and (2.17) have orders $q \geq p-1$ and $r \geq p$, respectively. This is an improvement compared to MRI-GARK methods [25] (where convergence theory is only available for nonstiff problems) that require both $q \geq p$ and $r \geq p$ for a method of order p.

2.4. Construction of specific MERB methods. Guided by Theorem 2.13, in order to derive MERB methods it is important to begin with base ExpRB methods that satisfy Lemma 2.2. Fortunately, such ExpRB methods are available up to order 6 in the literature; see [11, 19, 20]. In this subsection, we extend some of these methods to give their coefficients more generally and then derive MERB methods of orders 2 through 6 from these schemes. Note that, since a MERB method (2.14)–(2.17) is uniquely characterized by its polynomials $\hat{p}_{ni}(\tau)$ and $\hat{q}_n(\tau)$, we only provide those polynomials here. In particular, we note that these MERB methods require fewer modified ODEs to be solved per slow time step than comparable order MRI-GARK [25] and MERK methods [16]. We further note that for each method we specify its "total fast traversal time," corresponding to how many multiples of $[0, H]$ must occur when solving modified ODEs.

2.4.1. Second-order methods. First, consider the second-order ExpRB-Euler scheme (see [11] and [14, section 1.2.2] for nonautonomous problems)

$$u_{n+1} = u_n + H\varphi_1(HJ_n)F(t_n, u_n) + H^2\varphi_2(HJ_n)V_n.$$

Using Lemma 2.2 we immediately derive a second-order method called `MERB2`:

$$\hat{q}_n(\tau) = \hat{N}_n(t_n, \hat{u}_n) + (t_n + \tau)\hat{V}_n, \quad \tau \in [0, H]. \tag{2.58}$$

This only requires the solution of one modified ODE. We note that, since second-order multirate methods have been available for some time, we do not include `MERB2` in our numerical results and instead focus on higher-order multirate methods.

2.4.2. Third-order methods. In [11], a 2-stage third-order ExpRB method called `exprb32` was constructed (using $c_2 = 1$) for autonomous problems. Extending this to nonautonomous problems and writing this for general c_2, we solve condition 1 of Table 1 directly (with $s = 2$) to give a general family of third-order methods:

$$\begin{aligned} U_{n2} &= u_n + c_2 H\varphi_1(c_2 HJ_n)F(t_n, u_n) + c_2^2 H^2\varphi_2(c_2 HJ_n)V_n, \\ u_{n+1} &= u_n + H\varphi_1(HJ_n)F(t_n, u_n) + H^2\varphi_2(HJ_n)V_n + H\tfrac{2}{c_2^2}\varphi_3(HJ_n)D_{n2}. \end{aligned} \tag{2.59}$$

From this we construct the `MERB3` family of third-order methods:

$$\begin{aligned} \hat{p}_{n2}(\tau) &= \hat{N}_n(t_n, \hat{u}_n) + (t_n + \tau)\hat{V}_n, && \tau \in [0, c_2 H], \\ \hat{q}_n(\tau) &= \hat{N}_n(t_n, \hat{u}_n) + (t_n + \tau)\hat{V}_n + \frac{\tau^2}{c_2^2 H^2}\widehat{D}_{n2}, && \tau \in [0, H]. \end{aligned} \tag{2.60}$$

Clearly, this requires the solution of 2 modified ODEs per slow time step (whereas third-order MERK and MRI-GARK methods require solving 3 modified ODEs per step). In our numerical experiments we take $c_2 = \frac{1}{2}$, which gives rise to a total fast time step traversal for `MERB3` of $(1 + c_2)H = 1.5H$.

2.4.3. Fourth-order methods. There exist several fourth-order ExpRB schemes [11, 19, 20, 15, 17] with coefficients fulfilling Lemma 2.2. However, we chose a 2-stage fourth-order ExpRB method called `exprb42` which was constructed for autonomous problems in [15]. Transforming this to nonautonomous form, we have

$$\begin{aligned} U_{n2} &= u_n + \tfrac{3}{4}H\varphi_1(\tfrac{3}{4}HJ_n)F(t_n,u_n) + \tfrac{9}{16}H^2\varphi_2(\tfrac{3}{4}HJ_n)V_n, \\ u_{n+1} &= u_n + H\varphi_1(HJ_n)F(t_n,u_n) + H^2\varphi_2(HJ_n)V_n + H\tfrac{16}{9}\varphi_3(HJ_n)D_{n2}. \end{aligned} \tag{2.61}$$

We then apply Lemma 2.2 to construct the fourth-order `MERB4` method:

$$\begin{aligned} \hat{p}_{n2}(\tau) &= \hat{N}_n(t_n,\hat{u}_n) + (t_n+\tau)\hat{V}_n, && \tau \in \left[0, \frac{3}{4}H\right] \\ \hat{q}_n(\tau) &= \hat{N}_n(t_n,\hat{u}_n) + (t_n+\tau)\hat{V}_n + \frac{16}{9}\frac{\tau^2}{H^2}\widehat{D}_{n2}, && \tau \in [0,H]. \end{aligned} \tag{2.62}$$

`MERB4` only requires solving 2 modified ODEs per slow time step, whereas fourth-order MRI-GARK and MERK methods require 5 and 4 modified ODEs in each step, respectively. We further note that (2.62) has a total fast traversal time of $\frac{7}{4}H = 1.75H$.

2.4.4. Fifth-order methods. ExpRB methods of order 5 can be found in [19, 20]. Here, for efficiency purposes, we consider a parallel scheme called `pexprb54s4`, whose coefficients (with fixed nodes c_i) satisfy Lemma 2.2. It uses $s = 4$ stages and is embedded with a fourth-order scheme (for step size adaptivity) but can be implemented as a 3-stage method. A detailed derivation of `pexprb54s4` is given in [20] (solving conditions 1–4 of Table 1 with the choices $b_2(Z) = 0$, $a_{43}(Z) = 0$, $a_{32}(Z) = \frac{2c_3^3}{c_2^2}\varphi_3(c_3Z)$, and $a_{42} = \frac{2c_4^3}{c_2^2}\varphi_3(c_4Z)$). Following that derivation, we present here a family of fifth-order ExpRB methods (depending on parameters c_2, c_3, c_4) for nonautonomous problems:

$$\begin{aligned} U_{n2} &= u_n + H\left(c_2\varphi_1(c_2HJ_n)F(t_n,u_n) + c_2^2H\varphi_2(c_2HJ_n)V_n\right), \\ U_{n3} &= u_n + H\left(c_3\varphi_1(c_3HJ_n)F(t_n,u_n) + c_3^2H\varphi_2(c_3HJ_n)V_n + \tfrac{2c_3^3}{c_2^2}\varphi_3(c_3HJ_n)D_{n2}\right), \\ U_{n4} &= u_n + H\left(c_4\varphi_1(c_4HJ_n)F(t_n,u_n) + c_4^2H\varphi_2(c_4HJ_n)V_n + \tfrac{2c_4^3}{c_2^2}\varphi_3(c_4HJ_n)D_{n2}\right), \\ u_{n+1} &= u_n + H\left(\varphi_1(HJ_n)F(t_n,u_n) + H\varphi_2(HJ_n)V_n + b_3(HJ_n)D_{n3} + b_4(HJ_n)D_{n4}\right) \end{aligned} \tag{2.63}$$

with

$$\begin{aligned} b_3(HJ_n) &= \tfrac{1}{c_3^2(c_4-c_3)}\big(c_4\varphi_3(HJ_n) - 6\varphi_4(HJ_n)\big), \\ b_4(HJ_n) &= \tfrac{1}{c_4^2(c_3-c_4)}\big(2c_3\varphi_3(HJ_n) - 6\varphi_4(HJ_n)\big). \\ c_4 &= \tfrac{3(5c_3-4)}{5(4c_3-3)}. \end{aligned}$$

We note that the two internal stages $\{U_{n3}, U_{n4}\}$ are independent of one another and thus can be computed simultaneously. They also have the same format, in that they have the same formula but only act on different inputs c_3 and c_4, which we exploit below to give the same polynomial for their corresponding modified ODEs.

Applying Lemma 2.2 to (2.63) results in the fifth-order family of `MERB5` methods:

$$\begin{aligned}
\hat{p}_{n2}(\tau) &= \hat{N}_n(t_n,\hat{u}_n) + (t_n+\tau)\hat{V}_n, && \tau\in[0,c_2H],\\
\hat{p}_{n3}(\tau) &\equiv \hat{p}_{n4}(\tau) = \hat{N}_n(t_n,\hat{u}_n) + (t_n+\tau)\hat{V}_n + \left(\tfrac{\tau}{c_2H}\right)^2\widehat{D}_{n2}, && \tau\in[0,c_3H],\\
\hat{q}_n(\tau) &= \hat{N}_n(t_n,\hat{u}_n) + (t_n+\tau)\hat{V}_n + \tfrac{\tau^2}{H^2}\left(\tfrac{c_4}{c_3^2(c_4-c_3)}\widehat{D}_{n3} + \tfrac{c_3}{c_4^2(c_3-c_4)}\widehat{D}_{n4}\right)\\
&\quad - \tfrac{\tau^3}{H^3}\left(\tfrac{1}{c_3^2(c_4-c_3)}\widehat{D}_{n3} + \tfrac{1}{c_4^2(c_3-c_4)}\widehat{D}_{n4}\right), && \tau\in[0,H].
\end{aligned} \tag{2.64}$$

This only requires solving 3 modified ODEs per slow step (the only existing fifth-order multirate method, MERK5, requires 5). In our experiments we choose $c_2 = c_4 = \frac{1}{4} < c_3 = \frac{33}{40}$, so we can solve the modified ODE (2.14) using the polynomial $\hat{p}_{n3}(\tau)$ on $[0, c_3H]$ to obtain *both* $\widehat{U}_{n3} \approx U_{n3}$ and $\widehat{U}_{n4} \approx U_{n4}$ (since $c_4 < c_3$), without solving an additional fast ODE on $[0, c_4H]$. Using this strategy, the total fast traversal time for `MERB5` is $(1 + c_2 + c_3)H = \frac{83}{40}H = 2.075H$.

2.4.5. Sixth-order methods. To the best of our knowledge, the only existing ExpRB method of order 6, named `pexprb65s7`, is given in [20]. It uses $s = 7$ stages and is embedded with a fifth-order method. As with (2.63), this method consists of multiple independent internal stages (namely, the stages in two groups $\{U_{n2}, U_{n3}\}$ and $\{U_{n4}, U_{n5}, U_{n6}, U_{n7}\}$) that can be computed simultaneously, which we exploit to implement like a 3-stage method. While `pexprb65s7` is constructed for autonomous problems and uses a set of fixed nodes c_i, we extend the derivation from [20] to construct a family of 7-stage sixth-order methods for nonautonomous problems:

$$\begin{aligned}
U_{nk} &= u_n + c_kH\varphi_1(c_kHJ_n)F(t_n,u_n) + (c_kH)^2\varphi_2(c_kHJ_n)V_n,\ k=2,3,\\
U_{ni} &= u_n + c_iH\varphi_1(c_iHJ_n)F(t_n,u_n) + (c_iH)^2\varphi_2(c_iHJ_n)V_n,\\
&\quad + Ha_{i2}(HJ_n)D_{n2} + Ha_{i3}(HJ_n)D_{n3}, && i=4,5,6,7,\\
u_{n+1} &= u_n + H\varphi_1(HJ_n)F(t_n,u_n) + H^2\varphi_2(HJ_n)V_n + H\sum_{i=4}^{7} b_i(HJ_n)D_{ni},
\end{aligned} \tag{2.65}$$

where

$$\begin{aligned}
a_{i2}(HJ_n) &= \tfrac{1}{c_2^2(c_3-c_2)}\left(2c_i^3c_3\varphi_3(c_iHJ_n) - 6c_i^4\varphi_4(c_iHJ_n)\right),\\
a_{i3}(HJ_n) &= \tfrac{1}{c_3^2(c_2-c_3)}\left(2c_i^3c_2\varphi_3(c_iHJ_n) - 6c_i^4\varphi_4(c_iHJ_n)\right),\\
b_i(HJ_n) &= -2\hat{\alpha}_i\varphi_3(HJ_n) + 6\hat{\eta}_i\varphi_4(HJ_n) - 24\hat{\beta}_i\varphi_5(HJ_n) + 120\hat{\gamma}_i\varphi_6(HJ_n),\\
\hat{\gamma}_i &= \frac{1}{c_i^2(c_i-c_k)(c_i-c_l)(c_i-c_m)}, \qquad \hat{\alpha}_i = c_kc_lc_m\hat{\gamma}_i,\\
\hat{\beta}_i &= (c_k+c_l+c_m)\hat{\gamma}_i, \qquad \hat{\eta}_i = (c_kc_l + c_lc_m + c_kc_m)\hat{\gamma}_i.
\end{aligned}$$

Here $i,k,l,m \in \{4,5,6,7\}$ are distinct indices, and c_i, c_k, c_l, c_m are distinct (positive) nodes. Applying Lemma 2.2 we obtain the first-ever sixth-order infinitesimal multirate method, `MERB6`:

$$\hat{p}_{n2}(\tau) \equiv \hat{p}_{n3}(\tau) = \hat{N}_n(t_n, \hat{u}_n) + (t_n + \tau)\hat{V}_n, \qquad \tau \in [0, c_2 H],$$

$$\hat{p}_{n4}(\tau) \equiv \hat{p}_{n5}(\tau) \equiv \hat{p}_{n6}(\tau) \equiv \hat{p}_{n7}(\tau) = \hat{N}_n(t_n, \hat{u}_n) + (t_n + \tau)\hat{V}_n + \tfrac{\tau^2}{(c_3-c_2)H^2}\left(\tfrac{c_3}{c_2^2}\widehat{D}_{n2} - \tfrac{c_2}{c_3^2}\widehat{D}_{n3}\right) - \tfrac{\tau^3}{(c_3-c_2)H^3}\left(\tfrac{1}{c_2^2}\widehat{D}_{n2} - \tfrac{1}{c_3^2}\widehat{D}_{n3}\right), \quad \tau \in [0, c_4 H],$$

$$\hat{q}_n(\tau) = \hat{N}_n(t_n, \hat{u}_n) + (t_n + \tau)\hat{V}_n - \tfrac{\tau^2}{H^2}\sum_{i=4}^{7}\hat{\alpha}_i\widehat{D}_{ni} + \tfrac{\tau^3}{H^3}\sum_{i=4}^{7}\hat{\eta}_i\widehat{D}_{ni} - \tfrac{\tau^4}{H^4}\sum_{i=4}^{7}\hat{\beta}_i\widehat{D}_{ni} + \tfrac{\tau^5}{H^5}\sum_{i=4}^{7}\hat{\gamma}_i\widehat{D}_{ni}, \qquad \tau \in [0, H].$$

As seen, `MERB6` requires only 3 modified ODEs per slow time step like `MERB5`, reflecting the fact that its base sixth-order ExpRB method (2.65) has the structure of a 3-stage method. `MERB6` can be also implemented in an efficient way by choosing $c_3 < c_2$ and $c_5, c_6, c_7 < c_4$. With these choices, we can solve the modified ODE (2.14) using $\hat{p}_{n2}(\tau)$ on $[0, c_2 H]$ to obtain both $\widehat{U}_{n2} \approx U_{n2}$ and $\widehat{U}_{n3} \approx U_{n3}$ without solving an additional fast ODE on $[0, c_3 H]$. Similarly, we can solve (2.14) using $\hat{p}_{n4}(\tau)$ on $[0, c_4 H]$ to get *all four approximations* $\widehat{U}_{ni} \approx U_{ni}$ $(i = 4, 5, 6, 7)$ without solving 3 additional ODEs on $[0, c_5 H]$, $[0, c_6 H]$, and $[0, c_7 H]$. In our numerical experiments, we take $c_3 = c_5 = \frac{1}{10} < c_2 = c_6 = \frac{1}{9} < c_7 = \frac{1}{8} < c_4 = \frac{1}{7}$. This gives a total fast traversal time of $(1 + c_2 + c_4)H = \frac{79}{63}H \approx 1.253H$.

2.5. MERB method implementation. In Algorithm 2.1 we provide a precise description of the MERB algorithm. We note that, in our implementations of MERB

Algorithm 2.1 MERB method

- **Input:** F; J; V; t_0; u_0; s; c_i $(i = 1, \ldots, s)$; H
- **Initialization:** Set $n = 0$; $\hat{u}_n = u_0$.
  While $t_n < T$
  1. Set $\widehat{U}_{n1} = \hat{u}_n$.
  2. Compute $\widehat{J}_n = J(t_n, \hat{u}_n)$ and $\widehat{V}_n = V(t_n, \hat{u}_n)$.
  3. For $i = 2, \ldots, s$ do
     (a) Find $\hat{p}_{ni}(\tau)$ as in (2.15).
     (b) Solve (2.14) on $[0, c_i H]$ to obtain $\widehat{U}_{ni} \approx y_{ni}(c_i H)$.
  4. Find $\hat{q}_n(\tau)$ as in (2.16)
  5. Solve (2.17) on $[0, H]$ to get $\hat{u}_{n+1} \approx y_{n+1}(H)$.
  6. Update $t_{n+1} := t_n + H$, $n := n + 1$.
- **Output:** Approximate values $\hat{u}_n \approx u_n, n = 1, 2, \ldots$ (where u_n is the numerical solution at time t_n obtained by an ExpRB method).

methods, we found it beneficial to include formulas for $\widehat{N}_n(t, u)$ and $\widehat{D}_{ni}(t, u)$ as additional inputs to the algorithm (provided they can be precomputed) for use in (2.15) and (2.16) to avoid floating-point cancellation errors when seeking very accurate solutions. On the other hand, we note that, within the MERB algorithm, both the products Jw and $V\tau$ can be approximated from F using finite differences,

$$J(t, u)w = \tfrac{1}{\sigma}\left(F(t, u + \sigma w) - F(t, u)\right) + \mathcal{O}(\sigma) \quad \text{and}$$
$$V(t, u)\tau = \tfrac{1}{\sigma}\left(F(t + \sigma\tau, u) - F(t, u)\right) + \mathcal{O}(\sigma),$$

instead of J and V being provided analytically; however, when seeking high accuracy, then such approximations can cause excessive floating-point cancellation error.

3. Numerical experiments. In this section, we test MERB methods on select multirate problems to demonstrate their convergence rates, efficiency, and applicability to stiff systems of ODEs. In section 3.1 we examine a semilinear nonautonomous system with bidirectional coupling between the fast and slow variables. For this problem, we compare the proposed `MERB3`–`MERB6` methods with other recently developed multirate methods that treat the slow time scale explicitly, namely, `MERK3`, `MERK4`, and `MERK5` from [16], plus `MRI-GARK-ERK33a` and `MRI-GARK-ERK45a` from [25]. In section 3.2 we test the MERB methods on a much stiffer 2D, Gray–Scott reaction-diffusion PDE system. MATLAB implementations of all tests are provided on Github [4].

We provide three types of "log-log" efficiency plots that compare solution error versus different cost measurements: slow function calls, total function calls, and MATLAB runtimes, respectively. In such plots, the most efficient method corresponds to the curve that is closest to the bottom left corner. We compute solution error as the maximum absolute error over all spatial grid points and time outputs, as measured against either an analytical solution or highly accurate reference solution. We also estimate convergence rates using the maximum pointwise convergence rate once each method is within the asymptotic convergence regime. Each of our efficiency measurements tells a different story. First, slow function calls illustrate the cost of a multirate method when applied to an IVP system with expense dominated by the slow components $F_s(t,u)$. Second, total function calls capture the cost of $F_f(t,u)$ and highlight properties of methods related to their total fast traversal times. Lastly, even though MATLAB runtimes are a poor proxy for performance on HPC applications, we use these to capture the costs associated with dynamic linearization and to measure how these costs affect efficiency.

3.1. Bidirectional coupling system. Inspired by [6, section 5.1], we propose the semilinear, nonautonomous bidirectional coupling problem on $0 < t \le 1$

$$u' = \sigma v - w - \beta t, \tag{3.1a}$$

$$v' = -\sigma u, \tag{3.1b}$$

$$w' = -\lambda(w+\beta t) - \beta\left(u - \frac{a(w+\beta t)}{a\lambda + b\sigma}\right)^2 - \beta\left(v - \frac{b(w+\beta t)}{a\lambda + b\sigma}\right)^2, \tag{3.1c}$$

with exact solution $u(t) = \cos(\sigma t) + ae^{-\lambda t}$, $v(t) = -\sin(\sigma t) + be^{-\lambda t}$, and $w(t) = (a\lambda + b\sigma)e^{-\lambda t} - \beta t$. This problem features linear coupling from slow to fast time scales through (3.1a) and nonlinear coupling from fast to slow time scales through the equation for (3.1c). In addition, it includes tunable parameters $\{a, b, \beta, \lambda, \sigma\}$ taken here to be $\{1, 20, 0.01, 5, 100\}$, with $a\sigma = b\lambda$; σ determines the frequency of the fast time scale, and β controls the strength of the nonlinearity. In the case of dynamic linearization, smaller values of β correspond with weaker nonlinearity, resulting in higher values of the optimal time scale separation factor $m = H/h$.

While the splitting of this IVP into fast and slow components, $u'(t) = F(t,u) = F_f(t,u) + F_s(t,u)$, for MERB methods is dictated by the dynamic linearization process at each time step,

$$\hat{u}'(t) = F(t,\hat{u}(t)) = \left[\hat{J}_n\hat{u}(t)\right] + \left[\hat{V}_n t + \hat{N}_n(t,\hat{u}(t))\right] := F_f(t,u) + F_s(t,u), \tag{3.2}$$

MERK and MRI-GARK methods do not require dynamic linearization and thus have more freedom in how they are partitioned. While MERK methods require that $F_f(t,u) = \mathcal{L}u$, MRI-GARK methods support arbitrary splittings. Therefore, for this

TABLE 2
Multirate method properties: Number of slow internal stages and modified ODEs, total fast traversal times, and optimal m factors for (3.1).

Method	Slow stages	Modified ODEs	Fast traversal time of $[0, H]$	Bidirect. coupling optimal m	
				Dynamic	Fixed
`MERB3`	2	2	1.5	80	
`MERK3`	3	3	2.166	80	10
`MRI-GARK33a`	3	3	1	80	10
`MERB4`	2	2	1.75	40	
`MERK4`	6	4	2.833	40	10
`MRI-GARK45a`	5	5	1	40	1
`MERB5`	4	3	2.075	10	
`MERK5`	10	5	3.2	10	10
`MERB6`	7	3	1.253	5	

problem we consider two separate fast-slow splittings: in addition to the dynamic linearization, we consider a fixed splitting informed by the exact solution

$$F_f(t,\mathbf{u}) = \begin{bmatrix} \sigma v \\ -\sigma u \\ 0 \end{bmatrix}, \quad F_s(t,\mathbf{u}) = \begin{bmatrix} -w-\beta t \\ 0 \\ -\lambda(w+\beta t) - \beta\left(u - \frac{a(w-\beta t)}{a\lambda+b\sigma}\right)^2 - \beta\left(v - \frac{b(w-\beta t)}{a\lambda+b\sigma}\right)^2 \end{bmatrix};$$

in the ensuing results we call this the "fixed linearization." We denote methods run with the fixed linearization using an asterisk; e.g., `MERK3`* uses a fixed linearization, while `MERK3` uses dynamic linearization.

We note that, for problems that are dominated by their linear portion, $\hat{J}_n$, the dynamic linearization (3.2) can place more dynamics at the fast time scale than other fixed multirate splittings, thereby offering a potential for greater multirate accuracy at the expense of constructing the dynamic linearization at each slow step. To determine the optimal m for each splitting we follow the experimental approach from [16] that compares efficiency in terms of slow-only function evaluations and total (slow+fast) function evaluations for several different values of H and m corresponding to each multirate and inner method pairing. These values are given in Table 2 and largely confirm that dynamic linearization can leverage larger time scale separation factors than the fixed linearization (3.1).

Our implementations of multirate methods of the same order use identical explicit fast integrators for solving all modified ODEs. Third-order methods use a 3-stage $\mathcal{O}(h^3)$ method from [2, equation (233f)], fourth-order methods use Kutta's 4-stage $\mathcal{O}(h^4)$ method from [13], and fifth-order methods use the 8-stage $\mathcal{O}(h^5)$ explicit part of ARK5(4)8L[2]SA from [12], while `MERB6` uses an 8-stage $\mathcal{O}(h^6)$ method based on the 8,5(6) procedure of [29]. We assess error at 20 equally spaced points within the time interval and consider slow steps $H = 0.05 \times 2^{-k}$ for $k = 0, 1, \ldots, 7$.

Figures 1–3 show accuracy and efficiency results for this problem. Examining the legends from each figure, we see that all methods attain their expected order of convergence. In Figure 1, all $\mathcal{O}(H^3)$ methods incorporating dynamic linearization have similar errors, coinciding with their uniform time scale separation factor of $m = 80$. Similarly, the methods using fixed linearization `MERK3`* and `MRI-GARK33a`* have the same $m = 10$, leading to comparable errors. Dynamic linearization leads to lower errors than fixed linearization at the same step size (here up to 10^3). Examining method efficiency, we see that the proposed `MERB3` is the most efficient when considering

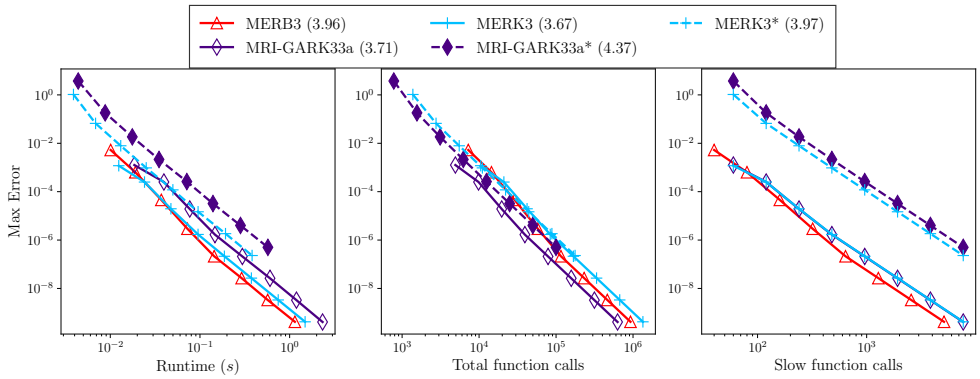

FIG. 1. *Convergence rates (given in parentheses in the legend) and efficiency of $\mathcal{O}(H^3)$ methods on the bidirectional coupling problem of section* 3.1.

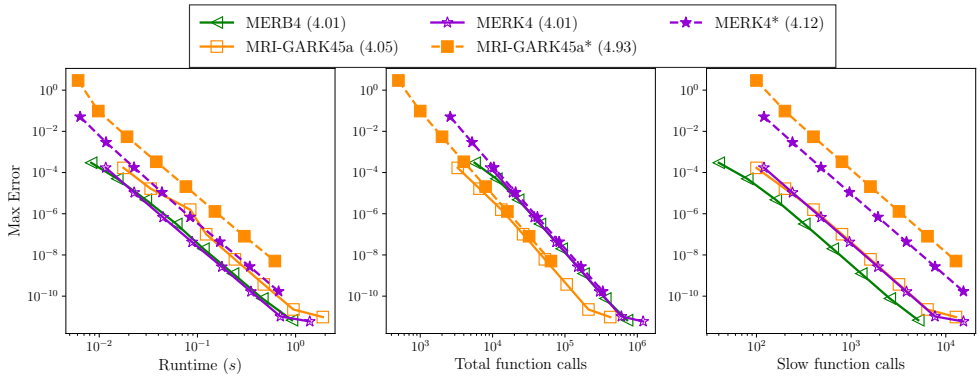

FIG. 2. *Convergence rates (in parentheses in the legend) and efficiency of $\mathcal{O}(H^4)$ methods on the bidirectional coupling problem of section* 3.1.

both overall runtime and slow function evaluations, while `MRI-GARK33a` is slightly more efficient in total function evaluations. Of particular note, we see a significant slow function call benefit for all methods that use dynamic linearization.

We plot results for $\mathcal{O}(H^4)$ methods in Figure 2. As seen for the $\mathcal{O}(H^3)$ methods, $\mathcal{O}(H^4)$ methods using dynamic linearization achieve improved error at the same H in comparison to those using fixed linearization. Here, `MERB4` and `MERK4` show optimal runtime efficiency, with `MRI-GARK-ERK45a` close behind. The MRI-GARK methods are slightly more efficient in total function calls, while `MERB4` is more efficient in slow function calls.

Finally, we compare the performance of $\mathcal{O}(H^5)$ and $\mathcal{O}(H^6)$ methods in Figure 3. The accuracy of the $\mathcal{O}(H^5)$ methods is almost identical on this test problem, with `MERB6` starting with slightly higher error but quickly catching up due to its higher convergence rate. The two new MERB methods are the most efficient for this test problem by all metrics. Focusing on runtime efficiency, `MERB5` is slightly more efficient at larger error values but is passed by `MERB6` at smaller errors. Focusing on function calls, `MERB6` is the most efficient in total function calls due to its smaller total traversal time, whereas the small number of stages for `MERB5` renders it more efficient in terms of slow function calls.

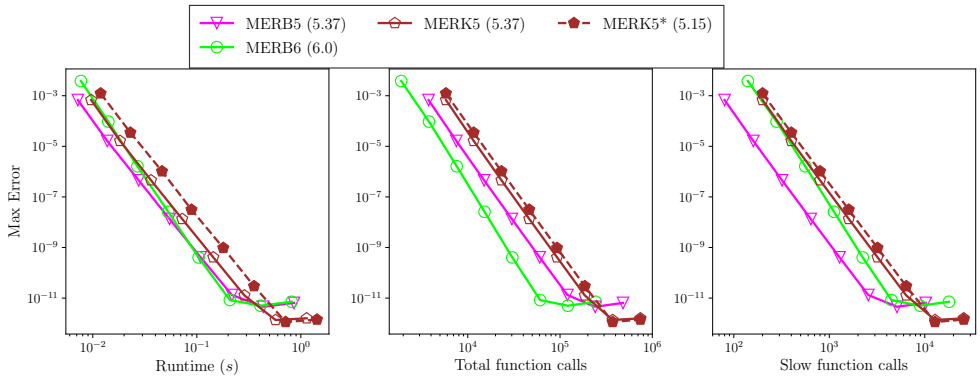

FIG. 3. *Convergence rates (given in parentheses in the legend) and efficiency of $\mathcal{O}(H^5)$ and $\mathcal{O}(H^6)$ methods on the bidirectional coupling problem of section* 3.1.

3.2. Gray–Scott model. As a challenge problem to test MERB methods in the stiff regime, we consider the Gray–Scott reaction-diffusion PDE [8]:

$$\begin{aligned} \partial_t u &= D_u \nabla^2 u - uv^2 + A(1-u), \\ \partial_t v &= D_v \nabla^2 v + uv^2 - (A+B)v, \end{aligned} \tag{3.3}$$

where $u(x,y,t)$ and $v(x,y,t)$ are defined over the domain $[0,1]\times[0,1]\times(0,0.2]$, satisfy periodic boundary conditions, and are spatially discretized with 50 centered finite difference grid points in each direction. Here the reaction coefficients are $A = 0.625, B = 0.25$, and the diffusion coefficients are $D_u = 0.312$ and $D_v = 0.156$. The initial conditions are Gaussian pulses

$$u(x,y,0) = 1 - e^{-150\left((x-0.5)^5+(y-0.5)^2\right)}, \quad v(x,y,0) = e^{-150\left((x-0.5)^5+2(y-0.5)^2\right)}.$$

With these parameters the Jacobian norm at the initial condition is 6.2×10^3, corresponding to a moderately stiff problem. We compute error by comparing against a reference solution (obtained using MATLAB's `ode15s` with relative and absolute tolerances 10^{-13} and 10^{-14}) at 10 evenly spaced points in time, and we test all methods with slow time steps $H = 0.01\times2^{-k}$ for $k = 0,\dots,7$. All methods use a time scale separation factor of $m = 10$. Due to the problem's stiffness, we employ fully implicit Runge–Kutta methods for the fast integration: `MERB3` uses a 2-stage RadauIIA method, `MERB4` uses a 3-stage LobattoIIIC method, `MERB5` uses a 3-stage RadauIIA method, and `MERB6` uses a 4-stage LobattoIIIC method [10].

To more readily compare the proposed MERB methods against one another, in Figure 4 we overlay plots showing the efficiency of these methods according to each of our three cost metrics. Again, as seen in the legend each method attains its theoretical convergence rate on even this significantly stiffer test problem. We note that, at the largest step size of $H = 0.01$ (the left-most point on each curve), `MERB5` has the least error for this problem, followed by `MERB4`, `MERB3`, and then `MERB6`. Due to this larger initial error, `MERB6` is only optimal when considering runtime efficiency at the smallest error values (unlike for the bidirectional test problem shown in Figures 1–3, where it is considerably more competitive in multiple metrics). Due to its low initial error and high convergence rate, `MERB5` is the most efficient of all MERB methods across a wide range of error levels and cost metrics, with `MERB3` and `MERB4` optimal for only the highest error values.

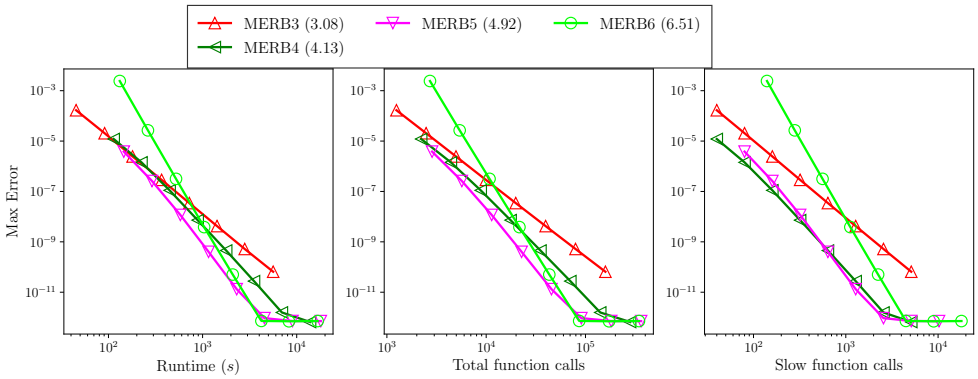

FIG. 4. *Convergence rates (given in parentheses in the legend) and efficiency of all MERB methods on the Gray–Scott problem of section* 3.2.

4. Conclusions. We have introduced a new approach for multirate integration of IVPs that evolve on multiple time scales. Employing an MIS-like approach wherein the coupling between slow and fast time scales occurs through defining a sequence of modified IVPs at the fast time scale and built off of existing ExpRB methods, the proposed MERB methods allow creation of multirate methods with very high order and minimize the amount of costly processing of the slow time scale operator. In addition to deriving a clear mechanism for constructing these from certain classes of ExpRB schemes, we provide rigorous convergence analysis for MERB methods. We note that the style of this analysis is much more elegant than our approach for MERK methods [16] in that we analyze the overall MERB error by separately quantifying the error between the MERB approximation of the underlying ExpRB method and the error in the ExpRB approximation of the original IVP. With this theory in hand, we propose a suite of MERB methods with orders 2 through 6, where in the cases of orders 3–6, we additionally provide generalizations of the base ExpRB methods and extend these to nonautonomous problems.

We examine the performance of the proposed MERB methods of orders 3 through 6 on two test problems: a nonautonomous bidirectional coupling problem and a 2D Gray–Scott model. For the bidirectional coupling problem, we compare MERB methods against existing MERK and explicit MRI-GARK methods, where the MERK and MRI-GARK methods are tested with two potential multirate splittings on each problem. While all MERB, MERK, and MRI-GARK methods exhibited their theoretical convergence rates on this problem and splittings, their efficiency varies. In order to provide results that potentially apply to a broad range of multirate applications, we investigate efficiency using three separate measurements of cost: MATLAB runtime, total function calls (both fast and slow), and slow function calls only. Within these metrics, some general patterns emerge. First, most of the methods exhibited optimal efficiency at higher $m = H/h$ values when using multirate splittings based on dynamic linearization as opposed to fixed splittings. Second, the proposed MERB methods show the best runtime efficiency of all methods and splittings, although in some cases the equivalent order MERK method with dynamic splitting is competitive. Third, due to their total fast time scale traversal times of $1.0H$, the MRI-GARK methods always exhibit the best total function call efficiency. Lastly, due to their low number of slow stages, the proposed MERB methods are uniformly the most efficient when considering slow function calls (only in a few instances MERK with dynamic

splitting was competitive). This is particular of interest for multirate problems where the fast component is much less costly to compute than the slow component. For the moderately stiff Gray–Scott system, we demonstrate that MERB methods maintain their expected orders of accuracy expanding the set of problems to which they are applicable.

Based on these results, we find that the newly proposed MERB methods provide a unique avenue to construction of high-order MIS-like multirate methods and that they are very competitive in comparison with other recently developed high-order MIS-like multirate schemes. More work remains, however. An obvious extension is to include embeddings to enable low-cost temporal error estimation, as well as to investigate robust techniques for error-based multirate time step adaptivity. A further extension of MERB methods could focus on applications that require implicit or mixed implicit-explicit treatment of processes at the slow time scale.

REFERENCES

[1] T. P. Bauer and O. Knoth, *Extended multirate infinitesimal step methods: Derivation of order conditions*, J. Comput. Appl. Math., 387 (2019), 112541.

[2] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, John Wiley & Sons, Hoboken, NJ, 2008.

[3] R. Chinomona and D. R. Reynolds, *Implicit-explicit multirate infinitesimal GARK methods*, SIAM J. Sci. Comput., 43 (2021), pp. A3082–A3113.

[4] R. Chinomona, D. R. Reynolds, and V. T. Luan, *Multirate exponential Rosenbrock methods (MERB)*, 2021, https://github.com/rujekoc/merbrepo.

[5] K. J. Engel and R. Nagel, *One-Parameter Semigroups for Linear Evolution Equations*, Springer, New York, 2000.

[6] D. Estep, V. Ginting, and S. Tavener, *A posteriori analysis of a multirate numerical method for ordinary differential equations*, Comput. Methods Appl. Mech. Engrg., 223–224 (2012), pp. 10–27.

[7] D. E. K. et al., *Multiphysics simulations: Challenges and opportunities*, Int. J. High Perform. Comput. Appl., 27 (2013), pp. 4–83.

[8] P. Gray and S. Scott, *Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Oscillations and instabilities in the system* $A + 2B \to 3B$; $B \to C$, Chem. Eng. Sci., 39 (1984), pp. 1087–1097.

[9] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations* I*: Nonstiff Problems*, 2nd ed., Springer, Berlin, 1993.

[10] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations* II*: Stiff and Differential-Algebraic Problems*, Springer, New York, 1996.

[11] M. Hochbruck, A. Ostermann, and J. Schweitzer, *Exponential Rosenbrock-type methods*, SIAM J. Numer. Anal., 47 (2009), pp. 786–803.

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

[13] W. Kutta, *Beitrag zur näherungsweisen integration totaler differentialgleichungen*, Z. Math. Phys., 46 (1901), pp. 435–453.

[14] V. T. Luan, *High-Order Exponential Integrators*, Ph.D. thesis, University of Innsbruck, 2014.

[15] V. T. Luan, *Fourth-order two-stage explicit exponential integrators for time-dependent PDEs*, Appl. Numer. Math., 112 (2017), pp. 91–103.

[16] V. T. Luan, R. Chinomona, and D. R. Reynolds, *A new class of high-order methods for multirate differential equations*, SIAM J. Sci. Comput., 42 (2020), pp. A1245–A1268.

[17] V. T. Luan and D. L. Michels, *Efficient exponential time integration for simulating nonlinear coupled oscillators*, J. Comput. Appl. Math., 391 (2021), p. 113429.

[18] V. T. Luan and A. Ostermann, *Exponential B-series: The stiff case*, SIAM J. Numer. Anal., 51 (2013), pp. 3431–3445.

[19] V. T. Luan and A. Ostermann, *Exponential Rosenbrock methods of order five–construction, analysis and numerical comparisons*, J. Comput. Appl. Math., 255 (2014), pp. 417–431.

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

[21] G. I. Marchuk, *Some application of splitting-up methods to the solution of mathematical physics problems*, Aplikace Mat., 13 (1968), pp. 103–132.

[22] R. I. McLachlan and G. R. W. Quispel, *Splitting methods*, Acta Numer., 11 (2002), pp. 341–434.
[23] A. Pazy, *Semigroups of Linear Operators and Applications to Partial Differential Equations*, Springer, New York, 1983.
[24] S. Roberts, A. Sarshar, and A. Sandu, *Coupled multirate infinitesimal GARK schemes for stiff systems with multiple time scales*, SIAM J. Sci. Comput., 42 (2020), pp. A1609–A1638.
[25] A. Sandu, *A class of multirate infinitesimal GARK methods*, SIAM J. Numer. Anal., 57 (2019), pp. 2300–2327.
[26] M. Schlegel, O. Knoth, M. Arnold, and R. Wolke, *Multirate Runge–Kutta schemes for advection equations*, J. Comput. Appl. Math., 226 (2009), pp. 345–357.
[27] J. M. Sexton and D. R. Reynolds, *Relaxed Multirate Infinitesimal Step Methods for Initial-Value Problems*, preprint, arXiv:1808.03718, 2018, https://arxiv.org/abs/1808.03718.
[28] G. Strang, *On the construction and comparison of difference schemes*, SIAM J. Numer. Anal., 5 (1968), pp. 506–517.
[29] J. H. Verner, *Explicit Runge–Kutta methods with estimates of the local truncation error*, SIAM J. Numer. Anal., 15 (1978), pp. 772–790.
[30] J. Wensch, O. Knoth, and A. Galant, *Multirate infinitesimal step methods for atmospheric flow simulation*, BIT, 49 (2009), pp. 449–473.